

Subjective assignment

December 18, 2019

1 Subjective assignment

Q1. Write a Python program to get the string from the given string where all the occurrence of its first char has been changed to '\$,' except first char itself?

```
In [ ]: import pandas as pd

In [2]: a = input('Enter a string: ')
        l=list(a)
        l1=[]
        x = 0
        for y in range(0,len(l)):
            for i in l[y]:
                l1.append(i)
                if l1.count(i) > 1:
                    l[y] = '$'
        print(''.join(l))
        #print(l1)
```

```
Enter a string: prospect
pros$ect
```

Q2. Write a Python program to get the single string from the two given strings, and separated by the space and swap the first two characters of each string?

```
In [4]: import pandas as pd
        a = input('Enter string with space: ')
        b = pd.Series(a.split())
        if b.count() >2:
            print('please enter only 2 strings')
        else:
            str1=b.str.slice(stop=2)
            str2=b.str.slice(start=2)
            print(str1[1]+str2[0]+' '+str1[0]+str2[1])
```

```
Enter string with space: August September
Segust Auptember
```

Q3. Write the Python program to add 'ing' at the end of the given string (length of the string should be at least 3). If given string already ends with 'ing,' then add 'ly' instead. If string length of the given string is less than 3, leave it unchanged?

```
In [5]: import pandas as pd
        a = input('Enter a string: ')
        b = pd.Series([a])
        if len(a) <3:
            print(a)
        elif len(a) >=3:
            if b.str.endswith('ing').bool():
                print(b.str.cat(['ly'],sep=""))
            else:
                print(b.str.cat(['ing'],sep=""))
```

```
Enter a string: Writing
0    Writingly
dtype: object
```

Q4. Write the Python program to find the first appearance of the substring 'not' and 'poor' from the given string, if 'not' follows the 'poor', replace the whole 'not...' poor' substring with 'good'.Return the resulting string.

```
In [8]: a = input('Enter a string: ')
        l= list(a.split(' '))
        #print(l)
        cnt = 0
        nl=[]
        str1=" "
        for i in l:
            if i == 'not' or (cnt ==1 and i != 'poor'):
                cnt = 1
                nl.append(i)
            elif i == 'poor' and cnt ==1:
                cnt = 0
                nl.append(i)
                if len(nl) == 3:
                    break
        #print(nl)
        if len(nl) ==0:
            print(a)
        else:
            n = a.replace(str1.join(nl),'good')
            print(n)
```

```
Enter a string: The lyrics are not that poor The lyrics are poor
The lyrics are good The lyrics are poor
```

Q5. Write the Python program to remove the characters which have odd index values of a given string.

In [9]: *#index value starts from 0,1,2,3....*

```
def even_odd(x):
    if x==0 or x%2==0:
        return 'Even'
    else: return 'Odd'

a = input('Enter a string: ')
#b = pd.Series([a])
l = list(a)
for i in range(0,len(l)):
    if even_odd(i)=='Even':
        print(a[i],end='')
```

Enter a string: amazing
aaig

Q6. Write the Python function to get a string made of 4 copies of the last two characters of the specified string (length must be at least 2).

```
In [10]: a = input('Enter a string: ')
b = a[-2:]
l1=[]
for i in range(0,4):
    l1.append(b)
print(''.join(l1))
```

Enter a string: vijay
ayayayay

Q7. Write the python function to get a string made of its first three characters of a specified string. If the length of the string is less than 3 then return the original string.

```
In [11]: def func(a):
    if len(a) >=3:
        print(a[0:3])
    else:
        print(a)

In [12]: a = input('Enter a string: ')
func(a)
```

Enter a string: Amazing
Ama

Q8. Write the python program to print the following floating numbers up to 2 decimal places?

```
In [13]: try:
          a = float(input("Enter float number: "))
        except ValueError:
            print('Please enter a float number')
        else:
            #print('{a:2f}')
            print('round off value is %.2f' % a)
```

Enter float number: 23.555
round off value is 23.55

Q9. Write the Python program to format a number with a percentage?

```
In [16]: try:
          a = float(input("Enter a float number to convert into %age: "))
        except ValueError:
            print('Please enter a float number')
        else:
            print('Percentage value is {:.1%}'.format(a))
```

Enter a float number to convert into %age: 0.0234
Percentage value is 2.3%

Q10. Write the Python program to count occurrences of a substring in a String?

```
In [17]: xx = input('Enter a sentence seperated by space: ')
          l2=list(xx.split(' '))
          l3=[]
          for i in l2:
              if i not in l3:
                  l3.append(i)
                  print(i,l2.count(i))
```

Enter a sentence seperated by space: john john john max max
john 3
max 2

Q11. Write the Python program to count repeated characters in a string.

```
In [18]: a = input('Enter a string: ')
          l=[]
          df = pd.DataFrame({'ch': [], 'cnt': []})
          for i in a:
              #counter = counter + 1
              if a.count(i) > 1 and i not in l:
                  l.append(i)
```

```

        new_row = [{'ch':i, 'cnt':a.count(i)}]
        df = df.append(new_row)
        print(i,a.count(i))
df = df.sort_values(['cnt'], ascending=[False])
df

```

Enter a string: amazing
a 2

```

Out[18]:   ch  cnt
         0  a  2.0

```

Q12. Write the Python program to print the square and cube symbol in the area of a rectangle and volume of a cylinder?

```

In [19]: a = 12.332228888882
        b = str(a)
        dec = len(b) - b.index('.')-1
        print("The area of the rectangle is {0:.{1}f}cm\u00b2".format(a, dec))
        print("The volume of the cylinder is {0:.{1}f}cm\u00b3".format(a, dec))

```

The area of the rectangle is 12.332228888882cmš
The volume of the cylinder is 12.332228888882cmş

Q13. Write the Python program to check if a string contains all letters of the alphabet?

```

In [20]: lt = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t']
        nn = input('Enter a string ')
        cnt=0
        for i in lt:
            if i not in nn:
                print('String does not have all alphabets')
                break
            else:
                cnt=cnt+1
        if cnt == 26:
            print('String has all alphabets')

```

Enter a string china
String does not have all alphabets

Q14. Write the Python program to find the second most repeated word in a given string?

```

In [21]: from collections import Counter
        a = input('Enter a string: ')
        l=[]
        l2=list(a.split(' '))
        cnt = Counter(l2)
        print(cnt.most_common(2)[-1]) #takes only first 2 most common values and display only

```

Enter a string: john john john max max michael
('max', 2)

Q15. Write the Python program to find the minimum window in the given string, which will contains all the characters of another given strings?

```
In [22]: import collections
def min_window(str1, str2):
    result_char, missing_char = collections.Counter(str2), len(str2)
    i = p = q = 0
    for j, c in enumerate(str1, 1):
        missing_char -= result_char[c] > 0
        result_char[c] -= 1
        if not missing_char:
            while i < q and result_char[str1[i]] < 0:
                result_char[str1[i]] += 1
                i += 1
            if not q or j - i <= q - p:
                p, q = i, j
    return str1[p:q]

str1 = "PRWSOERIUSFK"
str2 = "OSU"
print("Original Strings:\n",str1,"\n",str2)
print("Minimum window:")
print(min_window(str1,str2))
#     print(j,end=' ')
```

Original Strings:

PRWSOERIUSFK

OSU

Minimum window:

OERIUS

Q16. Write the Python program to find smallest window that contains all characters of the given string?

```
In [23]: from collections import defaultdict

def find_sub_string(str):
    str_len = len(str)

    # Count all distinct characters.
    dist_count_char = len(set([x for x in str]))

    ctr, start_pos, start_pos_index, min_len = 0, 0, -1, 9999999999
    curr_count = defaultdict(lambda: 0)
```

```

for i in range(str_len):
    curr_count[str[i]] += 1

    if curr_count[str[i]] == 1:
        ctr += 1

    if ctr == dist_count_char:
        while curr_count[str[start_pos]] > 1:
            if curr_count[str[start_pos]] > 1:
                curr_count[str[start_pos]] -= 1
            start_pos += 1

        len_window = i - start_pos + 1
        if min_len > len_window:
            min_len = len_window
            start_pos_index = start_pos
    return str[start_pos_index: start_pos_index + min_len]

str1 = "asdaewsqgtwwsa"
print("Original Strings:\n",str1)
print("\nSmallest window that contains all characters of the said string:")
print(find_sub_string(str1))

```

Original Strings:
asdaewsqgtwwsa

Smallest window that contains all characters of the said string:
daewsqgt

Q17. Write the Python program to count number of substrings from a given string of lowercase alphabets with exactly k distinct (given) characters?

```

In [24]: def count_k_dist(str1, k):
    str_len = len(str1)

    result = 0

    ctr = [0] * 27

    for i in range(0, str_len):
        dist_ctr = 0

        ctr = [0] * 27

        for j in range(i, str_len):

            if(ctr[ord(str1[j]) - 97] == 0):

```

```

        dist_ctr += 1

    ctr[ord(str1[j]) - 97] += 1

    if(dist_ctr == k):
        result += 1
    if(dist_ctr > k):
        break

    return result

str1 = input("Input a string (lowercase alphabets):")
k = int(input("Input k: "))
print("Number of substrings with exactly", k, "distinct characters : ", end = "")
print(count_k_dist(str1, k))

```

Input a string (lowercase alphabets):wolf
 Input k: 2
 Number of substrings with exactly 2 distinct characters : 3

Q18. Write the Python program to count number of non-empty substrings of the given string?

```

In [47]: str1 = input("Input a string: ")
        l = len(str1);
        print(int(l * (l + 1) / 2))

```

Input a string: vijay
 15

Q19. Write the Python program to count number of substrings with same first and last characters of the given string?

```

In [44]: x = input("Input a string: ")
        cnt = 0
        n = len(x)
        for i in range(n):
            for j in range(i, n):
                if (x[i] == x[j]):
                    cnt = cnt + 1
        print(cnt)

```

Input a string: abc
 3

Q20. Write the Python program to count the number of strings where the string length is 2 or more, and first and last character are same from a given list of strings.


```
In [26]: mm = input('Enter a list of string values with space ')
new_list=list(mm.split(' '))
new_list = [i for i in new_list if i != '']
c = 0
#print(new_list)
for i in new_list:
    if len(i)>=2 and i[0] == i[-1]:
        c = c+1
print('string with first and last character are same: ',c)
```

Enter a list of string values with space mam mam john john
string with first and last character are same: 2

Q21. Write the Python program to get a list, sorted in increasing order by the last element in each tuple from the given list of non-empty tuples?

```
In [28]: lst = [(9,10,2),(2,8),(1,3,4),(1,1,5,6)]
#empty list
el = []
#new list
new_list = []
#list for testing purpose
chk_list = []
for i in lst:
    #print(i[-1])
    el.append(i[-1])
el.sort()
#print(el)
for z in el:
    for k in lst:
        if k[len(k)-1] == z and k not in chk_list:
            chk_list.append(k)
            new_list.append(k)
print(new_list)
```

[(9, 10, 2), (1, 3, 4), (1, 1, 5, 6), (2, 8)]

Q22. Write the Python program to remove duplicates from a list?

```
In [29]: mm = input('Enter list values seperated by space ')
lst=list(mm.split(' '))
new_list = []
for i in lst:
    if i not in new_list:
        new_list.append(i)
print(new_list)
```

Enter list values seperated by space mam mam john john
['mam', 'john']

Q23. Write the Python program to find the list of words that are longer than n from a given list of words?

```
In [32]: zz = input('Enter a list of string seperated by comma ')
xx = int(input('Enter the length of word '))
n_list=list(zz.split(','))
e_list=[]
n_list = [i for i in n_list if i != '']
for i in n_list:
    if len(i) > xx:
        e_list.append(i)
print(e_list)
```

Enter a list of string seperated by comma micheal,john,vincent,richard
Enter the length of word 5
['micheal', 'vincent', 'richard']

Q24. Write the Python program to print a specified list after removing the 0th, 4th, and 5th elements?

```
In [34]: zz = input('Enter a list of string seperated by comma: ')
n_list=list(zz.split(','))
l_list=[]
for i in range(0,len(n_list)):
    #print(i)
    if i in (0,4,5):
        l_list.append(n_list[i])
for j in l_list:
    n_list.remove(j)
print(n_list)
```

Enter a list of string seperated by comma: abc,vijay,john,max,richard,isaac
['vijay', 'john', 'max']

Q25. Write the Python program to generate all permutations of a list in Python?

```
In [35]: # permutations using library function
a = ([1,2,3,4])
from itertools import permutations
p= permutations(a)
# Print the obtained permutations
for i in list(p):
    print(i)
```

```

(1, 2, 3, 4)
(1, 2, 4, 3)
(1, 3, 2, 4)
(1, 3, 4, 2)
(1, 4, 2, 3)
(1, 4, 3, 2)
(2, 1, 3, 4)
(2, 1, 4, 3)
(2, 3, 1, 4)
(2, 3, 4, 1)
(2, 4, 1, 3)
(2, 4, 3, 1)
(3, 1, 2, 4)
(3, 1, 4, 2)
(3, 2, 1, 4)
(3, 2, 4, 1)
(3, 4, 1, 2)
(3, 4, 2, 1)
(4, 1, 2, 3)
(4, 1, 3, 2)
(4, 2, 1, 3)
(4, 2, 3, 1)
(4, 3, 1, 2)
(4, 3, 2, 1)

```

Q26. Write the Python program to convert a pair of values into a sorted unique array?

```

In [36]: lst = [(9,10),(2,8),(1,3),(1,1)]
         new_list = []
         unique_list = []
         for k in lst:
             for l in k:
                 new_list.append(l)
         new_list.sort()
         print(new_list)
         for z in new_list:
             if z not in unique_list:
                 unique_list.append(z)
         #unique_list.sort()
         print(unique_list)

[1, 1, 1, 2, 3, 8, 9, 10]
[1, 2, 3, 8, 9, 10]

```

Q27. Write the Python class to convert an integer to a roman numeral?

```

In [37]: class integer_to_roman_class:
         def int_to_Roman(self, num):

```

```

val = [1000, 900, 500, 400,100, 90, 50, 40,10, 9, 5, 4,1]
syb = ["M", "CM", "D", "CD","C", "XC", "L", "XL","X", "IX", "V", "IV","I"]
roman_num = ''
i = 0
while num > 0:
    for j in range(num // val[i]):
        roman_num += syb[i]
        num -= val[i]
    i += 1
return roman_num
print(integer_to_roman_class().int_to_Roman(1))
print(integer_to_roman_class().int_to_Roman(4000))

```

I
MMMM

Q28 Write the Python class to convert a Roman numeral to an integer?

```

In [38]: class py_solution:
def roman_to_int(self, s):
    rom_val = {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}
    int_val = 0
    for i in range(len(s)):
        if i > 0 and rom_val[s[i]] > rom_val[s[i - 1]]:
            int_val += rom_val[s[i]] - 2 * rom_val[s[i - 1]]
        else:
            int_val += rom_val[s[i]]
    return int_val

print(py_solution().roman_to_int('MMM'))
print(py_solution().roman_to_int('MMMM'))
print(py_solution().roman_to_int('C'))

```

3000
4000
100

Q29. Write the Python class to find the validity of the string of the parentheses, '(', ')', '{', '}', '[' and ']' and the brackets must be closed in the correct order, example - "()" and "()" are valid but "[)", "([)]" and "{{{" are invalid.

```

In [40]: class chk_parenthese:
def is_valid_parenthese(self, str1):
    stack = []
    pchar = {"(": ")", "{": "}", "[": "]"}
    for i in str1:
        if i in pchar:

```

```

        stack.append(i)
    elif len(stack) == 0 or pchar[stack.pop()] != i:
        return False
    return len(stack) == 0
print(chk_parenthese().is_valid_parenthese("(){}[]"))
print(chk_parenthese().is_valid_parenthese("() [{}])"))
print(chk_parenthese().is_valid_parenthese("("))

```

True
False
True

Q30. Write the Python class to get all possible unique subsets from a set of distinct integers?

```

In [41]: class unique_class:
        def u_set(self):
            from itertools import combinations
            i = [4,5,6]
            for j in range(0, len(i)+1):
                for jj in combinations(i, j):
                    print(jj)
unique_class().u_set()

```

()
(4,)
(5,)
(6,)
(4, 5)
(4, 6)
(5, 6)
(4, 5, 6)

Q31. Write the Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals the specific target number?

```

In [42]: lst = [10,20,10,40,20,30,70]
        target = 50
        class sample:
            def pairsum(list_of_values, target_value):
                new_lst = []
                lst_tup = []
                l = len(list_of_values)-1
                for i in range(0,len(list_of_values)):
                    if l != i and list_of_values[i] + list_of_values[i+1] == target_value:
                        print('index of pair of elements:',i,i+1)

```

```

In [43]: sample.pairsum(lst,target)

```

index of pair of elements: 2 3
index of pair of elements: 4 5

Q32. Write the Python class to find the three elements that sum to zero from the set of n real numbers?

```
In [44]: from itertools import combinations
i = [-25, -10, -7, -3, 2, 4, 8, 10]
ii = list(combinations(i,3))
e_list = []
for j in ii:
    if sum(j) == 0:
        e_list.append(j)
print(e_list)

[(-10, 2, 8), (-7, -3, 10)]
```

Q33. Write the Python class to implement pow(x, n)?

```
In [46]: class class_pow:
    def pow(self, x, n):
        if x==0 or x==1 or n==1:
            return x

        if x==-1:
            if n%2 ==0:
                return 1
            else:
                return -1
        if n==0:
            return 1
        if n<0:
            return 1/self.pow(x,-n)
        val = self.pow(x,n//2)
        if n%2 ==0:
            return val*val
        return val*val*x
print(class_pow().pow(10, 5));

100000
```

Q34. Write the Python class which has two methods get_String and print_String. get_String accept the string from the user and print_String print the string in upper case.

```
In [47]: class str_class:
    def __init__(self):
```

```

        self.st=""
    def get_string(self):
        self.st = input('Enter a string: ')
    def print_string(self):
        print(self.st.upper())

```

```

In [48]: st = str_class()
        st.get_string()
        st.print_string()

```

Enter a string: michael
MICHAEL

Q35. Write the Python class named Rectangle constructed by a length and width and the method which will compute the area of the rectangle?

```

In [12]: class rectangle:
        def __init__(self):
            self.st=""
        def get_input(self):
            self.l = int(input('Enter length: '))
            self.b = int(input('Enter breadth: '))
        def calc_rectangle(self):
            return (self.l*self.b)
    st = rectangle()
    st.get_input()
    st.calc_rectangle()

```

Enter length: 3
Enter breadth: 7

Out[12]: 21

Q36. Write the Python class named Circle constructed by the radius and two methods which will compute the area and perimeter of the circle?

```

In [15]: class circle:
        def __init__(self):
            self.st=""
        def calc_area(self,r):
            return(3.14*r*r)
        def calc_perimeter(self,r):
            return (2*3.14*r)
    st = circle()
    st.calc_area(3)

```

Out[15]: 28.259999999999998

```
In [16]: st.calc_perimeter(3)
```

```
Out[16]: 18.84
```

Q37. Write the Python program to get the class name of an instance in Python?

```
In [17]: import numpy as np
         x = np.array([1,2,3,4])
         print(type(x).__name__)
```

```
ndarray
```

Q38. Write the Python program to count the number of students of individual class?

```
In [4]: import collections as c
        classes = ( ('V', 1), ('VI', 1), ('V', 2), ('VI', 2), ('VI', 3), ('VII', 1) )
        student_count = c.Counter(class_name for class_name, no_students in classes)
        print(student_count)
```

```
Counter({'VI': 3, 'V': 2, 'VII': 1})
```

Q39. Write the Python program to create an instance of an OrderedDict using the given dictionary and sort dictionary during the creation and print members of the dictionary in reverse order?

```
In [7]: import collections as c
        d = {'Angola': 244, 'Andorra': 376, 'Algeria': 213, 'Afghanistan': 93, 'Albania': 355}
        new_d = c.OrderedDict(d.items())
        for i in new_d:
            print (i, new_d[i])

        print("\nIn reverse order:")
        for j in reversed(new_d):
            print (j, new_d[j])
```

```
Angola 244
Andorra 376
Algeria 213
Afghanistan 93
Albania 355
```

```
In reverse order:
Albania 355
Afghanistan 93
Algeria 213
Andorra 376
Angola 244
```


Q40. Write the Python program to compare two unordered lists (not sets)?

```
In [9]: a = [1, 2, 3, 1, 2, 3]
        b = [3, 2, 1, 3, 2, 1]
        len(a)==len(b) and len(a)==sum([1 for i,j in zip(a,b) if i==j])
```

Out[9]: False

Q41. Write the Python program to get an array buffer information?

```
In [14]: import array as ar
        a = ar.array("I", (1,2))
        print("Array buffer start address in memory and number of elements.")
        print(a.buffer_info())
```

```
array('I', [1, 2])
Array buffer start address in memory and number of elements.
(2482260966064, 2)
```

Q42. Write the Python program to convert an array to an array of machine values and return the bytes representation?

```
In [52]: from array import *
        print("Bytes to String: ")
        sample_array = ar.array('b', [1, 2, 3, 4, 5, 6])
        s = sample_array.tobytes()
        print(s)
```

```
Bytes to String:
b'\x01\x02\x03\x04\x05\x06'
```

Q43. Write the Python program to read a string and interpreting the string as an array of machine values?

```
In [17]: from array import array
        import binascii
        array1 = array('i', [7, 8, 9, 10])
        print('array1:', array1)
        as_bytes = array1.tobytes()
        print('Bytes:', binascii.hexlify(as_bytes))
        array2 = array('i')
        array2.frombytes(as_bytes)
        print('array2:', array2)
```

```
array1: array('i', [7, 8, 9, 10])
Bytes: b'0700000008000000090000000a000000'
array2: array('i', [7, 8, 9, 10])
```

Q44. Write the Python program to push three items into the heap and return the smallest item from the heap. Also, return and pop the smallest item from the heap?

```
In [25]: import heapq
         heap = []
         heapq.heappush(heap, ('VII', 22))
         heapq.heappush(heap, ('VI', 1))
         heapq.heappush(heap, ('VIII', 32))
         print("Items in the heap:")
         for a in heap:
             print(a)
         print("-----")
         print("return the smallest item.")
         print(heapq.heappop(heap))
         print("Pop the smallest item in the heap")
         for a in heap:
             print(a)
```

```
Items in the heap:
('VI', 1)
('VII', 22)
('VIII', 32)
-----
return the smallest item.
('VI', 1)
Pop the smallest item in the heap
('VII', 22)
('VIII', 32)
```

Q45. Write the Python program to locate the left insertion point for a specified value in sorted order?

```
In [56]: import bisect
         def index(a, x):
             i = bisect.bisect_left(a, x)
             return i

         a = [1,2,3,5,7]
         print(index(a, 6))
         print(index(a, 3))
```

```
4
2
```

Q46. Write the Python program to create the FIFO queue?

```
In [27]: import queue
         q = queue.Queue()
```

```

#insert items at the end of the queue
for x in range(4):
    q.put(str(x))
    print(q.get(), end=" ")

```

0 1 2 3

Q47. Write the Python program to calculate the harmonic sum of n-1. Note: The harmonic sum is the sum of reciprocals of the positive Integers?

```

In [28]: def harmonic_sum(n):
        if n < 2:
            return 1
        else:
            return 1 / n + (harmonic_sum(n - 1))

        print(harmonic_sum(1))
        print(harmonic_sum(4))

```

1
2.0833333333333333

Q48. Write the NumPy program to create a 2d array with 6 on the border and 0 inside?

```

In [60]: import numpy as np
        #n = np.empty((5, 5))
        n = np.full((5,5),6)
        n[1:-1,1:-1] = 0
        n

Out[60]: array([[6, 6, 6, 6, 6],
               [6, 0, 0, 0, 6],
               [6, 0, 0, 0, 6],
               [6, 0, 0, 0, 6],
               [6, 6, 6, 6, 6]])

```

Q49. Write the NumPy program to create a 8x8 matrix and fill it with the checkerboard pattern?

```

In [66]: import numpy as np
        x = np.ones((3,3))
        print("Checkerboard pattern:")
        x = np.full((8,8),3,dtype=int)
        x[1::2,::2] = 9
        x[:,2,1::2] = 9
        print(x)

```

Checkerboard pattern:
[[3 9 3 9 3 9 3 9]

```
[9 3 9 3 9 3 9 3]
[3 9 3 9 3 9 3 9]
[9 3 9 3 9 3 9 3]
[3 9 3 9 3 9 3 9]
[9 3 9 3 9 3 9 3]
[3 9 3 9 3 9 3 9]
[9 3 9 3 9 3 9 3]]
```

Q50. Write the NumPy program to create an empty and a full array.

```
In [9]: import numpy as np
        y = np.empty((3, 4))
        print(y)
        x = np.full((3,3),6)
        print(x)

[[ 0.00000000e+000  0.00000000e+000  0.00000000e+000  0.00000000e+000]
 [ 9.33883053e+252 -1.46899100e-310  0.00000000e+000  0.00000000e+000]
 [ 0.00000000e+000  0.00000000e+000  0.00000000e+000  0.00000000e+000]]
[[6 6 6]
 [6 6 6]
 [6 6 6]]
```

Q51. Write the NumPy program to convert the values of Centigrade degrees into the Fahrenheit degrees and the centigrade values are stored in the NumPy array.

```
In [10]: import numpy as np
         fvalues = [0, 12, 45.21, 34, 99.91]
         F = np.array(fvalues)
         print("Values in Fahrenheit degrees:")
         print(F)
         print("Values in Centigrade degrees:")
         print(5*F/9 - 5*32/9)

Values in Fahrenheit degrees:
[ 0.  12.  45.21 34.  99.91]
Values in Centigrade degrees:
[-17.77777778 -11.11111111  7.33888889  1.11111111 37.72777778]
```

Q52. Write the NumPy program to find the real and imaginary parts of an array of complex numbers?

```
In [17]: import numpy as np
         x = np.sqrt([1+0j,0+1j ])
         print("Original array:x ",x)
         print("Real part of the array:")
```

```

print(x.real)
print("Imaginary part of the array:")
print(x.imag)

```

```

Original array:x [1.          +0.j          0.70710678+0.70710678j]
Real part of the array:
[1.          0.70710678]
Imaginary part of the array:
[0.          0.70710678]

```

Q53. Write the NumPy program to test whether each element of a 1-D array is also present in the second array?

```

In [67]: a = np.array([0,10,20,40,60])
        b = np.array([0,40])
        np.in1d(a,b)

```

```

Out[67]: array([ True, False, False,  True, False])

```

Q54. Write the NumPy program to find common values between two arrays?

```

In [70]: a = np.array([0,10,20,40,60])
        b = np.array([10,30,40])
        # function used to check whether each element of a 1-D array is also present in a se
        print('Common values between a and b is: ',np.intersect1d(a,b))

```

```

Common values between a and b is:  [10 40]

```

Q55. Write the NumPy program to get the unique elements of an array?

```

In [72]: a = np.array([10, 10, 20, 20, 30, 30])
        i = np.unique(a)
        i

```

```

Out[72]: array([10, 20, 30])

```

```

In [74]: b = np.array([[1,1],[2,3]])
        j = np.unique(b)
        j

```

```

Out[74]: array([1, 2, 3])

```

Q56. Write the NumPy program to find the set exclusive-or of two arrays. Set exclusive-or will return the sorted, unique values that are in only one (not both) of the input arrays?

```

In [ ]: Array1= np.array([0, 10, 20, 40, 60, 80])
        Array2= np.array([10, 30, 40, 50, 70])

```

```

#find common values between 2 arrays
it = Array1[np.in1d(Array1,Array2)]

#concatenate both arrays
con = np.unique(np.concatenate((Array1,Array2), axis=0))

#Find the set difference of two arrays.
#Return the unique values in Array1 that are not in Array2.

Return the unique values in ar1 that are not in ar2
z = np.setdiff1d(con,it)
z

```

Q57. Write the NumPy program to test if all elements in an array evaluate to True ?

```

In [75]: print(np.all([[True,False],[True,True]])) # return false because of one of the value
print(np.all([1,2,3,4])) # return true because all are numbers
print(np.all([0,2,3,4])) # return False because one of the value is 0

```

```

False
True
False

```

Q58 Write the NumPy program to test whether any array element along the given axis evaluates to True?

```

In [ ]: # return True if one of the element of a evaluate to True
print(np.any([[True,False],[True,True]]))
print(np.any([1,2,3,4]))
print(np.any([0,2,3,4]))

```

Q59. Write the NumPy program to construct an array by repeating?

```

In [77]: # based on 2nd parameter, the numbers will repeat
np.tile([1,2,3,4], 2)

```

```

Out[77]: array([1, 2, 3, 4, 1, 2, 3, 4])

```

Q60. Write the NumPy program to find the indices of the maximum and minimum values with the given axis of an array?

```

In [78]: a = np.array([1,2,3,4,5,6])
print('Index of max value:',np.argmax(a,axis=0))
print('Index of min value:',np.argmin(a,axis=0))

```

```

Index of max value: 5
Index of min value: 0

```

Q61. Write the NumPy program compare two arrays using numpy?

```
In [19]: a = np.array([1,2])
         b = np.array([4,5])
         print(a>b)
         print(a>=b)
         print(a<b)
         print(a<=b)
```

```
[False False]
[False False]
[ True  True]
[ True  True]
```

Q62. Write the NumPy program to sort an along the first, last axis of an array?

```
In [80]: import numpy as np
         arr = np.array([[4, 6],[2, 1]])
         print('Original array: ')
         print(arr)
         print('Sort along the first axis: ')
         x = np.sort(arr, axis=0)
         print(x)
         print("Sort along the last axis: ")
         y = np.sort(x, axis=1)
         print(y)
```

Original array:

```
[[4 6]
 [2 1]]
```

Sort along the first axis:

```
[[2 1]
 [4 6]]
```

Sort along the last axis:

```
[[1 2]
 [4 6]]
```

Q63. Write the NumPy program to sort pairs of first name and last name return their indices (first by last name, then by first name).

```
In [81]: import numpy as np
         first_names = ('Betsey', 'Shelley', 'Lanell', 'Genesis','Margery')
         last_names = ('Battle', 'Brien', 'Plotner', 'Stahl','Woolum')
         x = np.lexsort((first_names, last_names))
         print(x)
```

```
[0 1 2 3 4]
```

Q64. Write the NumPy program to get the values and indices of the elements that are bigger than 10 in the given array?

```
In [82]: org= np.array([[0, 10, 20], [20, 30, 40]])  
         # show array value greater than 10  
         a = org[org >10]  
         a
```

```
Out[82]: array([20, 20, 30, 40])
```

```
In [83]: # show the indices of array stored in a  
         np.where(np.isin(org, a))
```

```
Out[83]: (array([0, 1, 1, 1], dtype=int64), array([2, 0, 1, 2], dtype=int64))
```

Q65. Write the NumPy program to find the memory size of a NumPy array?

```
In [84]: import numpy as np  
         #array filled with zero  
         zz = np.zeros((4,4))  
         zz  
         print('size of zz array in bytes:',zz.nbytes)
```

```
size of zz array in bytes: 128
```

Q66. Write the NumPy program to create an array of ones and an array of zeros?

```
In [85]: import numpy as np  
         #array filled with zero  
         z = np.zeros((4,4))  
         z.astype(int)
```

```
Out[85]: array([[0, 0, 0, 0],  
                [0, 0, 0, 0],  
                [0, 0, 0, 0],  
                [0, 0, 0, 0]])
```

```
In [86]: #array filled with 1  
         o = np.ones((4,4))  
         o.astype(int)
```

```
Out[86]: array([[1, 1, 1, 1],  
                [1, 1, 1, 1],  
                [1, 1, 1, 1],  
                [1, 1, 1, 1]])
```

Q67. Write the NumPy program to change the dimension of an array?


```
In [22]: import numpy as np
x = np.array([1, 2, 3, 4, 5, 6])
print("6 rows and 0 columns")
print(x.shape)

y = np.array([[1, 2, 3],[4, 5, 6],[7,8,9]])
print("3 rows and 3 columns ")
print(y)

x = np.array([1,2,3,4,5,6,7,8,9])
print("Change array shape to (3, 3) -> 3 rows and 3 columns ")
x.shape = (3, 3)
print(x)
```

```
6 rows and 0 columns
(6,)
(3, 3) -> 3 rows and 3 columns
[[1 2 3]
 [4 5 6]
 [7 8 9]]
Change array shape to (3, 3) -> 3 rows and 3 columns
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

Q68. Write the NumPy program to create a new shape to an array without changing its data ?

```
In [87]: #create a array with shape of (2,3)
a = np.array([[1,2], [3, 4], [5, 6]])
# use reshape function to change the dimension of rows and columns
a.reshape(2,3)
```

```
Out[87]: array([[1, 2, 3],
               [4, 5, 6]])
```

Q69. Write the NumPy program to create a new array of 3*5, filled with 2?

```
In [88]: import numpy as np
n = np.empty((3, 5))
n.fill(2)
n.astype(int)
```

```
Out[88]: array([[2, 2, 2, 2, 2],
               [2, 2, 2, 2, 2],
               [2, 2, 2, 2, 2]])
```

Q70. Write the NumPy program to create a 3-D array with ones on a diagonal and zeros elsewhere?

```
In [89]: import numpy as np
         n = np.eye(3)
         print(n)
```

```
[[1.  0.  0.]
 [0.  1.  0.]
 [0.  0.  1.]]
```

Q71. Write the NumPy program to split an array of 14 elements into the 3 arrays and each of which has 2, 4, and 8 elements in original order?

```
In [90]: org_array= np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14])
         org_array
         np.split(org_array, [2,6])
```

```
Out[90]: [array([1, 2]), array([3, 4, 5, 6]), array([ 7,  8,  9, 10, 11, 12, 13, 14])]
```

Q72. Write the NumPy program to split of an array of shape 4x4 it into two arrays along the second axis ?

```
In [91]: sample_arr = np.array([[ 0, 1, 2, 3], [ 4, 5, 6, 7], [ 8, 9, 10, 11], [12, 13, 14, 15])
         #sample_arr
         np.split(sample_arr,2,axis=1)
```

```
Out[91]: [array([[ 0,  1],
                [ 4,  5],
                [ 8,  9],
                [12, 13]]), array([[ 2,  3],
                [ 6,  7],
                [10, 11],
                [14, 15]])]
```

Q73. Write the NumPy program to create a 5x5 matrix with row values ranging from 0 to 4?

```
In [92]: import numpy as np
         n = np.empty((5, 5))
         n.fill(0)
         a = np.array([[0,1,2,3,4]])
         z = n + a
         z
```

```
Out[92]: array([[0., 1., 2., 3., 4.],
                [0., 1., 2., 3., 4.],
                [0., 1., 2., 3., 4.],
                [0., 1., 2., 3., 4.],
                [0., 1., 2., 3., 4.]])
```

Q74. Write the NumPy program to create an array of zeros and three column types (integer, float, character)?

```
In [93]: import numpy as np
x = np.zeros((3,), dtype=('i4,f4,a40'))
new_data = [(1, 2., "Albert Einstein"), (2, 2., "Edmond Halley"), (3, 3., "Gertrude B
x[:] = new_data
print(x)
```

```
[(1, 2., b'Albert Einstein') (2, 2., b'Edmond Halley')
 (3, 3., b'Gertrude B. Elion')]
```

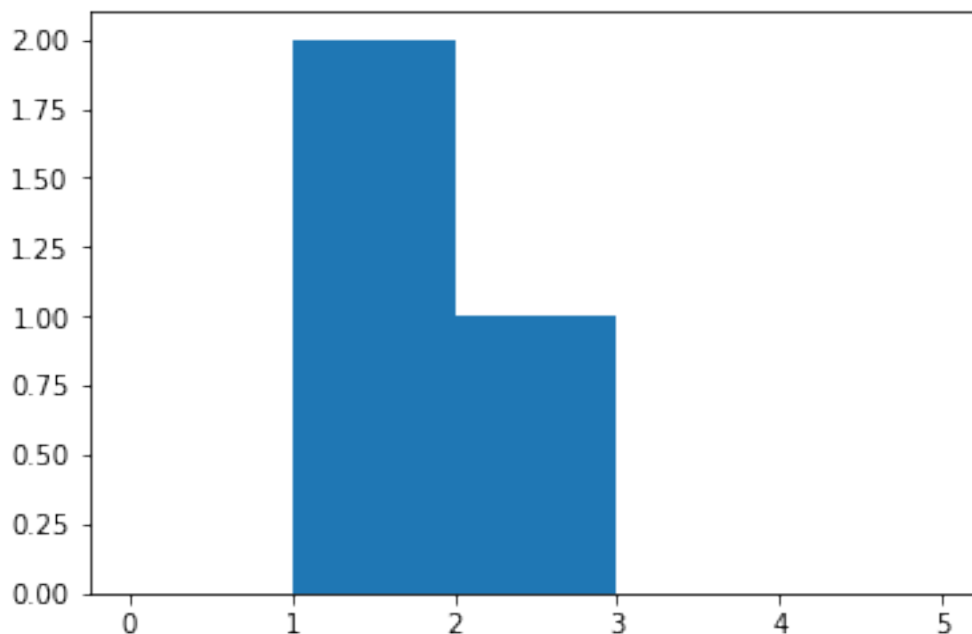
Q75. Write the NumPy program to remove the negative values in the numpy array with 0?

```
In [94]: arr = np.array([-1, -4, 0, 2, 3, 4, 5, -6])
arr[arr < 0] = 0
print(arr)
```

```
[0 0 0 2 3 4 5 0]
```

Q76. Write the NumPy program to compute the histogram of a set of data?

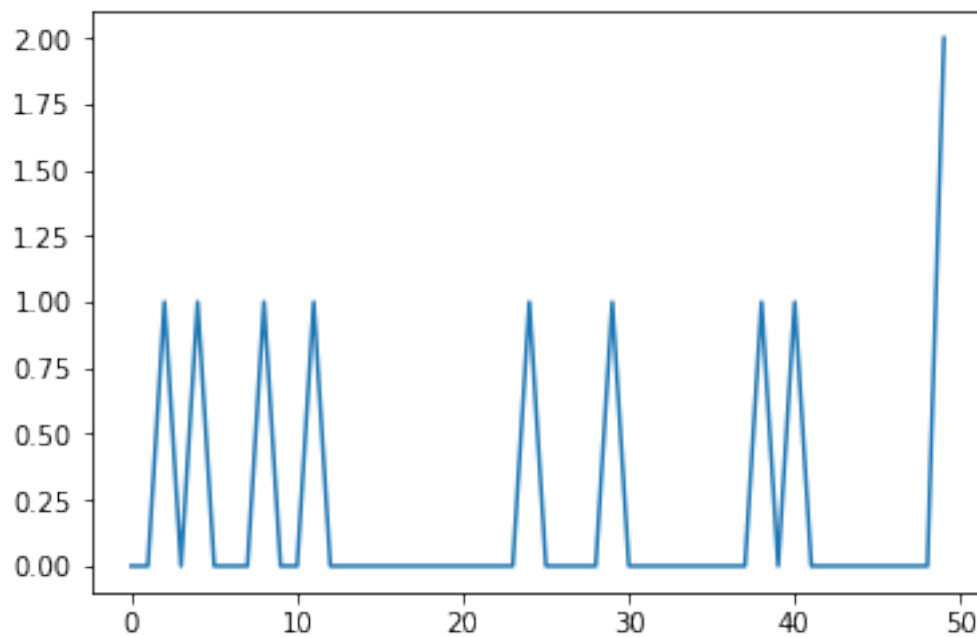
```
In [13]: import numpy as np
import matplotlib.pyplot as plt
plt.hist([1, 2, 1], bins=[0, 1, 2, 3, 5])
plt.show()
```



Q77. Write the NumPy program to compute the line graph of a set of data?

```
In [14]: import numpy as np
import matplotlib.pyplot as plt
arr = np.random.randint(1, 50, 10)
y, x = np.histogram(arr, bins=np.arange(51))
fig, ax = plt.subplots()
ax.plot(x[:-1], y)
fig.show()
```

C:\Apps\Continuum\anaconda3\lib\site-packages\matplotlib\figure.py:445: UserWarning: Matplotlib is not a GUI application. You may want to use the % get_backend() function to set the backend to a non-GUI backend.



Q78. Write the NumPy program to extracts all the elements from second row from given (4x4) array?

```
In [95]: d = np.array([[ 0, 1, 2, 3], [ 4, 5, 6, 7], [ 8, 9, 10, 11], [12, 13, 14, 15]])
d[1,:]
```

Out[95]: array([4, 5, 6, 7])

Q79. Write the NumPy program to extract first element of the second row and fourth element of fourth row from a given (4x4) array?

```
In [96]: d = np.array([[ 0, 1, 2, 3], [ 4, 5, 6, 7], [ 8, 9, 10, 11], [12, 13, 14, 15]])
[d[1,0],d[3,3]]
```

Out[96]: [4, 15]

Q80. Write the NumPy program to add two arrays A and B of sizes (3,3) and (,3)?

```
In [97]: import numpy as np
         n = np.empty((3, 3))
         n.fill(1)
         a = np.array([[0,1,2]])
         z = n + a
         z.astype(int)
```

```
Out[97]: array([[1, 2, 3],
               [1, 2, 3],
               [1, 2, 3]])
```

Q81. Write the NumPy program to copy data from a given array to another array?

```
In [98]: d = np.array([[ 24, 27, 30, 29, 18,14]])
         b = np.copy(d)
         print(d)
         print(b)
```

```
[[24 27 30 29 18 14]]
[[24 27 30 29 18 14]]
```

Q82. Write the NumPy program to calculate the sum of all columns of the 2D numpy array?

```
In [99]: d = np.array([[ 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],
                    np.sum(d,axis=0)
```

```
Out[99]: array([54, 58, 62, 66, 70, 74, 78, 82, 86])
```

Q83. Write the NumPy program to calculate averages without NaNs along the given array?

```
In [100]: s_array = np.array ([[10, 20, 30,], [40, 50, np.nan], [np.nan, 6, np.nan], [np.nan, 7, np.nan]])
         np.nanmean(s_array,axis=1)
```

C:\Apps\Continuum\anaconda3\lib\site-packages\ipykernel_launcher.py:2: RuntimeWarning: Mean of empty slice

```
Out[100]: array([20., 45., 6., nan])
```

Q84. Create two arrays of six elements. Write the NumPy program to count the number of instances of a value occurring in one array on the condition of another array.

```
In [3]: import numpy as np
         aa = np.array([10,-10,10,-10,-10,10])
         bb = np.array([.85,.45,.9,.8,.12,.6])
         print("Original arrays:")
         print(aa)
         print(bb)
         no_of_instance = np.sum((aa == 10) & (bb > .6))
         print("\nNumber of instances of a value occurring in one array on the condition of another array")
         print(no_of_instance)
```

Original arrays:

```
[ 10 -10  10 -10 -10  10]
[0.85 0.45 0.9  0.8  0.12 0.6 ]
```

Number of instances of a value occurring in one array on the condition of another array:
2

Q85. Write the NumPy program to convert a Python dictionary to a Numpy ndarray ?

```
In [101]: b = {'column0': {'a': 1, 'b': 0.0, 'c': 0.0, 'd': 2.0}, 'column1': {'a': 3.0, 'b': 1.0, 'c': 0.0, 'd': 0.0}}
          arr= np.array([list(i.values()) for i in b.values()])
          print((arr))

[[ 1.  0.  0.  2.]
 [ 3.  1.  0. -1.]
 [ 4.  1.  5. -1.]
 [ 3. -1. -1. -1.]]
```

Q86. Write the Numpy program to find and store the non-zero unique rows in an array after comparing each row with other row in the given matrix?

```
In [102]: a = np.array([[ 1, 1, 0], [ 0, 0, 0], [ 0, 2, 3], [ 0, 0, 0], [ 0, -1, 1], [ 0, 0, 0]])
          # get unique rows
          b = np.unique(a,axis=0)
          # remove rows with zero values
          b[~np.all(b == 0, axis=1)]

Out[102]: array([[ 0, -1,  1],
                 [ 0,  2,  3],
                 [ 1,  1,  0]])
```

Q87. Write the NumPy program to multiply the matrix by another matrix of complex numbers and create a new matrix of complex numbers?

```
In [103]: first_array= np.array([ 1.+2.j, 3.+4.j])
          second_array=np.array([ 5.+6.j, 7.+8.j])
          final_array = np.vdot(first_array, second_array)
          final_array

Out[103]: (70-8j)
```

Q88. Write a NumPy program to generate the matrix product of two Arrays?

```
In [104]: x= np.array([[1, 0], [1, 1]])
          y= np.array([[3, 1], [2, 2]])
          np.matmul(x,y)

Out[104]: array([[3, 1],
                 [5, 3]])
```

Q89. Write the NumPy program to find roots of the following Polynomials?

```
In [4]: import numpy as np
        print("Roots of the first polynomial:")
        print(np.roots([1, -2, 1]))
        print("Roots of the second polynomial:")
        print(np.roots([1, -12, 10, 7, -10]))
```

Roots of the first polynomial:

[1. 1.]

Roots of the second polynomial:

[11.04461946+0.j -0.8711421 +0.j 0.91326132+0.4531004j
0.91326132-0.4531004j]

Q90. Write the NumPy program to calculate inverse of sine, cosine, and inverse tangent for all elements in a given array?

```
In [105]: import numpy as np
          x = np.array([1, 0, -1])
          y = np.array([-1, 0, 1])
          print("Inverse sine:", np.arcsin(x))
          print("Inverse cosine:", np.arccos(y))
          print("Inverse tangent:", np.arctan(x))
```

Inverse sine: [1.57079633 0. -1.57079633]

Inverse cosine: [3.14159265 1.57079633 0.]

Inverse tangent: [0.78539816 0. -0.78539816]

Q91. Write the NumPy program to calculate the difference between in neighbouring elements, element-wise of a given array?

```
In [106]: ss = np.array([1, 3, 5, 7, 0])
          np.diff(ss)
```

```
Out[106]: array([ 2,  2,  2, -7])
```

Q92. Write the Python program to find the maximum and the minimum value of a given flattened array?

```
In [107]: n = np.array([[0, 1], [2, 3]])
          print(np.amax(n))
          print(np.amin(n))
```

3

0

Q93. Write the NumPy program to calculate the difference between in the maximum and the minimum values of a given array along the second axis ?

```
In [108]: a = np.array([[0,1,2,3,4,5], [6,7,8,9,10,11]])
          print(np.ptp(a,1))
```

[5 5]

Q94. Write the NumPy program to compute the weighted of the given array ?

```
In [110]: import numpy as np
          x = np.array([0,1,2,3,4])
          print("\nOriginal array:")
          print(x)
          weights = np.arange(1, len(x)+1)
          avg = np.average(x, weights=weights)
          print(avg)
```

Original array:

[0 1 2 3 4]

2.6666666666666665

Q95. Write the NumPy program to compute the mean, standard deviation, and the variance of a given array along the second axis?

```
In [111]: import numpy as np
          x = np.array([0, 1, 2, 3, 4, 5])
          m = np.mean(x)
          s = np.std(x)
          v = np.var(x)

          print ('mean:',m)
          print ('standard deviation:',s)
          print ('variance:',v)
```

mean: 2.5

standard deviation: 1.707825127659933

variance: 2.9166666666666665

Q96. Write the Numpy program to compute the covariance matrix of the two given arrays?

```
In [112]: import numpy as np
          a = np.array ([[0, 1, 2],[2,1,0]])
          np.cov(a)
```

```
Out[112]: array([[ 1., -1.],
                 [-1.,  1.]])
```

Q97. Write a NumPy program to compute the cross-correlation of two given arrays ?


```
In [10]: import numpy as np
         a = np.array ([0, 1, 3])
         b = np.array([2,4,5])
         print('Original arrays:')
         print(a)
         print(b)
         print('Cross correlation of above arrays:')
         np.cov(a,b)
```

Original arrays:

```
[0 1 3]
```

```
[2 4 5]
```

Cross correlation of above arrays:

```
Out[10]: array([[2.33333333, 2.16666667],
                [2.16666667, 2.33333333]])
```

Q98. Write the NumPy program to compute Pearson product-moment correlation coefficients of two given arrays?

```
In [15]: import numpy as np
         x = np.array([0, 1, 3])
         y = np.array([2, 4, 5])
         print("\nOriginal array1:")
         print(x)
         print("\nOriginal array1:")
         print(y)
         print("\nPearson product-moment correlation coefficients of above arrays:\n",np.corrcoef(x,y))
```

Original array1:

```
[0 1 3]
```

Original array1:

```
[2 4 5]
```

Pearson product-moment correlation coefficients of above arrays:

```
[[1.          0.92857143]
 [0.92857143 1.          ]]
```

Q99. Write the python program to count the number of occurrences of each value in a given array of non-negative integers?

```
In [16]: import numpy as np
         array1 = [0, 1, 6, 1, 4, 1, 2, 2, 7]
         print("Original array:")
         print(array1)
         print("Number of occurrences of each value in array: ")
         print(np.bincount(array1))
```

Original array:

```
[0, 1, 6, 1, 4, 1, 2, 2, 7]
```

Number of occurrences of each value in array:

```
[1 3 2 0 1 0 1 1]
```

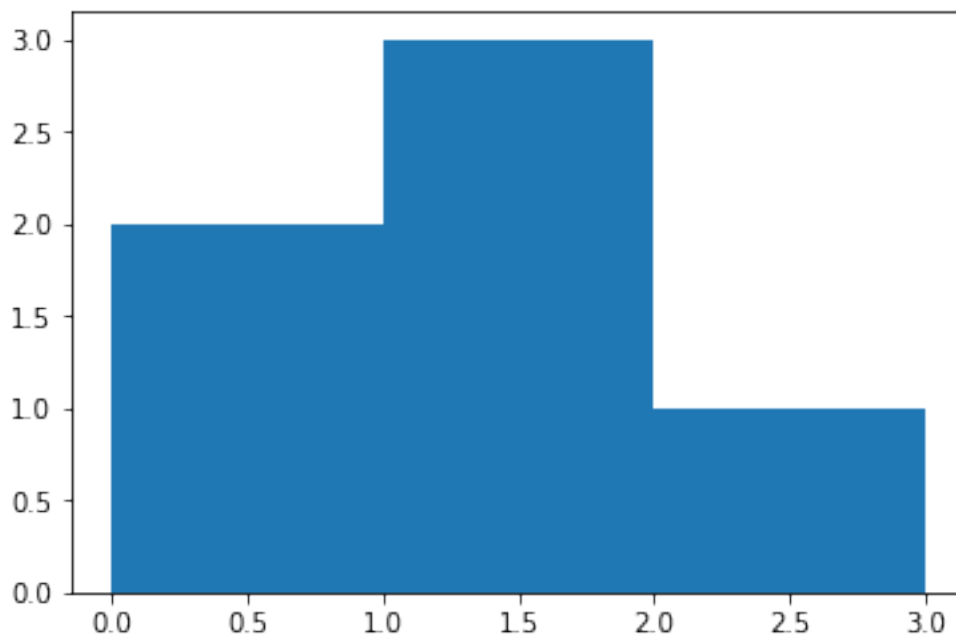
Q100. Write a Numpy program to compute the histogram of nums against the bins?

```
In [18]: import numpy as np
import matplotlib.pyplot as plt
nums = np.array([0.5, 0.7, 1.0, 1.2, 1.3, 2.1])
bins = np.array([0, 1, 2, 3])
print("nums: ",nums)
print("bins: ",bins)
print("Result:", np.histogram(nums, bins))
plt.hist(nums, bins=bins)
plt.show()
```

```
nums: [0.5 0.7 1.  1.2 1.3 2.1]
```

```
bins: [0 1 2 3]
```

```
Result: (array([2, 3, 1], dtype=int64), array([0, 1, 2, 3]))
```



Q101. Write the Python program to add, subtract, multiply and divide two pandas series ?

```
In [113]: import pandas as pd
a = pd.Series([2, 4, 6, 8, 10])
```

```

b = pd.Series([1, 3, 5, 7, 9])
print('addition:',a+b)
print('subtraction:',a-b)
print('multiplication:',a*b)
print('division:',a/b)
#print(type(a))
#print(type(b))

```

```

addition: 0      3
1      7
2     11
3     15
4     19
dtype: int64
subtraction: 0      1
1      1
2      1
3      1
4      1
dtype: int64
multiplication: 0      2
1     12
2     30
3     56
4     90
dtype: int64
division: 0      2.000000
1     1.333333
2     1.200000
3     1.142857
4     1.111111
dtype: float64

```

Q102. Write a Python program to convert a dictionary to the Pandas Series?

```

In [114]: import pandas as pd
dict = {'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 800}
s = pd.Series(dict)
print(s)

```

```

a      100
b      200
c      300
d      400
e      800
dtype: int64

```

Q103. Write a python program to change the data type of given a column or a Series?

```
In [115]: import pandas as pd
          s = pd.Series([100,200,'python',300.12,400])
          n = pd.to_numeric(s,errors='coerce')
          print(n)

0    100.00
1    200.00
2         NaN
3    300.12
4    400.00
dtype: float64
```

Q104. Write the python pandas program to convert the first column of a DataFrame as a Series?

```
In [116]: df = pd.DataFrame({'col1': [1, 2,3,4,7,11], 'col2': [4,5,6,9,5,0], 'col3': [7,5,8,12,9,6]})
          a = df.col1
          #s = pd.Series(df)
          print(type(a))
          print(a)

<class 'pandas.core.series.Series'>
0     1
1     2
2     3
3     4
4     7
5    11
Name: col1, dtype: int64
```

Q105. Write a pandas program to create the mean and standard deviation of the data of a given Series?

```
In [117]: import pandas as pd
          a = pd.Series([1,2,3,4,5,6,7,8,9,5,3])
          print('mean:',a.mean())
          print('standard deviation:',a.std())

mean: 4.818181818181818
standard deviation: 2.522624895547565
```

Q106. Write a pandas program to get powers of an array values element-wise?

```
In [118]: import pandas as pd
          df = pd.DataFrame({'X':[78,85,96,80,86], 'Y':[84,94,89,83,86], 'Z':[86,97,96,72,83]})
          print(df)
```

	X	Y	Z
0	78	84	86
1	85	94	97
2	96	89	96
3	80	83	72
4	86	86	83

Q107. Write the pandas program to get the first 3 rows of a given DataFrame?

```
In [119]: import pandas as pd
exam_data = pd.DataFrame ({'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily'],
                           'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
                           'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
                           'qualify': [ 'yes', 'no', 'yes' , 'no', ' no ', ' yes ', 'yes', 'no', 'no', 'no'],
                           index = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j' ] })
exam_data.head(3)
```

```
Out[119]:
```

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes

Q108: Write the pandas program to select the specified columns and the rows from a given data frame?

```
In [120]: import pandas as pd

exam_data = pd.DataFrame ({'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily'],
                           'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
                           'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
                           'qualify': [ 'yes', 'no', 'yes' , 'no', ' no ', ' yes ', 'yes', 'no', 'no', 'no'],
                           index = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j' ] })
exam_data.loc[['b','d','f','g'],['name','score']]
```

```
Out[120]:
```

	name	score
b	Dima	9.0
d	James	NaN
f	Michael	20.0
g	Matthew	14.5

Q109. Write the pandas program to calculate mean score for each different student in DataFrame?

```
In [121]: import pandas as pd

exam_data = pd.DataFrame ({'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily'],
                           'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
                           'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
```

```

        'qualify': [ 'yes', 'no', 'yes' , 'no', ' no ', ' yes ', 'yes', 'no', 'no' ],
        index = [ 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j' ]
    exam_data.loc[:, "score"].mean()

```

Out[121]: 13.5625

Q110. Write the Pandas program to rename columns of a given DataFrame ?

```

In [122]: sample_data = pd.DataFrame ({'col1': [1,2,3],
        'col2': [4,5,6],
        'col3': [7,8,9]})
sample_data = sample_data.rename(columns={"col1": "Column1", "col2": "Column2", "col3": "Column3"})
sample_data

```

```

Out[122]:
   Column1  Column2  Column3
0         1         4         7
1         2         5         8
2         3         6         9

```

Q111. Write a pandas program to count city-wise number of people from a given of data set (city, name of the person)?

```

In [123]: import pandas as pd
df = pd.DataFrame({'name': ['Vijay', 'Raaghav', 'Viraan', 'Virat', 'Raj', 'Johnson'],
        'city': ['California', 'Los Angeles', 'California', 'California', 'California', 'Los Angeles'],
        'cnt': [4, 2, 4, 2, 2, 2]})
Cnt = df.groupby(["city"]).size().reset_index(name='Number of people')
print(Cnt)

```

```

   city  Number of people
0  California             4
1   Georgia             2
2 Los Angeles             4

```

Q112. Write a pandas program to widen output display to see more columns?

```

In [124]: data = pd.DataFrame ({'col1': [1,4,3,4,5],
        'col2': [4,5,6,7,8],
        'col3': [7,8,9,0,1]})

pd.set_option('display.max_columns', 50)

data

```

```

Out[124]:
   col1  col2  col3
0     1     4     7
1     4     5     8
2     3     6     9
3     4     7     0
4     5     8     1

```

Q113. Write a pandas program to convert the data frame column type from string to DateTime?

```
In [127]: from datetime import datetime
st = pd.DataFrame({'Dt': ['3/11/2000', '3/12/2000', '3/13/2000']})
st['Dt'] = pd.to_datetime(st['Dt'], format='%m/%d/%Y')
st
```

```
Out[127]:      Dt
0 2000-03-11
1 2000-03-12
2 2000-03-13
```

Q114. Write a pandas program to append the data to an empty DataFrame?

```
In [126]: # creating empty dataframe
data = pd.DataFrame(columns=['col1', 'col2'])
# appending sample data
data = data.append({'col1': 0, 'col2': 0, }, ignore_index=True)
data = data.append({'col1': 1, 'col2': 1, }, ignore_index=True)
data = data.append({'col1': 2, 'col2': 2, }, ignore_index=True)
data
```

```
Out[126]:   col1 col2
0      0     0
1      1     1
2      2     2
```

Q115. Write a pandas program to count the number of columns of a DataFrame?

```
In [128]: data = pd.DataFrame({'col1': [1,4,3,4,7],
                              'col2': [4,5,6,9,5],
                              'col3': [7,8,12,1,11]})
print('No.of columns:', len(data.columns))
```

No.of columns: 3

Q116. Write a Pandas program to remove the last n rows of a given DataFrame ?

```
In [129]: data = pd.DataFrame({'col1': [1,2,3,4,7,11],
                              'col2': [4,5,6,9,5,0],
                              'col3': [7,5,8,12,1,11]})
data1 = data.iloc[:3]
print(data1)
```

```
   col1  col2  col3
0      1     4     7
1      2     5     5
2      3     6     8
```

Q117. Write a Pandas program to import excel data (coalpublic2013.xlsx) into a Pandas data frame.

```
In [130]: #(https://drive.google.com/drive/folders/105ftuIwN9kqyPNEEm3E6IM7LqywjyvJa?usp=sharing)
# above google drive link contains 2 excel workbooks (coalpublic2013.xlsx with one sheet
# and employee.xlsx with 3 sheets - Sheet1,Sheet2 and Sheet3)
import pandas as pd
import os
os.chdir('C:/Users/S405720/Desktop/Assignment')
print (os.getcwd())
pb = pd.read_excel("coalpublic2013.xlsx")
```

C:\Users\S405720\Desktop\Assignment

Q118. Write a Pandas program to import excel data (coalpublic2013.xlsx) into a data frame and find details where “Mine Name” starts with "P".

```
In [131]: pb[pb.Mine_Name.str.startswith('P')]
```

```
Out[131]:
```

	Year	MSHA ID	Mine_Name	Production	Labor_Hours
13	2013	103332	Powhatan Mine	140521	61394
18	2013	102976	Piney Woods Preparation Plant	0	14828
19	2013	102976	Piney Woods Preparation Plant	0	23193
46	2013	103321	Poplar Springs	189370	76366

Q119. Write a Pandas program to import excel data (employee.xlsx) into a Pandas dataframe and find the list of employees where hire_date> 01-01-07.

```
In [132]: emp = pd.read_excel("employee.xlsx",sheet_name = ['Sheet1','Sheet2','Sheet3'])
# concatenating multiple sheets in to one dataframe
emp1 = pd.concat(emp[i] for i in emp.keys())
```

```
In [133]: emp1[emp1.hire_date>'2007-01-01']
```

```
Out[133]:
```

	emp_id	first_name	last_name	hire_date
4	104	Bruce	Ernst	2007-05-21
7	107	Diana	Lorentz	2007-02-07
13	113	Luis	Popp	2007-12-07
19	119	Karen	Colmenares	2007-08-10
4	124	Kevin	Mourgos	2007-11-16
7	127	James	Landry	2007-01-14
8	128	Steven	Markle	2008-03-08
12	132	TJ	Olson	2007-04-10
15	135	Ki	Gee	2007-12-12
16	136	Hazel	Philtanker	2008-02-06
7	148	Gerald	Cambrault	2007-10-15
8	149	Eleni	Zlotkey	2008-01-29

Q120. Write a Pandas program to import excel data (employee.xlsx) into a Pandas dataframe and find a list of the employees of a specified year

```
In [134]: #emp.dtypes
emp1[emp1['hire_date'].dt.year==2007]
```

```
Out[134]:
```

	emp_id	first_name	last_name	hire_date
4	104	Bruce	Ernst	2007-05-21
7	107	Diana	Lorentz	2007-02-07
13	113	Luis	Popp	2007-12-07
19	119	Karen	Colmenares	2007-08-10
4	124	Kevin	Mourgoss	2007-11-16
7	127	James	Landry	2007-01-14
12	132	TJ	Olson	2007-04-10
15	135	Ki	Gee	2007-12-12
7	148	Gerald	Cambrault	2007-10-15

Q121. Write a pandas program to import three datasheets from a given excel data (coalpublic2013.xlsx) in to a single dataframe.

```
In [135]: # created excel workbook with 3 sheets (dataset1,dataset2 and dataset3)
pb1 = pd.read_excel('coalpublic2013_X.xlsx',sheet_name=['dataset1','dataset2','dataset3'])
pb2 = pd.concat(pb1[i] for i in pb1.keys())
```

```
In [136]: pb2
```

```
Out[136]:
```

	Year	MSHA ID	Mine_Name	Production	Labor_Hours
0	2013	103381	Tacoa Highwall Miner	56004	22392
1	2013	103404	Reid School Mine	28807	28447
2	2013	100759	North River #1 Underground Min	1440115	474784
3	2013	103246	Bear Creek	87587	29193
4	2013	103451	Knight Mine	147499	46393
5	2013	103433	Crane Central Mine	69339	47195
6	2013	100329	Concord Mine	0	144002
7	2013	100851	Oak Grove Mine	2269014	1001809
8	2013	102901	Shoal Creek Mine	0	12396
9	2013	102901	Shoal Creek Mine	1453024	1237415
10	2013	103180	Sloan Mountain Mine	327780	196963
11	2013	103182	Fishtrap	175058	87314
12	2013	103285	Narley Mine	154861	90584
13	2013	103332	Powhatan Mine	140521	61394
14	2013	103375	Johnson Mine	580	1900
15	2013	103419	Maxine-Pratt Mine	125824	107469
16	2013	103432	Skelton Creek	8252	220
17	2013	103437	Black Warrior Mine No 1	145924	70926
18	2013	102976	Piney Woods Preparation Plant	0	14828
0	2013	102976	Piney Woods Preparation Plant	0	23193
1	2013	103380	Calera	0	12621
2	2013	103380	Calera	0	1402

3	2013	103422	Clark No 1 Mine	122727	140250
4	2013	103467	Helena Surface Mine	59664	30539
5	2013	101247	No 4 Mine	2622528	1551141
6	2013	101401	No 7 Mine	5405412	2464719
7	2013	103172	Searles Mine No. 2, 3, 4, 5, 6	258078	119542
8	2013	103179	Fleetwood Mine No 1	75937	63745
9	2013	103303	Shannon Mine	317491	164388
10	2013	103323	Deerlick Mine	133452	46381
11	2013	103364	Brc Alabama No. 7 Llc	0	14324
12	2013	103436	Swann's Crossing	137511	77190
13	2013	100347	Choctaw Mine	537429	215295
14	2013	101362	Manchester Mine	219457	116914
15	2013	102996	Jap Creek Mine	375715	164093
16	2013	103155	Corinth Prep Plant	0	27996
17	2013	103155	Corinth Prep Plant	0	51994
18	2013	103195	Mccollum/Sparks Branch Mine	71910	17411
19	2013	103342	Reese's Branch Mine	263888	115123
0	2013	103370	Crescent Valley Mine	2860	621
1	2013	103372	Cane Creek Mine	66258	32401
2	2013	103376	Town Creek	299167	176499
3	2013	103389	Carbon Hill Mine	76241	84966
4	2013	103410	Coal Valley Mine	407841	158591
5	2013	103423	Dutton Hill Mine	37275	9162
6	2013	1519322	Ghm #25	25054	3108
7	2013	103321	Poplar Springs	189370	76366
8	2013	103358	Old Union	284563	161805
9	2013	5000030	Usibelli	1631584	286079
10	2013	201195	Kayenta Mine	7602722	1015333

Q 122. Write a pandas program to import three datasheets from a given excel data (employee.xlsx) into a single data frame and export the result into new Excel file.

```
In [137]: emp2 = pd.read_excel("employee.xlsx",sheet_name = ['Sheet1','Sheet2','Sheet3'])
# concatenating multiple sheets in to one dataframe
emp3 = pd.concat(emp2[i] for i in emp2.keys())
```

```
In [138]: # use excelwriter to store the dataframe in to new excel called 'all_employee_list.xlsx'
w = pd.ExcelWriter('all_employee_list.xlsx',engine='xlsxwriter')
emp3.to_excel(w)
w.save()
```

Q123. Write a pandas program to create the Pivot table with multiple indexes from the data set of the titanic.csv.

```
In [139]: import pandas as pd
import os
os.chdir('C:/Users/S405720/Desktop/Assignment')
print (os.getcwd())
titanic = pd.read_csv("titanic.csv")
```

C:\Users\S405720\Desktop\Assignment

```
In [15]: # This table shows the average age of survived across sex and class
pd.pivot_table(titanic, index=["survived", "sex"], values=["age"], columns=["class"])
```

```
Out[15]:
```

		age		
class		First	Second	Third
0	female	25.666667	36.000000	23.818182
	male	44.581967	33.369048	27.255814
1	female	34.939024	28.080882	19.329787
	male	36.248000	16.022000	22.274211

Q124. Write a Pandas program to create the Pivot table and find survival rate by gender?

```
In [140]: t = pd.pivot_table(titanic, index=["survived", "sex"], values=["who"], aggfunc=len)
t['% of Total'] = round((t.who / t.who.sum() * 100)).astype(str) + '%'
t['% of Survived'] = round(t.who / t.groupby(level=0).who.transform(sum) * 100).astype(str)
t
```

```
Out[140]:
```

		who	% of Total	% of Survived
0	female	81	9.0%	15.0%
	male	468	53.0%	85.0%
1	female	233	26.0%	68.0%
	male	109	12.0%	32.0%

Q125. Write a pandas program to make partition each of the passengers into 4 categories based on their age.

```
In [141]: # updated 0 age column where value is Nan
titanic['age'].fillna(0, inplace=True)
```

```
In [142]: # Create a function to assign age bucket for each record
age_bkt = []
for row in titanic['age']:
    if (row >= 0 and row <=10):    age_bkt.append('0-10')
    elif (row > 10 and row <=30):  age_bkt.append('11-30')
    elif (row > 30 and row <=60):  age_bkt.append('31-60')
    elif (row > 60 and row <=80):  age_bkt.append('61-80')
    else:
        age_bkt.append('Failed')
    # Create a column from the list
titanic['age_bucket'] = age_bkt
```

```
In [143]: titanic
```

```

Out[143]:
  survived  pclass   sex  age  sibsp  parch   fare embarked  class \
0         0      3  male  22.0    1     0   7.2500         S   Third
1         1      1 female  38.0    1     0  71.2833         C   First
2         1      3 female  26.0    0     0   7.9250         S   Third
3         1      1 female  35.0    1     0  53.1000         S   First
4         0      3  male  35.0    0     0   8.0500         S   Third
5         0      3  male   0.0    0     0   8.4583         Q   Third
6         0      1  male  54.0    0     0  51.8625         S   First
7         0      3  male   2.0    3     1  21.0750         S   Third
8         1      3 female  27.0    0     2  11.1333         S   Third
9         1      2 female  14.0    1     0  30.0708         C  Second
10        1      3 female   4.0    1     1  16.7000         S   Third
11        1      1 female  58.0    0     0  26.5500         S   First
12        0      3  male  20.0    0     0   8.0500         S   Third
13        0      3  male  39.0    1     5  31.2750         S   Third
14        0      3 female  14.0    0     0   7.8542         S   Third
15        1      2 female  55.0    0     0  16.0000         S  Second
16        0      3  male   2.0    4     1  29.1250         Q   Third
17        1      2  male   0.0    0     0  13.0000         S  Second
18        0      3 female  31.0    1     0  18.0000         S   Third
19        1      3 female   0.0    0     0   7.2250         C   Third
20        0      2  male  35.0    0     0  26.0000         S  Second
21        1      2  male  34.0    0     0  13.0000         S  Second
22        1      3 female  15.0    0     0   8.0292         Q   Third
23        1      1  male  28.0    0     0  35.5000         S   First
24        0      3 female   8.0    3     1  21.0750         S   Third
25        1      3 female  38.0    1     5  31.3875         S   Third
26        0      3  male   0.0    0     0   7.2250         C   Third
27        0      1  male  19.0    3     2 263.0000         S   First
28        1      3 female   0.0    0     0   7.8792         Q   Third
29        0      3  male   0.0    0     0   7.8958         S   Third
..        ...      ...      ...      ...      ...      ...
861       0      2  male  21.0    1     0  11.5000         S  Second
862       1      1 female  48.0    0     0  25.9292         S   First
863       0      3 female   0.0    8     2  69.5500         S   Third
864       0      2  male  24.0    0     0  13.0000         S  Second
865       1      2 female  42.0    0     0  13.0000         S  Second
866       1      2 female  27.0    1     0  13.8583         C  Second
867       0      1  male  31.0    0     0  50.4958         S   First
868       0      3  male   0.0    0     0   9.5000         S   Third
869       1      3  male   4.0    1     1  11.1333         S   Third
870       0      3  male  26.0    0     0   7.8958         S   Third
871       1      1 female  47.0    1     1  52.5542         S   First
872       0      1  male  33.0    0     0   5.0000         S   First
873       0      3  male  47.0    0     0   9.0000         S   Third
874       1      2 female  28.0    1     0  24.0000         C  Second
875       1      3 female  15.0    0     0   7.2250         C   Third
876       0      3  male  20.0    0     0   9.8458         S   Third

```

877	0	3	male	19.0	0	0	7.8958	S	Third
878	0	3	male	0.0	0	0	7.8958	S	Third
879	1	1	female	56.0	0	1	83.1583	C	First
880	1	2	female	25.0	0	1	26.0000	S	Second
881	0	3	male	33.0	0	0	7.8958	S	Third
882	0	3	female	22.0	0	0	10.5167	S	Third
883	0	2	male	28.0	0	0	10.5000	S	Second
884	0	3	male	25.0	0	0	7.0500	S	Third
885	0	3	female	39.0	0	5	29.1250	Q	Third
886	0	2	male	27.0	0	0	13.0000	S	Second
887	1	1	female	19.0	0	0	30.0000	S	First
888	0	3	female	0.0	1	2	23.4500	S	Third
889	1	1	male	26.0	0	0	30.0000	C	First
890	0	3	male	32.0	0	0	7.7500	Q	Third

	who	adult_male	deck	embark_town	alive	alone	age_bucket
0	man	True	NaN	Southampton	no	False	11-30
1	woman	False	C	Cherbourg	yes	False	31-60
2	woman	False	NaN	Southampton	yes	True	11-30
3	woman	False	C	Southampton	yes	False	31-60
4	man	True	NaN	Southampton	no	True	31-60
5	man	True	NaN	Queenstown	no	True	0-10
6	man	True	E	Southampton	no	True	31-60
7	child	False	NaN	Southampton	no	False	0-10
8	woman	False	NaN	Southampton	yes	False	11-30
9	child	False	NaN	Cherbourg	yes	False	11-30
10	child	False	G	Southampton	yes	False	0-10
11	woman	False	C	Southampton	yes	True	31-60
12	man	True	NaN	Southampton	no	True	11-30
13	man	True	NaN	Southampton	no	False	31-60
14	child	False	NaN	Southampton	no	True	11-30
15	woman	False	NaN	Southampton	yes	True	31-60
16	child	False	NaN	Queenstown	no	False	0-10
17	man	True	NaN	Southampton	yes	True	0-10
18	woman	False	NaN	Southampton	no	False	31-60
19	woman	False	NaN	Cherbourg	yes	True	0-10
20	man	True	NaN	Southampton	no	True	31-60
21	man	True	D	Southampton	yes	True	31-60
22	child	False	NaN	Queenstown	yes	True	11-30
23	man	True	A	Southampton	yes	True	11-30
24	child	False	NaN	Southampton	no	False	0-10
25	woman	False	NaN	Southampton	yes	False	31-60
26	man	True	NaN	Cherbourg	no	True	0-10
27	man	True	C	Southampton	no	False	11-30
28	woman	False	NaN	Queenstown	yes	True	0-10
29	man	True	NaN	Southampton	no	True	0-10
..
861	man	True	NaN	Southampton	no	False	11-30

862	woman	False	D	Southampton	yes	True	31-60
863	woman	False	NaN	Southampton	no	False	0-10
864	man	True	NaN	Southampton	no	True	11-30
865	woman	False	NaN	Southampton	yes	True	31-60
866	woman	False	NaN	Cherbourg	yes	False	11-30
867	man	True	A	Southampton	no	True	31-60
868	man	True	NaN	Southampton	no	True	0-10
869	child	False	NaN	Southampton	yes	False	0-10
870	man	True	NaN	Southampton	no	True	11-30
871	woman	False	D	Southampton	yes	False	31-60
872	man	True	B	Southampton	no	True	31-60
873	man	True	NaN	Southampton	no	True	31-60
874	woman	False	NaN	Cherbourg	yes	False	11-30
875	child	False	NaN	Cherbourg	yes	True	11-30
876	man	True	NaN	Southampton	no	True	11-30
877	man	True	NaN	Southampton	no	True	11-30
878	man	True	NaN	Southampton	no	True	0-10
879	woman	False	C	Cherbourg	yes	False	31-60
880	woman	False	NaN	Southampton	yes	False	11-30
881	man	True	NaN	Southampton	no	True	31-60
882	woman	False	NaN	Southampton	no	True	11-30
883	man	True	NaN	Southampton	no	True	11-30
884	man	True	NaN	Southampton	no	True	11-30
885	woman	False	NaN	Queenstown	no	False	31-60
886	man	True	NaN	Southampton	no	True	11-30
887	woman	False	B	Southampton	yes	True	11-30
888	woman	False	NaN	Southampton	no	False	0-10
889	man	True	C	Cherbourg	yes	True	11-30
890	man	True	NaN	Queenstown	no	True	31-60

[891 rows x 16 columns]

Q126. Write a pandas program to create the Pivot table and find survival rate by the gender, age of the different categories of various classes.

```
In [144]: t = titanic.pivot_table('survived', index=["sex", "age_bucket"], columns='class')
t
```

```
Out[144]: class          First    Second    Third
sex  age_bucket
female 0-10          0.900000    1.000000    0.562500
       11-30          0.970588    0.916667    0.508772
       31-60          0.979167    0.900000    0.272727
       61-80          1.000000         NaN    1.000000
male   0-10          0.304348    0.611111    0.146552
       11-30          0.458333    0.023256    0.139073
       31-60          0.412698    0.090909    0.118421
       61-80          0.083333    0.333333    0.000000
```

Q127. Write a pandas program to create the Pivot table and calculate number of women and men were in a particular cabin class.

```
In [145]: t = pd.pivot_table(titanic, index=["who"], columns=["class"], values=["survived"], aggfunc=
#t.index.name = None
t
```

```
Out[145]:
```

	survived		
class	First	Second	Third
who			
child	6	19	58
man	119	99	319
woman	91	66	114

Q128. Write a pandas program to create the Pivot table and separate the gender according to whether they travelled alone or not to get the probability of survival

```
In [146]: import pandas as pd
import numpy as np
titanic = pd.read_csv('titanic.csv')
xx = titanic.pivot_table( 'survived' , [ 'sex' , 'alone' ] , 'class' )
print(xx)
```

		First	Second	Third
sex	alone			
female	False	0.966667	0.931818	0.416667
	True	0.970588	0.906250	0.616667
male	False	0.425532	0.277778	0.180723
	True	0.333333	0.097222	0.121212

Q129. Write a pandas program to create the Pivot table and find the probability of survival by class, gender, solo boarding, and the port of embarkation.

```
In [24]: import pandas as pd
import numpy as np
titanic = pd.read_csv('titanic.csv')
xx = titanic.pivot_table( 'survived' , [ 'sex' , 'alone' ] , [ 'embark_town', 'class' ]
print(xx)
```

		Cherbourg			Queenstown			
		First	Second	Third	First	Second	Third	
sex	alone							
female	False	1.000000	1.000000	0.611111	1.0	NaN	0.625000	
	True	0.944444	1.000000	0.800000	NaN	1.0	0.760000	
male	False	0.473684	0.166667	0.500000	0.0	NaN	0.100000	
	True	0.347826	0.250000	0.151515	NaN	0.0	0.068966	

embark_town Southampton

class		First	Second	Third
sex	alone			
female	False	0.941176	0.923077	0.327586
	True	1.000000	0.892857	0.466667
male	False	0.407407	0.300000	0.142857
	True	0.326923	0.089552	0.123762

Q130. Write a pandas program to get current date, oldest date and number of days between Current date and the oldest date of Ufo dataset.

```
In [21]: import pandas as pd
         from datetime import datetime as dt
         import os
         os.chdir('C:/Users/S405720/Desktop/Assignment')
         df = pd.read_csv('ufo.csv')

         df['Date_time'] = pd.to_datetime(df.Date_time.str.replace('24:', '00:'))
         print("Original Dataframe:")
         print(df.head())
         print("\nCurrent date of Ufo dataset:")
         print(df.Date_time.max())
         print("\nOldest date of Ufo dataset:")
         print(df.Date_time.min())
         print((df.Date_time.max() - df.Date_time.min()).days)

         #df["Date_time"] = pd.to_datetime(df["Date_time"]).dt.strftime("%m-%d-%Y %H:%M:%S")
         #df['Date_time'] = dt.strptime(df['Date_time'], '%m/%d/%y %H:%M:%S')#df['Date_time'].
         #df['Date_time'] = df['Date_time'].apply(lambda x: dt.strptime(x, '%m/%d/%Y %H:%M'))
```

C:\Apps\Continuum\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2728: DtypeWarning:
interactivity=interactivity, compiler=compiler, result=result)

Original Dataframe:

	Date_time	city	state/province	country	UFO_shape	\
0	1949-10-10 20:30:00	san marcos	tx	us	cylinder	
1	1949-10-10 21:00:00	lackland afb	tx	NaN	light	
2	1955-10-10 17:00:00	chester	(uk/england)	NaN	gb	circle
3	1956-10-10 21:00:00	edna	tx	us	circle	
4	1960-10-10 20:00:00	kaneohe	hi	us	light	

	length_of_encounter_seconds	described_duration_of_encounter	\
0	2700	45 minutes	
1	7200	1-2 hrs	
2	20	20 seconds	
3	20	1/2 hour	
4	900	15 minutes	

	description	date_documented	\
0	This event took place in early fall around 194...	4/27/2004	
1	1949 Lackland AFB, TX. Lights racing across...	12/16/2005	
2	Green/Orange circular disc over Chester, En...	1/21/2008	
3	My older brother and twin sister were leaving ...	1/17/2004	
4	AS a Marine 1st Lt. flying an FJ4B fighter/att...	1/22/2004	

	latitude	longitude
0	29.8830556	-97.941111
1	29.38421	-98.581082
2	53.2	-2.916667
3	28.9783333	-96.645833
4	21.4180556	-157.803611

Current date of Ufo dataset:
2014-06-06 00:00:00

Oldest date of Ufo dataset:
1906-11-11 00:00:00
39289

Q131. Write a pandas program to get all sighting days of the unidentified flying object (ufo) between 1950-10-10 and 1960-10-10.

```
In [22]: sighting_period = df[(df['Date_time'] >= '1950-01-01 00:00:00') & (df['Date_time'] <=
print(sighting_period)
```

	Date_time	city	\
2	1955-10-10 17:00:00	chester (uk/england)	
3	1956-10-10 21:00:00	edna	
4	1960-10-10 20:00:00	kaneohe	
479	1950-10-01 11:00:00	pikeville	
480	1952-10-01 03:30:00	fukuoka (japan)	
481	1952-10-01 12:00:00	kansas city	
482	1954-10-01 19:00:00	flatwoods	
483	1955-10-01 22:00:00	mono lake (area)	
484	1956-10-01 19:00:00	newport	
485	1957-10-01 16:00:00	nelchina (near)	
486	1957-10-01 21:00:00	paterson	
487	1957-10-01 22:00:00	northeast cape afs	
488	1959-10-01 19:00:00	jacksonville	
1707	1950-10-15 15:00:00	sandy hook/highlands	
1708	1952-10-15 21:00:00	richland	
1709	1952-10-15 00:00:00	edmonton (southwest of) (canada)	
1710	1953-10-15 14:00:00	mauriceville	
1711	1954-10-15 16:30:00	blanco	
1712	1954-10-15 21:30:00	manistee	

1713	1955-10-15	20:00:00		rhinebeck
1714	1956-10-15	01:00:00		mechra bel ksire (morocco)
1715	1956-10-15	21:30:00		mt. vernon
1716	1957-10-15	02:00:00		new york city (queens)
1717	1957-10-15	20:30:00		houston (ellington afb)
1718	1957-10-15	22:00:00		commerce city (n. of denver)
1719	1959-10-15	00:00:00		marysville
2280	1959-10-16	22:30:00		four states
3088	1958-10-20	21:00:00		st. paul
4428	1958-10-25	20:00:00		terrebonne
5413	1958-10-30	01:00:00		riverside
...	
72743	1957-09-10	19:15:00		st. paul
72744	1960-09-10	10:00:00		wichita
73293	1954-09-01	13:00:00		geneva
73294	1957-09-01	16:00:00		graham
73295	1957-09-01	23:00:00		brecksville
73296	1958-09-01	20:00:00		fargo
74266	1958-09-14	19:30:00		ashland
74517	1952-09-15	14:00:00		norridgewock
74518	1954-09-15	09:30:00		palmer township
74519	1956-09-15	20:00:00		middle river
74520	1957-09-15	04:00:00		el paso
74521	1957-09-15	21:30:00		lake jackson
74522	1957-09-15	23:00:00		tacoma
74523	1959-09-15	01:00:00		goldfield
74524	1959-09-15	11:30:00		dayton
74525	1959-09-15	00:00:00		fallon
74526	1960-09-15	19:00:00		kingsport
74527	1960-09-15	20:30:00		seattle
74528	1960-09-15	21:00:00		susanville
74529	1960-09-15	21:30:00		barrhead (canada)
75992	1956-09-20	16:00:00		hope
75993	1957-09-20	00:00:00		hopkinsville
75994	1958-09-20	22:20:00	davis jct. (on highway72 east to highway 51 no...	
76277	1956-09-21	00:00:00		greenwich
79261	1957-09-05	17:30:00		los gatos (west of)
79262	1960-09-05	21:00:00		buffalo
79668	1958-09-07	19:00:00		arthur
80101	1952-09-09	20:00:00		philadelphia
80102	1954-09-09	12:30:00		beaumont
80103	1956-09-09	05:55:00		norfolk

	state/province	country	UFO_shape	length_of_encounter_seconds	\
2		NaN	gb	circle	20
3		tx	us	circle	20
4		hi	us	light	900
479		ky	us	disk	600

480	NaN	NaN	disk	1200
481	mo	us	disk	90
482	wv	us	circle	60
483	ca	NaN	circle	5
484	nh	us	rectangle	300
485	ak	us	unknown	180
486	nj	us	disk	10
487	ak	NaN	NaN	900
488	il	us	unknown	60
1707	nj	NaN	triangle	300
1708	wa	us	circle	5
1709	ab	ca	cylinder	120
1710	tx	us	disk	300
1711	tx	us	other	300
1712	mi	us	circle	8
1713	ny	us	light	2400
1714	NaN	NaN	unknown	300
1715	in	NaN	disk	8
1716	ny	us	NaN	10800
1717	tx	us	light	3
1718	co	us	fireball	600
1719	wa	us	other	75
2280	wv	us	sphere	600
3088	mn	us	other	2
4428	or	us	unknown	120
5413	il	us	light	600
...
72743	mn	us	disk	300
72744	ks	us	oval	1200
73293	ny	us	disk	120
73294	tx	us	sphere	30
73295	oh	us	circle	180
73296	nd	us	disk	900
74266	ky	us	circle	45
74517	me	us	disk	300
74518	pa	NaN	other	600
74519	md	us	circle	60
74520	tx	us	disk	600
74521	tx	us	fireball	13
74522	wa	us	disk	20
74523	nv	us	light	60
74524	oh	us	disk	900
74525	nv	us	oval	7200
74526	tn	us	circle	300
74527	wa	us	disk	120
74528	ca	us	NaN	360
74529	ab	ca	other	1200
75992	ar	us	circle	1200

75993	ky	us	cigar	600
75994	il	NaN	fireball	300
76277	ct	us	light	1290
79261	ca	us	disk	60
79262	ny	us	oval	180
79668	nd	us	unknown	900
80101	pa	us	circle	180
80102	tx	us	disk	300
80103	va	us	cigar	90

	described_duration_of_encounter \
2	20 seconds
3	1/2 hour
4	15 minutes
479	hr
480	about 20 mins
481	90 sec.
482	1 minute
483	five seconds+/-
484	5 minutes
485	3 minutes maximum
486	10 seconds
487	10-15min
488	1 minute
1707	5 minutes
1708	5 seconds
1709	approx. 2 min.
1710	5 min.
1711	3-5 mins
1712	8 seconds
1713	40 minutes
1714	approx. 5 minutes
1715	7 to 8 seconds
1716	2x3 hrs
1717	3 seconds
1718	10 min.
1719	01:15
2280	10 minutes
3088	2 seconds
4428	1-2 minutes
5413	10 minutes (?)
...	...
72743	five minutes
72744	20 min.
73293	couple of minutes
73294	30 sec
73295	3 minutes
73296	15 min

74266	45 seconds
74517	4-5 minutes
74518	9-10 minutes
74519	1 min
74520	10 min
74521	13 sec
74522	20 seconds
74523	1 minute
74524	15 mini
74525	2 hours
74526	5 minutes
74527	2 min. for me, longer for
74528	5-6 minutes
74529	20 minutes
75992	20 minutes
75993	10 minutes
75994	5min
76277	21:30
79261	one minute or less
79262	3 minutes
79668	5-15 min.
80101	3 minutes
80102	5 minutes
80103	1.5 minutes

	description	date_documented \
2	Green/Orange circular disc over Chester, En...	1/21/2008
3	My older brother and twin sister were leaving ...	1/17/2004
4	AS a Marine 1st Lt. flying an FJ4B fighter/att...	1/22/2004
479	SEVENTY SOME FLYING DISK OBSERVED IN 1950	4/16/2005
480	UFO seen by multiple U. S. military personnel;...	12/07/2006
481	1952 daylight sighting of multiple discs in fo...	10/31/2008
482	I saw the craft go across the horizon. It app...	04/12/2013
483	NO KNOWN AIRCRAFT (1955 and maybe today) COULD...	3/19/2009
484	Disk with glass center section, and flames ...	4/16/2005
485	This incident occurred about October of 1956 o...	12/07/2006
486	Formation of approximately 8 light gray metall...	4/16/2005
487	Radar sighting and filming	1/28/1999
488	Late 1959, bright lights from a hovering ob...	4/16/2005
1707	Triangles over New York Harbor 1950	10/30/2006
1708	Night sighting of seven large luminous UFOs fl...	11/21/2010
1709	Good evening folks---What I am about to discri...	1/17/2004
1710	My father was a WW1 Vet. and loved hunting. T...	11/04/2002
1711	We saw a silver football, motionless in the...	4/27/2004
1712	Saucer shaped, moved north to south directl...	1/21/2008
1713	This very distant star-like point of light cam...	7/19/2010
1714	Air Defense radar reflected an object with gre...	4/16/2005
1715	Clear night observation of lumenescent object ...	07/11/2000

1716	tall unidentified being enters home - N.Y.C. 1957	3/21/2003
1717	5 to 6" diameter light extremely fast...	9/17/2000
1718	A lot of methane gas in area	9/19/2002
1719	UFO Over Golf Course	4/27/2007
2280	MYSTERIOUS RED LIGHTS IN NORTHERN WEST VIRGINIA	3/31/2008
3088	A dull-orange craft sped southbound at low alt...	3/16/2000
4428	Piercing shrill tone.	08/05/2012
5413	Chicago suburbs strange lights and figures...	4/27/2007
...
72743	3 UFOs saucer-shaped with flashing lights ...	4/16/2005
72744	MADE ONLY A HUMMING SOUND EVEN WHEN IT TOOK...	7/25/2004
73293	While marching on a bivouac maneuver in (?) Se...	10/15/2003
73294	Three shiny spheres over Graham Texas	01/10/2009
73295	20 ft in diameter glowing ufo landed a few fee...	02/01/2007
73296	Flying Saucer seen while driving home from work.	5/27/2003
74266	Near observations of a slow moving silent obje...	2/14/2008
74517	SUNNY DAY WITHIN 450 FEET LOW...REPORT...	12/02/2000
74518	Saturn-shaped object appearing over the Pennsy...	11/28/2007
74519	three round blue-white disks in V formation mo...	12/16/2005
74520	Unidified Flying Object	2/14/2008
74521	fireball seen on country road.	01/10/2009
74522	a large disk spacecraft the size of a Boeing 7...	03/02/2004
74523	The object hovered over the pick-up and ra...	4/16/2005
74524	formation of 6 saucers moving from Wright Pate...	02/08/2005
74525	I was an Air Force radar operator at Fallon Na...	10/28/2002
74526	An orange circular craft drifted over the ...	1/28/1999
74527	Triple disc sighting prior to dusk with a ligh...	4/27/2004
74528	Three craft in formation high altitude...	4/27/2004
74529	HBCCUFO CANADIAN REPORT: Daddy there is a...	8/28/2003
75992	Sitting on a pond bank on sunny afternoon when...	4/16/2005
75993	4 matalic silver cigar shaped objects; no soun...	8/28/2003
75994	red light with A GLOW AROUND IT DRIVING TO WOR...	04/02/1999
76277	Strange movements among the stars	08/12/2008
79261	3 craft seen in day light very low altitud...	10/30/2006
79262	Precise movements of a "craft"; appar...	9/15/2005
79668	Two lights of alterneting color traveling and ...	10/31/2003
80101	saucers in a line over Phila Pa.	12/12/2009
80102	Aprox. 30 Disk shaped UFOs fell out of clouds ...	1/17/2004
80103	Unidentified Object Hovering over Interstate 2...	2/24/2007

	latitude	longitude
2	53.2	-2.916667
3	28.9783333	-96.645833
4	21.4180556	-157.803611
479	37.4791667	-82.518889
480	33.590355	130.401716
481	39.0997222	-94.578333
482	38.7230556	-80.650000

483	38.007059	-119.012259
484	43.3652778	-72.173889
485	61.9919444	-146.768611
486	40.9166667	-74.172222
487	70.292489	-161.903334
488	39.7338889	-90.228889
1707	40.443164	-73.989861
1708	46.2858333	-119.283333
1709	53.55	-113.500000
1710	30.2033333	-93.866111
1711	30.0977778	-98.421111
1712	44.2444444	-86.324167
1713	41.9266667	-73.913056
1714	34.56	-5.950000
1715	37.932266	-87.895027
1716	40.7141667	-74.006389
1717	29.7630556	-95.363056
1718	39.8083333	-104.933333
1719	48.0519444	-122.175833
2280	39.4797222	-80.310000
3088	44.9444444	-93.093056
4428	44.3530556	-121.176667
5413	41.835	-87.822778
...
72743	44.9444	-93.093056
72744	37.6922	-97.337222
73293	42.8689	-76.978056
73294	33.1069	-98.589167
73295	41.3197	-81.626944
73296	46.8772	-96.789444
74266	38.4783	-82.638056
74517	44.7131	-69.791111
74518	40.7168	-75.249344
74519	39.3342	-76.439722
74520	31.7586	-106.486389
74521	29.0336	-95.434167
74522	47.2531	-122.443056
74523	37.7086	-117.234722
74524	39.7589	-84.191667
74525	39.4736	-118.776389
74526	36.5483	-82.561944
74527	47.6064	-122.330833
74528	40.4164	-120.651944
74529	54.1167	-114.400000
75992	33.6669	-93.591389
75993	36.8656	-87.488611
75994	42.1017	-89.093157
76277	41.0264	-73.628889

```

79261      37.2267 -121.973611
79262      42.8864 -78.878611
79668      47.1042 -97.217778
80101      39.9522 -75.164167
80102      30.0858 -94.101667
80103      36.8467 -76.285556

```

[495 rows x 11 columns]

Q132. Write a Pandas program to extract the year, month, day, hour, minute, second, and weekday from unidentified flying object (UFO) reporting date.

```

In [23]: print("Sample Dataframe:")
          print(df.head())
          print("\nYear:")
          print(df.Date_time.dt.year.head())
          print("\nMonth:")
          print(df.Date_time.dt.month.head())
          print("\nDay:")
          print(df.Date_time.dt.day.head())
          print("\nHour:")
          print(df.Date_time.dt.hour.head())
          print("\nMinute:")
          print(df.Date_time.dt.minute.head())
          print("\nSecond:")
          print(df.Date_time.dt.second.head())
          print("\nWeekday:")
          print(df.Date_time.dt.weekday_name.head())

```

Sample Dataframe:

	Date_time	city	state/province	country	UFO_shape	\
0	1949-10-10 20:30:00	san marcos	tx	us	cylinder	
1	1949-10-10 21:00:00	lackland afb	tx	NaN	light	
2	1955-10-10 17:00:00	chester	(uk/england)	NaN	gb	circle
3	1956-10-10 21:00:00	edna	tx	us	circle	
4	1960-10-10 20:00:00	kaneohe	hi	us	light	

	length_of_encounter_seconds	described_duration_of_encounter	\
0	2700	45 minutes	
1	7200	1-2 hrs	
2	20	20 seconds	
3	20	1/2 hour	
4	900	15 minutes	

	description	date_documented	\
0	This event took place in early fall around 194...	4/27/2004	
1	1949 Lackland AFB, TX. Lights racing acros...	12/16/2005	

2	Green/Orange circular disc over Chester, En...	1/21/2008
3	My older brother and twin sister were leaving ...	1/17/2004
4	AS a Marine 1st Lt. flying an FJ4B fighter/att...	1/22/2004

	latitude	longitude
0	29.8830556	-97.941111
1	29.38421	-98.581082
2	53.2	-2.916667
3	28.9783333	-96.645833
4	21.4180556	-157.803611

Year:

0	1949
1	1949
2	1955
3	1956
4	1960

Name: Date_time, dtype: int64

Month:

0	10
1	10
2	10
3	10
4	10

Name: Date_time, dtype: int64

Day:

0	10
1	10
2	10
3	10
4	10

Name: Date_time, dtype: int64

Hour:

0	20
1	21
2	17
3	21
4	20

Name: Date_time, dtype: int64

Minute:

0	30
1	0
2	0
3	0

```
4      0
Name: Date_time, dtype: int64
```

Second:

```
0      0
1      0
2      0
3      0
4      0
Name: Date_time, dtype: int64
```

Weekday:

```
0      Monday
1      Monday
2      Monday
3  Wednesday
4      Monday
Name: Date_time, dtype: object
```

Q133. Write a pandas program to count year-country wise frequency of reporting dates of the unidentified flying object(UFO).

```
In [24]: df['Year'] = df['Date_time'].apply(lambda x: "%d" % (x.year))
         result = df.groupby(['Year', 'country']).size()
         print("\nCountry-year wise frequency of reporting dates of UFO:")
         print(result)
```

Country-year wise frequency of reporting dates of UFO:

Year	country	
1910	us	2
1920	us	1
1925	us	1
1929	us	1
1931	us	2
1934	us	1
1936	ca	1
	us	1
1937	us	2
1939	us	3
1941	us	1
1942	us	2
1943	gb	1
	us	1
1944	us	3
1945	us	7
1946	ca	1

```

      us      8
1947 us     33
1948 us      7
1949 us     15
1950 us     21
1951 ca      2
      gb      1
      us     13
1952 ca      2
      gb      1
      us     37
1953 ca      2
      us     26
      ...
2009 au     14
      ca    134
      de      6
      gb    202
      us   3677
2010 au     16
      ca    137
      de      4
      gb    115
      us   3548
2011 au     13
      ca    127
      de      3
      gb     51
      us   4379
2012 au     19
      ca    242
      de      6
      gb     82
      us   6320
2013 au     32
      ca    248
      de      6
      gb     48
      us   6056
2014 au     14
      ca     45
      de      3
      gb     21
      us   1964
Length: 282, dtype: int64

```

Q134. Write a pandas program to get the difference (in days) between documented date and

reporting date of unidentified flying object (UFO).

```
In [26]: df['date_documented'] = pd.to_datetime(df.date_documented.str.replace('24:', '00:'))
print("\nDifference (in days) between documented date and reporting date of UFO:")
df['Difference_in_days'] = (df['date_documented'] - df['Date_time']).dt.days
print(df)
```

Difference (in days) between documented date and reporting date of UFO:

	Date_time	city \
0	1949-10-10 20:30:00	san marcos
1	1949-10-10 21:00:00	lackland afb
2	1955-10-10 17:00:00	chester (uk/england)
3	1956-10-10 21:00:00	edna
4	1960-10-10 20:00:00	kaneohe
5	1961-10-10 19:00:00	bristol
6	1965-10-10 21:00:00	penarth (uk/wales)
7	1965-10-10 23:45:00	norwalk
8	1966-10-10 20:00:00	pell city
9	1966-10-10 21:00:00	live oak
10	1968-10-10 13:00:00	hawthorne
11	1968-10-10 19:00:00	brevard
12	1970-10-10 16:00:00	bellmore
13	1970-10-10 19:00:00	manchester
14	1971-10-10 21:00:00	lexington
15	1972-10-10 19:00:00	harlan county
16	1972-10-10 22:30:00	west bloomfield
17	1973-10-10 19:00:00	niantic
18	1973-10-10 23:00:00	bermuda nas
19	1974-10-10 19:30:00	hudson
20	1974-10-10 21:30:00	cardiff (uk/wales)
21	1974-10-10 23:00:00	hudson
22	1975-10-10 17:00:00	north charleston
23	1976-10-10 20:30:00	washougal
24	1976-10-10 22:00:00	stoke mandeville (uk/england)
25	1977-10-10 12:00:00	san antonio
26	1977-10-10 22:00:00	louisville
27	1978-10-10 02:00:00	elmont
28	1979-10-10 00:00:00	poughkeepsie
29	1979-10-10 22:00:00	saddle lake (canada)
...
80302	2012-09-09 20:00:00	wilson
80303	2012-09-09 20:10:00	elmont
80304	2012-09-09 20:30:00	mt. juliet
80305	2012-09-09 20:30:00	ventura
80306	2012-09-09 20:52:00	south jordan
80307	2012-09-09 21:00:00	elkhart
80308	2012-09-09 21:00:00	new york city (brooklyn)

80309	2012-09-09	21:00:00		pawleys island
80310	2012-09-09	21:00:00		ventura
80311	2012-09-09	21:55:00		charleston
80312	2012-09-09	23:00:00		gainesville
80313	2013-09-09	00:15:00		norfolk
80314	2013-09-09	01:50:00	buffalo (west of; on highway 90 west)	
80315	2013-09-09	03:00:00		struthers
80316	2013-09-09	09:51:00		san diego
80317	2013-09-09	12:34:00		cedar park
80318	2013-09-09	13:10:00		calmar (canada)
80319	2013-09-09	20:15:00		clifton
80320	2013-09-09	20:20:00		tuscaloosa
80321	2013-09-09	20:21:00		clarksville
80322	2013-09-09	21:00:00	aleksandrow (poland)	
80323	2013-09-09	21:00:00		gainesville
80324	2013-09-09	21:00:00	hamstead (hollyridge)	
80325	2013-09-09	21:00:00		milton (canada)
80326	2013-09-09	21:00:00		woodstock
80327	2013-09-09	21:15:00		nashville
80328	2013-09-09	22:00:00		boise
80329	2013-09-09	22:00:00		napa
80330	2013-09-09	22:20:00		vienna
80331	2013-09-09	23:00:00		edmond

	state/province	country	UFO_shape	length_of_encounter_seconds	\
0	tx	us	cylinder	2700	
1	tx	NaN	light	7200	
2	NaN	gb	circle	20	
3	tx	us	circle	20	
4	hi	us	light	900	
5	tn	us	sphere	300	
6	NaN	gb	circle	180	
7	ct	us	disk	1200	
8	al	us	disk	180	
9	fl	us	disk	120	
10	ca	us	circle	300	
11	nc	us	fireball	180	
12	ny	us	disk	1800	
13	ky	us	unknown	180	
14	nc	us	oval	30	
15	ky	us	circle	1200	
16	mi	us	disk	120	
17	ct	us	disk	1800	
18	NaN	NaN	light	20	
19	ma	us	other	2700	
20	NaN	gb	disk	1200	
21	ks	us	light	1200	
22	sc	us	light	360	

23	wa	us	oval	60
24	NaN	gb	cigar	3
25	tx	us	other	30
26	ky	us	light	30
27	ny	us	rectangle	300
28	ny	us	chevron	900
29	ab	NaN	triangle	270
...
80302	nc	us	light	10800
80303	ny	us	circle	600
80304	tn	us	light	120
80305	ca	us	chevron	900
80306	ut	us	circle	10
80307	in	us	oval	600
80308	ny	us	light	1290
80309	sc	us	oval	60
80310	ca	us	circle	300
80311	sc	us	flash	900
80312	ga	us	light	5
80313	va	us	unknown	1
80314	ny	us	triangle	180
80315	oh	us	unknown	120
80316	ca	us	light	4
80317	tx	us	cigar	8
80318	ab	ca	unknown	90
80319	nj	NaN	other	3600
80320	al	us	fireball	60
80321	tn	us	fireball	3
80322	NaN	NaN	light	15
80323	fl	us	triangle	60
80324	nc	NaN	light	120
80325	on	ca	fireball	180
80326	ga	us	sphere	20
80327	tn	us	light	600
80328	id	us	circle	1200
80329	ca	us	other	1200
80330	va	us	circle	5
80331	ok	us	cigar	1020

	described_duration_of_encounter \
0	45 minutes
1	1-2 hrs
2	20 seconds
3	1/2 hour
4	15 minutes
5	5 minutes
6	about 3 mins
7	20 minutes

8	3 minutes
9	several minutes
10	5 min.
11	3 minutes
12	30 min.
13	3 minutes
14	30 seconds
15	20minutes
16	2 minutes
17	20-30 min
18	20 sec.
19	45 minutes
20	20 minutes
21	one hour?
22	5-6 minutes
23	1 minute
24	3 seconds
25	30 seconds
26	approx: 30 seconds
27	5min
28	15 minutes
29	4.5 or more min.
...	...
80302	3 hours
80303	10 minutes
80304	2 minutes
80305	15 minutes
80306	10 seconds
80307	10 minutes
80308	21:30
80309	less than a minute
80310	5 minutes
80311	15 minutes
80312	5 seconds
80313	split second
80314	3 minutes
80315	2 minutes
80316	~4 seconds
80317	5-8 seconds
80318	45-90 seconds
80319	~1hr+
80320	01:00
80321	3 seconds
80322	15 seconds
80323	1 minute
80324	2 minutes
80325	3 minutes
80326	20 seconds

80327	10 minutes
80328	20 minutes
80329	hour
80330	5 seconds
80331	17 minutes

	description	date_documented \
0	This event took place in early fall around 194...	2004-04-27
1	1949 Lackland AFB TX. Lights racing across...	2005-12-16
2	Green/Orange circular disc over Chester En...	2008-01-21
3	My older brother and twin sister were leaving ...	2004-01-17
4	AS a Marine 1st Lt. flying an FJ4B fighter/att...	2004-01-22
5	My father is now 89 my brother 52 the girl wit...	2007-04-27
6	penarth uk circle 3mins stayed 30ft above m...	2006-02-14
7	A bright orange color changing to reddish colo...	1999-10-02
8	Strobe Lighted disk shape object observed clos...	2009-03-19
9	Saucer zaps energy from powerline as my pregna...	2005-05-11
10	ROUND ORANGE WITH WHAT I WOULD SAY W...	2003-10-31
11	silent red /orange mass of energy floated by t...	2008-06-12
12	silver disc seen by family and neighbors	2000-05-11
13	Slow moving silent craft accelerated at an...	2008-02-14
14	green oval shaped light over my local church#...	2010-02-14
15	On october 10 1972 myself my 5yrs.daugh...	2005-09-15
16	The UFO was so close my battery in the car...	2007-08-14
17	Oh what a night Two (2) saucer-shape...	2003-09-24
18	saw fast moving blip on the radar scope thin w...	2002-01-11
19	Not sure of the exact month or year of this sig...	1999-08-10
20	back in 1974 I was 19 at the time and lived i...	2007-02-01
21	The light chased us.	2004-07-25
22	Several Flashing UFO lights over Charleston Na...	2008-02-14
23	Three extremely large lights hanging above nea...	2014-02-07
24	White object over Buckinghamshire UK.	2009-12-12
25	i was about six or seven and my family and me ...	2005-02-24
26	HBCCUFO CANADIAN REPORT: Pilot Sighting Of Un...	2004-03-17
27	A memory I will never forget that happened men...	2007-02-01
28	1/4 moon-like its chord or flat s...	2005-04-16
29	Lights far above that glance; then flee f...	2005-01-19
...
80302	Bright orb being chased by a jet along with se...	2012-09-24
80303	Orange lights seen in Elmont Long Island#...	2012-09-24
80304	Bright white light moving slowly across sky wi...	2012-09-24
80305	Beautiful bright blue delta shaped aerobatics.	2012-09-24
80306	Circular disk with blinking lights scares two ...	2012-09-24
80307	It was the night of sept 9 between 9 and 10 pm...	2012-09-24
80308	Glowing circular lights visible in the clo...	2012-09-24
80309	One large bright orange flanked by three small...	2012-09-24
80310	Bright Blue Object seen floating in sky near C...	2012-09-24
80311	Orb of light flashing reds and blues stati...	2012-09-24

80312	Ball of light	2012-09-24
80313	Two or three lights shoot across sky over nava...	2013-09-30
80314	Massive Flat Black triangle with 3 red lights.	2013-09-30
80315	I saw a routaing line of stares that seemed to...	2013-09-09
80316	2 white lights zig-zag over Qualcomm Stadium (...)	2013-09-30
80317	Cigar Shaped Object Descending in the Directio...	2013-09-09
80318	Fastest dot I have ever seen in the sky!	2013-09-09
80319	Luminous line seen in New Jersey sky.	2013-09-30
80320	White/green object much larger than "shoo...	2013-09-30
80321	Green fireball like object shooting across the...	2013-09-30
80322	Two points of light following one another in a...	2013-09-30
80323	Three lights in the sky that didn't look li...	2013-09-30
80324	8 to ten lights bright orange in color large t...	2013-09-30
80325	Massive Bright Orange Fireball in Sky	2013-09-30
80326	Driving 575 at 21:00 hrs saw a white and green...	2013-09-30
80327	Round from the distance/slowly changing colors...	2013-09-30
80328	Boise, ID, spherical, 20 min, 10 r...	2013-09-30
80329	Napa UFO,	2013-09-30
80330	Saw a five gold lit cicular craft moving fastl...	2013-09-30
80331	2 witnesses 2 miles apart, Red & White...	2013-09-30

	latitude	longitude	Year	Difference_in_days
0	29.8830556	-97.941111	1949	19922
1	29.38421	-98.581082	1949	20520
2	53.2	-2.916667	1955	19095
3	28.9783333	-96.645833	1956	17264
4	21.4180556	-157.803611	1960	15808
5	36.595	-82.188889	1961	16634
6	51.434722	-3.180000	1965	14736
7	41.1175	-73.408333	1965	12409
8	33.5861111	-86.286111	1966	15500
9	30.2947222	-82.984167	1966	14092
10	33.9163889	-118.351667	1968	12803
11	35.2333333	-82.734444	1968	14489
12	40.6686111	-73.527500	1970	10805
13	37.1536111	-83.761944	1970	13640
14	35.8238889	-80.253611	1971	14006
15	36.8430556	-83.321944	1972	12027
16	42.5377778	-83.233056	1972	12725
17	41.3252778	-72.193611	1973	10940
18	32.364167	-64.678611	1973	10319
19	42.3916667	-71.566667	1974	9069
20	51.5	-3.200000	1974	11801
21	38.1055556	-98.659722	1974	10880
22	32.8544444	-79.975000	1975	11814
23	45.5827778	-122.352222	1976	13633
24	51.783333	-0.783333	1976	12115
25	29.4238889	-98.493333	1977	9998

26	38.2541667	-85.759444	1977	9654
27	40.7008333	-73.713333	1978	10340
28	41.7002778	-73.921389	1979	9320
29	53.970571	-111.689885	1979	9232
...
80302	35.7211	-77.915833	2012	14
80303	40.7008	-73.713333	2012	14
80304	36.2	-86.518611	2012	14
80305	34.2783	-119.292222	2012	14
80306	40.5622	-111.928889	2012	14
80307	41.6819	-85.976667	2012	14
80308	40.7142	-74.006389	2012	14
80309	33.4331	-79.121667	2012	14
80310	34.2783	-119.292222	2012	14
80311	32.7764	-79.931111	2012	14
80312	34.2978	-83.824167	2012	14
80313	36.8467	-76.285556	2013	20
80314	42.8864	-78.878611	2013	20
80315	41.0525	-80.608056	2013	-1
80316	32.7153	-117.156389	2013	20
80317	30.505	-97.820000	2013	-1
80318	53.25	-113.783333	2013	-1
80319	40.8584	-74.163755	2013	20
80320	33.2097	-87.569167	2013	20
80321	36.5297	-87.359444	2013	20
80322	50.4658	22.891814	2013	20
80323	29.6514	-82.325000	2013	20
80324	34.3676	-77.710548	2013	20
80325	46.3	-63.216667	2013	20
80326	34.1014	-84.519444	2013	20
80327	36.1658	-86.784444	2013	20
80328	43.6136	-116.202500	2013	20
80329	38.2972	-122.284444	2013	20
80330	38.9011	-77.265556	2013	20
80331	35.6528	-97.477778	2013	20

[80332 rows x 13 columns]

Q135. Write a pandas program to generate sequences of fixed- frequency dates and time spans.

```
In [27]: import pandas as pd
dtr = pd.date_range('2018-01-01', periods=12, freq='H')
print("Hourly frequency:")
print(dtr)
dtr = pd.date_range('2018-01-01', periods=12, freq='min')
print("\nMinutely frequency:")
print(dtr)
```

```

dtr = pd.date_range('2018-01-01', periods=12, freq='S')
print("\nSecondly frequency:")
print(dtr)
dtr = pd.date_range('2018-01-01', periods=12, freq='2H')
print("\nMultiple Hourly frequency:")
print(dtr)
dtr = pd.date_range('2018-01-01', periods=12, freq='5min')
print("\nMultiple Minutely frequency:")
print(dtr)
dtr = pd.date_range('2018-01-01', periods=12, freq='BQ')
print("\nMultiple Secondly frequency:")
print(dtr)
dtr = pd.date_range('2018-01-01', periods=12, freq='w')
print("\nWeekly frequency:")
print(dtr)
dtr = pd.date_range('2018-01-01', periods=12, freq='2h20min')
print("\nCombine together day and intraday offsets-1:")
print(dtr)
dtr = pd.date_range('2018-01-01', periods=12, freq='1D10U')
print("\nCombine together day and intraday offsets-2:")
print(dtr)

```

Hourly frequency:

```

DatetimeIndex(['2018-01-01 00:00:00', '2018-01-01 01:00:00',
               '2018-01-01 02:00:00', '2018-01-01 03:00:00',
               '2018-01-01 04:00:00', '2018-01-01 05:00:00',
               '2018-01-01 06:00:00', '2018-01-01 07:00:00',
               '2018-01-01 08:00:00', '2018-01-01 09:00:00',
               '2018-01-01 10:00:00', '2018-01-01 11:00:00'],
              dtype='datetime64[ns]', freq='H')

```

Minutely frequency:

```

DatetimeIndex(['2018-01-01 00:00:00', '2018-01-01 00:01:00',
               '2018-01-01 00:02:00', '2018-01-01 00:03:00',
               '2018-01-01 00:04:00', '2018-01-01 00:05:00',
               '2018-01-01 00:06:00', '2018-01-01 00:07:00',
               '2018-01-01 00:08:00', '2018-01-01 00:09:00',
               '2018-01-01 00:10:00', '2018-01-01 00:11:00'],
              dtype='datetime64[ns]', freq='T')

```

Secondly frequency:

```

DatetimeIndex(['2018-01-01 00:00:00', '2018-01-01 00:00:01',
               '2018-01-01 00:00:02', '2018-01-01 00:00:03',
               '2018-01-01 00:00:04', '2018-01-01 00:00:05',
               '2018-01-01 00:00:06', '2018-01-01 00:00:07',
               '2018-01-01 00:00:08', '2018-01-01 00:00:09',
               '2018-01-01 00:00:10', '2018-01-01 00:00:11'],
              dtype='datetime64[ns]', freq='S')

```

nMultiple Hourly frequency:

```
DatetimeIndex(['2018-01-01 00:00:00', '2018-01-01 02:00:00',
               '2018-01-01 04:00:00', '2018-01-01 06:00:00',
               '2018-01-01 08:00:00', '2018-01-01 10:00:00',
               '2018-01-01 12:00:00', '2018-01-01 14:00:00',
               '2018-01-01 16:00:00', '2018-01-01 18:00:00',
               '2018-01-01 20:00:00', '2018-01-01 22:00:00'],
              dtype='datetime64[ns]', freq='2H')
```

Multiple Minutely frequency:

```
DatetimeIndex(['2018-01-01 00:00:00', '2018-01-01 00:05:00',
               '2018-01-01 00:10:00', '2018-01-01 00:15:00',
               '2018-01-01 00:20:00', '2018-01-01 00:25:00',
               '2018-01-01 00:30:00', '2018-01-01 00:35:00',
               '2018-01-01 00:40:00', '2018-01-01 00:45:00',
               '2018-01-01 00:50:00', '2018-01-01 00:55:00'],
              dtype='datetime64[ns]', freq='5T')
```

Multiple Secondly frequency:

```
DatetimeIndex(['2018-03-30', '2018-06-29', '2018-09-28', '2018-12-31',
               '2019-03-29', '2019-06-28', '2019-09-30', '2019-12-31',
               '2020-03-31', '2020-06-30', '2020-09-30', '2020-12-31'],
              dtype='datetime64[ns]', freq='BQ-DEC')
```

Weekly frequency:

```
DatetimeIndex(['2018-01-07', '2018-01-14', '2018-01-21', '2018-01-28',
               '2018-02-04', '2018-02-11', '2018-02-18', '2018-02-25',
               '2018-03-04', '2018-03-11', '2018-03-18', '2018-03-25'],
              dtype='datetime64[ns]', freq='W-SUN')
```

Combine together day and intraday offsets-1:

```
DatetimeIndex(['2018-01-01 00:00:00', '2018-01-01 02:20:00',
               '2018-01-01 04:40:00', '2018-01-01 07:00:00',
               '2018-01-01 09:20:00', '2018-01-01 11:40:00',
               '2018-01-01 14:00:00', '2018-01-01 16:20:00',
               '2018-01-01 18:40:00', '2018-01-01 21:00:00',
               '2018-01-01 23:20:00', '2018-01-02 01:40:00'],
              dtype='datetime64[ns]', freq='140T')
```

Combine together day and intraday offsets-2:

```
DatetimeIndex(['2018-01-01 00:00:00', '2018-01-02 00:00:00.000010',
               '2018-01-03 00:00:00.000020', '2018-01-04 00:00:00.000030',
               '2018-01-05 00:00:00.000040', '2018-01-06 00:00:00.000050',
               '2018-01-07 00:00:00.000060', '2018-01-08 00:00:00.000070',
               '2018-01-09 00:00:00.000080', '2018-01-10 00:00:00.000090',
               '2018-01-11 00:00:00.000100', '2018-01-12 00:00:00.000110'],
              dtype='datetime64[ns]', freq='86400000010U')
```

Q136. Write a pandas program to manipulate and convert date times with timezone information.

```
In [29]: import pandas as pd
        dtt = pd.date_range('2019-12-12', periods=3, freq='H')
        dtt = dtt.tz_localize('UTC')
        print(dtt)
        print("\nFrom UTC to America/Los_Angeles:")
        dtt = dtt.tz_convert('America/Los_Angeles')
        print(dtt)

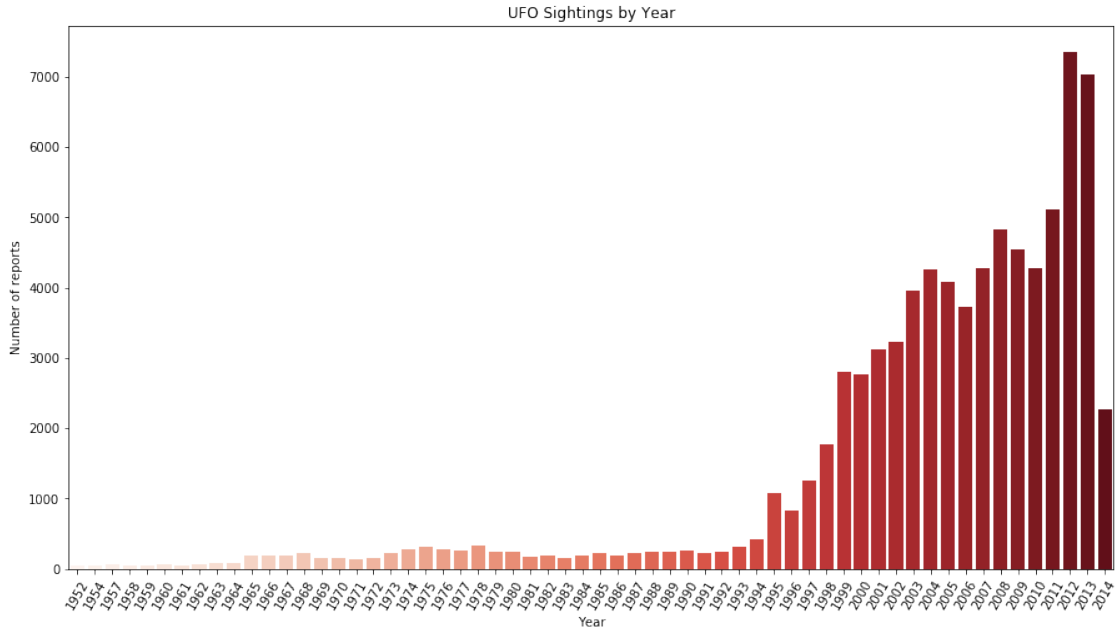
DatetimeIndex(['2019-12-12 00:00:00+00:00', '2019-12-12 01:00:00+00:00',
               '2019-12-12 02:00:00+00:00'],
              dtype='datetime64[ns, UTC]', freq='H')

From UTC to America/Los_Angeles:
DatetimeIndex(['2019-12-11 16:00:00-08:00', '2019-12-11 17:00:00-08:00',
               '2019-12-11 18:00:00-08:00'],
              dtype='datetime64[ns, America/Los_Angeles]', freq='H')
```

Q137. Write a pandas program to create the graphical analysis of UFO (unidentified flying object) Sightings year.

```
In [31]: import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        df = pd.read_csv(r'ufo.csv')
        df['Date_time'] = pd.to_datetime(df.Date_time.str.replace('24:', '00:'))
        df["ufo_yr"] = df.Date_time.dt.year
        years_data = df.ufo_yr.value_counts()
        years_index = years_data.index # x ticks
        years_values = years_data.get_values()
        plt.figure(figsize=(15,8))
        plt.xticks(rotation = 60)
        plt.title('UFO Sightings by Year')
        plt.xlabel("Year")
        plt.ylabel("Number of reports")
        years_plot = sns.barplot(x=years_index[:60], y=years_values[:60], palette = "Reds")

C:\Apps\Continuum\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2728: DtypeWarning:
interactivity=interactivity, compiler=compiler, result=result)
```



Q138. Write a pandas program to create a comparison of the top 10 years in which the (UFO) was sighted VS each Month.

```
In [32]: import pandas as pd
#Source: https://bit.ly/1l9yjm9
df = pd.read_csv(r'ufo.csv')
df['Date_time'] = pd.to_datetime(df.Date_time.str.replace('24:', '00:'))
most_sightings_years = df['Date_time'].dt.year.value_counts().head(10)
def is_top_years(year):
    if year in most_sightings_years.index:
        return year
hour_v_year = df.pivot_table(columns=df['Date_time'].dt.hour, index=df['Date_time'].dt
hour_v_year.columns = hour_v_year.columns.astype(int)
hour_v_year.columns = hour_v_year.columns.astype(str) + ":00"
hour_v_year.index = hour_v_year.index.astype(int)
print("\nComparison of the top 10 years in which the UFO was sighted vs the hours of t
print(hour_v_year.head(10))
```

C:\Apps\Continuum\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2728: DtypeWarning:
interactivity=interactivity, compiler=compiler, result=result)

Comparison of the top 10 years in which the UFO was sighted vs the hours of the day:

	0:00	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	...	\
Date_time											...	
2003	247	194	156	105	74	80	65	49	32	34	...	
2004	256	172	134	118	117	97	75	54	37	53	...	

2005	258	159	119	116	74	87	64	52	42	47	...
2007	234	166	107	99	92	112	84	38	53	57	...
2008	321	203	149	120	81	106	85	67	64	65	...
2009	300	158	97	94	106	90	71	37	46	73	...
2010	288	155	104	90	76	87	59	47	44	67	...
2011	301	205	114	122	84	106	71	42	59	71	...
2012	405	244	160	152	120	122	113	85	71	96	...
2013	370	201	127	109	88	94	93	79	70	73	...

	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	\
Date_time										
2003	83	80	123	144	225	265	378	439	474	
2004	73	99	105	129	186	319	469	508	536	
2005	87	90	95	153	226	384	400	485	453	
2007	83	90	83	160	221	295	429	557	567	
2008	92	75	89	151	259	380	495	578	581	
2009	62	66	82	143	227	380	544	686	555	
2010	38	76	72	98	182	293	422	687	658	
2011	58	74	72	163	245	399	587	829	758	
2012	65	87	108	204	309	559	901	1307	1148	
2013	44	55	114	224	371	593	919	1332	1141	

	23:00
Date_time	
2003	448
2004	410
2005	415
2007	425
2008	502
2009	455
2010	463
2011	474
2012	759
2013	673

[10 rows x 24 columns]

Q139. Write a pandas program to create a heatmap (rectangular data as a colour-encoded matrix) for comparison of top 10 years in which (UFO) was sighted VS each Month.

```
In [48]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv(r'ufo.csv')
df['Date_time'] = pd.to_datetime(df.Date_time.str.replace('24:', '00:'))
most_sightings_years = df['Date_time'].dt.year.value_counts().head(10)
#df[df.most_sightings_years]
```

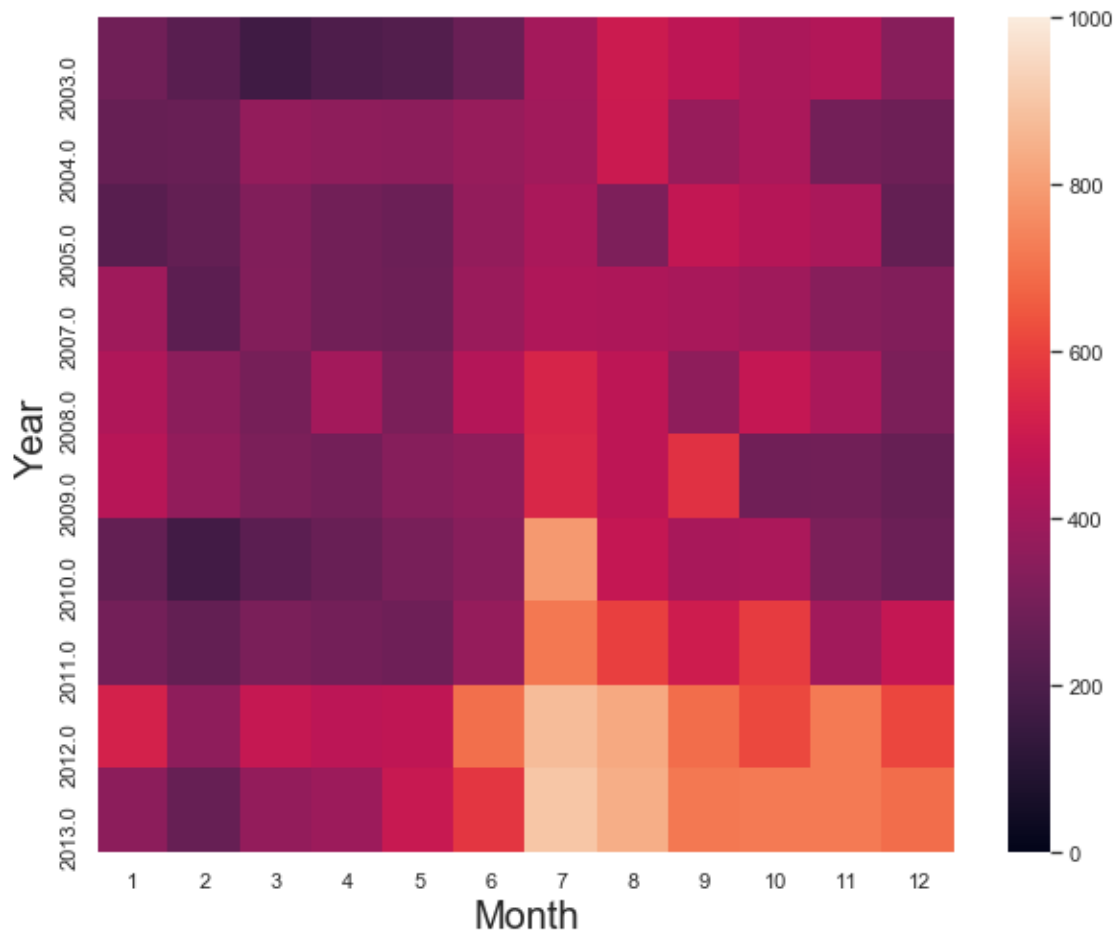
```

def is_top_years(year):
    if year in most_sightings_years.index:
        return year
month_vs_year = df.pivot_table(columns=df['Date_time'].dt.month,index=df['Date_time'])
month_vs_year.columns = month_vs_year.columns.astype(int)
print("\nHeatmap for comparison of the top 10 years in which the UFO was sighted vs ea
plt.figure(figsize=(10,8))
ax = sns.heatmap(month_vs_year, vmin=0, vmax=1000)
ax.set_xlabel('Month').set_size(20)
ax.set_ylabel('Year').set_size(20)

```

C:\Apps\Continuum\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2728: DtypeWarning:
interactivity=interactivity, compiler=compiler, result=result)

Heatmap for comparison of the top 10 years in which the UFO was sighted vs each month:



Q140. Write a pandas program to create a Timewheel of Hour VS Year comparison of the top 10 years in which the (UFO) was sighted.

```
In [49]: import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
import matplotlib.cm as cm
df = pd.read_csv(r'ufu.csv')
df['Date_time'] = pd.to_datetime(df.Date_time.str.replace('24:', '00:'))
most_sightings_years = df['Date_time'].dt.year.value_counts().head(10)
def is_top_years(year):
    if year in most_sightings_years.index:
        return year
month_vs_year = df.pivot_table(columns=df['Date_time'].dt.month, index=df['Date_time'])
month_vs_year.index = month_vs_year.index.astype(int)
month_vs_year.columns = month_vs_year.columns.astype(int)
print("\nComparison of the top 10 years in which the UFO was sighted vs each month:")
def pie_heatmap(table, cmap='coolwarm_r', vmin=None, vmax=None, inner_r=0.25, pie_args=):
    n, m = table.shape
    vmin= table.min().min() if vmin is None else vmin
    vmax= table.max().max() if vmax is None else vmax

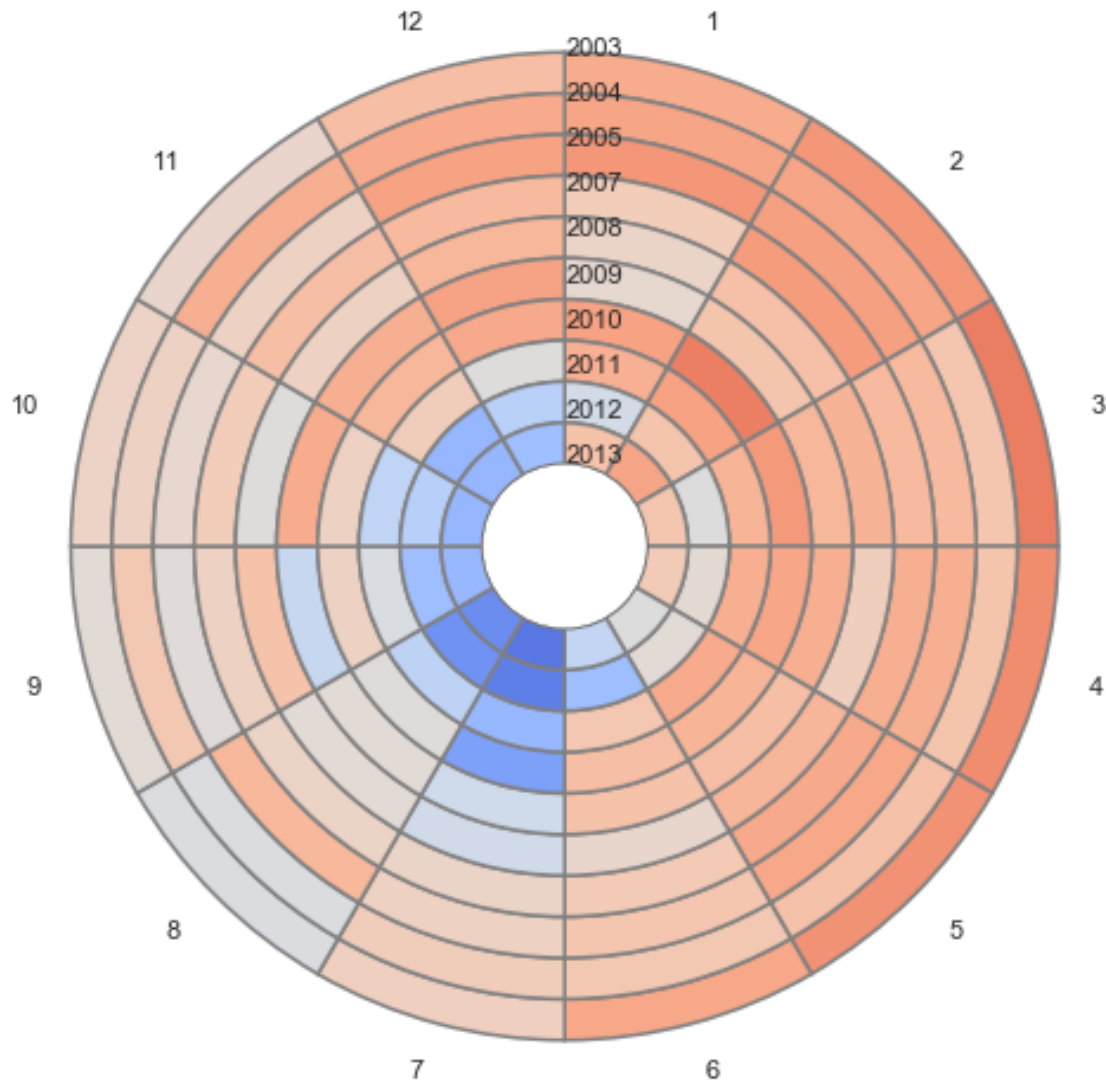
    centre_circle = plt.Circle((0,0),inner_r,edgecolor='black',facecolor='white',fill=True)
    plt.gcf().gca().add_artist(centre_circle)
    norm = mpl.colors.Normalize(vmin=vmin, vmax=vmax)
    cmapper = cm.ScalarMappable(norm=norm, cmap=cmap)

    for i, (row_name, row) in enumerate(table.iterrows()):
        labels = None if i > 0 else table.columns
        wedges = plt.pie([1] * m, radius=inner_r+float(n-i)/n, colors=[cmapper.to_rgba(x) for x in row],
            labels=labels, startangle=90, counterclock=False, wedgeprops={'linewidth': 1.5})
        plt.setp(wedges[0], edgecolor='grey', linewidth=1.5)
        wedges = plt.pie([1], radius=inner_r+float(n-i-1)/n, colors=['w'], labels=[row_name])
        plt.setp(wedges[0], edgecolor='grey', linewidth=1.5)
    plt.figure(figsize=(8,8))
    plt.title("Timewheel of Hour Vs Year", y=1.08, fontsize=30)
    pie_heatmap(month_vs_year, vmin=-20, vmax=1000, inner_r=0.2)
```

```
C:\Apps\Continuum\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2728: DtypeWarning:
interactivity=interactivity, compiler=compiler, result=result)
```

Comparison of the top 10 years in which the UFO was sighted vs each month:

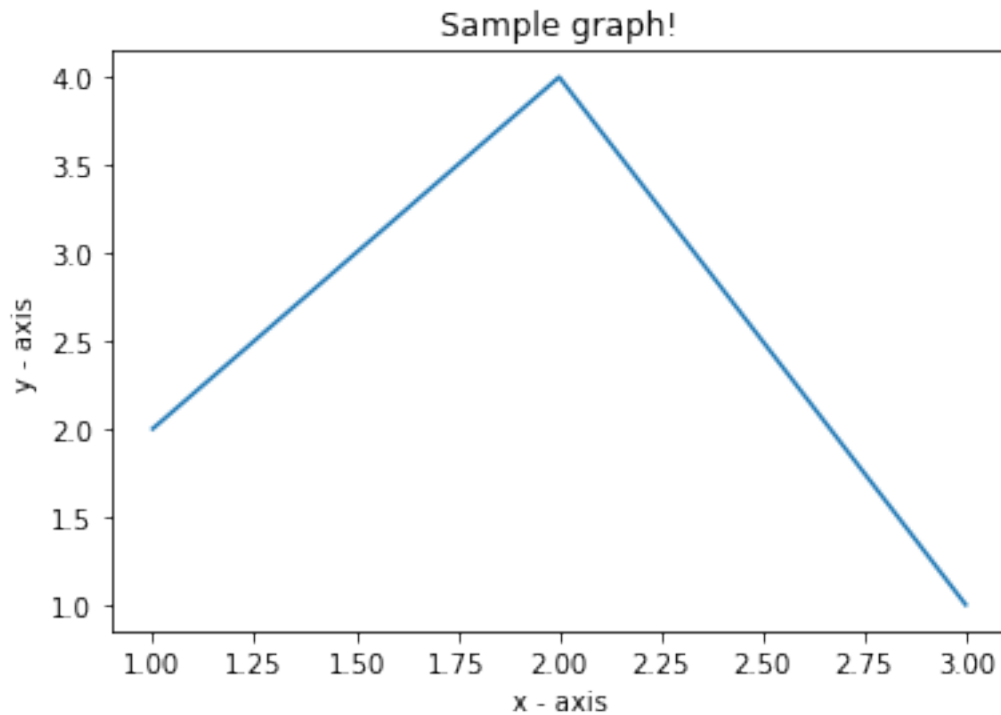
Timewheel of Hour Vs Year



Q141. Write a python program to draw the line using given axis values with the suitable label in the x-axis, y-axis, and a title.

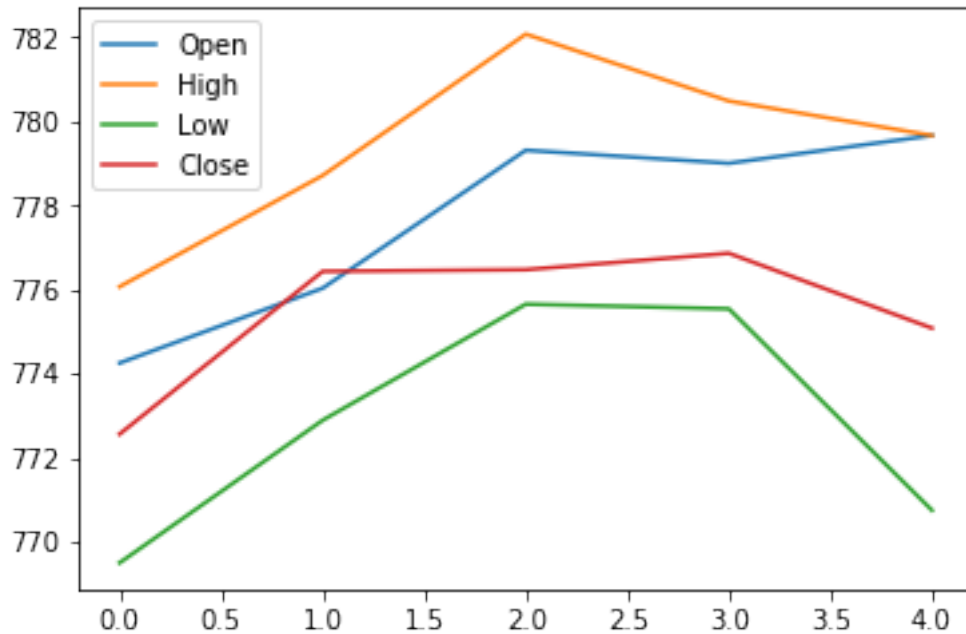
```
In [11]: import matplotlib.pyplot as plt
import numpy as np
x = [1,2,3]
y = [2,4,1]
plt.plot(x, y)
# Set the x axis label of the current axis.
plt.xlabel('x - axis')
# Set the y axis label of the current axis.
```

```
plt.ylabel('y - axis')
# Set a title
plt.title('Sample graph!')
# Display a figure.
plt.show()
```



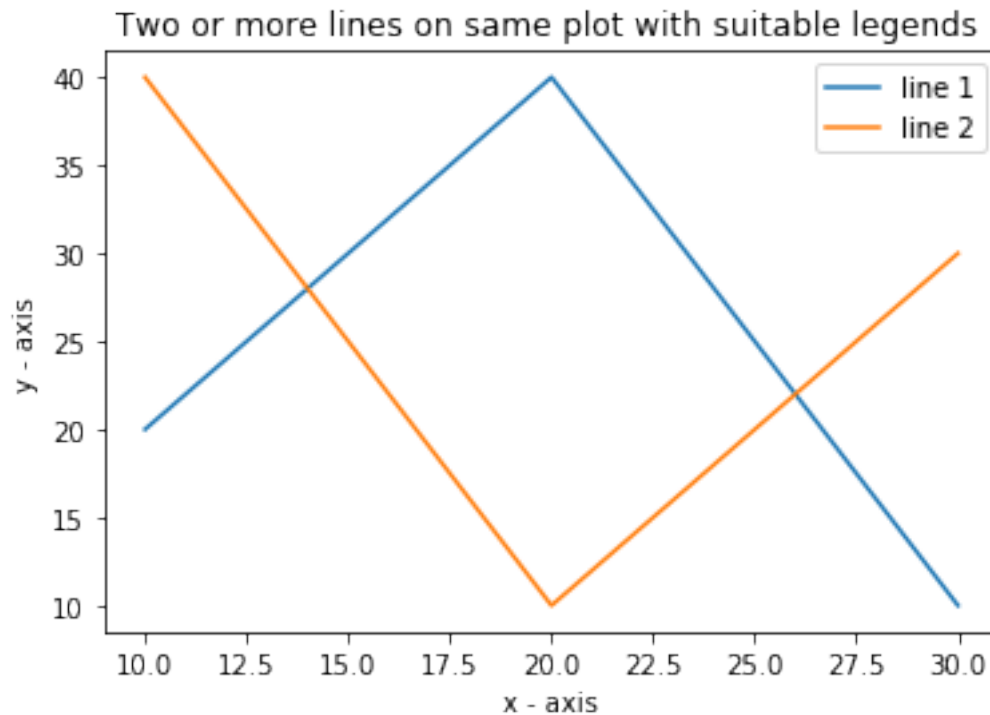
Q142. Write a python program to draw the line charts of the financial data of the Alphabet Inc., between October.

```
In [20]: import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv('finance_data.csv', sep=',', parse_dates=True)
df.plot()
plt.show()
```



Q143. Write a Python program to plot two or more lines on same plot with the suitable legends of each line.

```
In [21]: import matplotlib.pyplot as plt
x1 = [10,20,30]
y1 = [20,40,10]
# plotting the line 1 points
plt.plot(x1, y1, label = "line 1")
x2 = [10,20,30]
y2 = [40,10,30]
# plotting the line 2 points
plt.plot(x2, y2, label = "line 2")
# set label for x and y axis
plt.xlabel('x - axis')
plt.ylabel('y - axis')
# Title
plt.title('Two or more lines on same plot with suitable legends ')
# show a legend on the plot
plt.legend()
plt.show()
```



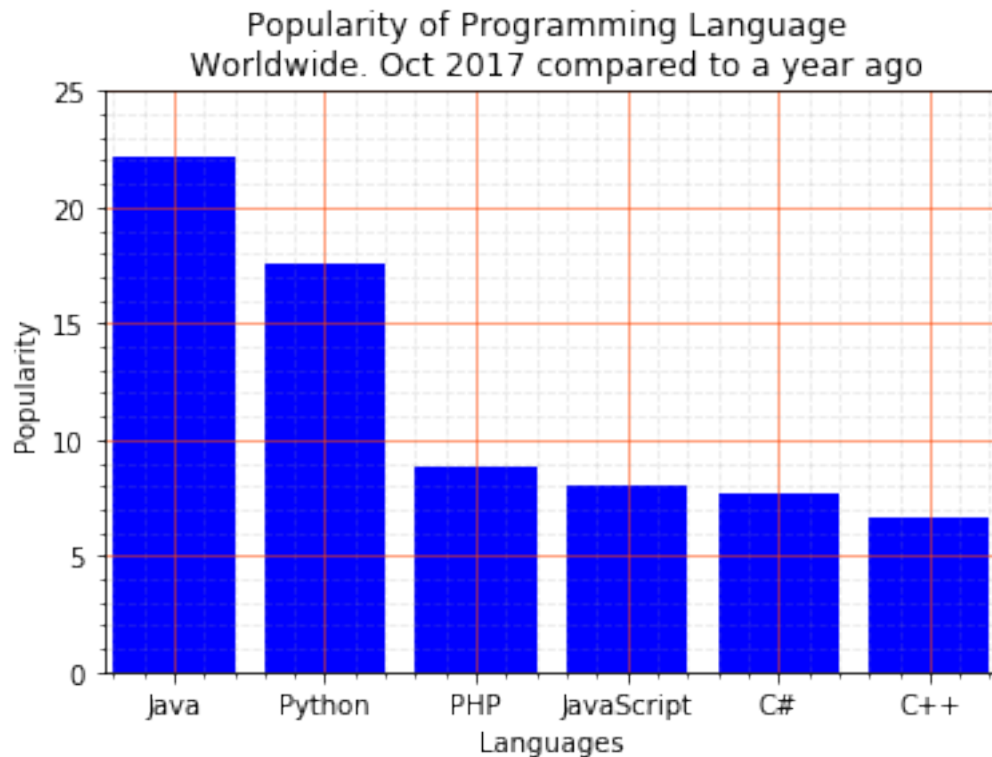
Q144. Write a python programming to display a bar chart of the popularity of programming languages.

```
In [101]: import matplotlib.pyplot as plt
import numpy as np

Languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
Popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

index = np.arange(len(Languages))
plt.bar(index, Popularity, color = 'blue')
plt.xlabel('Languages', fontsize=10)
plt.ylabel('Popularity', fontsize=10)
plt.ylim(0, 25)
plt.xticks(index, Languages, fontsize=10, rotation=0)
plt.margins(0.01)
plt.title('Popularity of Programming Language \n Worldwide. Oct 2017 compared to a y
# Show the major grid lines with dark grey lines
plt.grid(zorder=0)
plt.grid(axis='both', which='major', color='#FF4500', linestyle='--', linewidth=0.5 ) #
# Show the minor grid lines with very faint and almost transparent grey lines
plt.minorticks_on()
plt.grid(which='minor', color='#999999', linestyle='--', alpha=0.2)

plt.show()
```



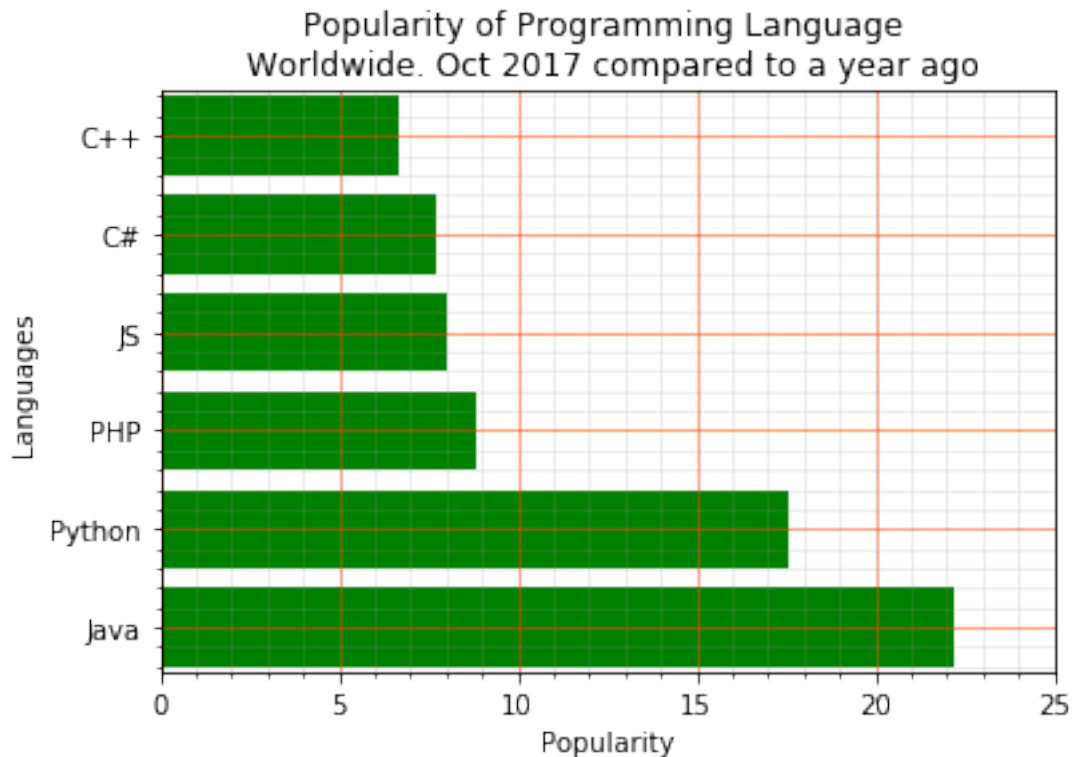
Q145. Write a python programming to display a horizontal bar chart of the popularity of programming languages.

```
In [83]: Languages = ['Java', 'Python', 'PHP', 'JS', 'C#', 'C++']
Popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

index = np.arange(len(Languages))
plt.barh(index, Popularity,color = 'green')
plt.xlabel('Popularity', fontsize=10)
plt.ylabel('Languages', fontsize=10)
plt.xlim(0, 25)
plt.yticks(index, Languages, fontsize=10, rotation=0)
plt.title('Popularity of Programming Language \n Worldwide. Oct 2017 compared to a year ago')
# Show the major grid lines with dark grey lines
plt.grid(b=True,which='major', color='#FF4500', linestyle='-',linewidth=0.5 ) #linestyle='solid'

#plt.grid(b=True, which='major', color='#666666', linestyle='-')

# Show the minor grid lines with very faint and almost transparent grey lines
plt.minorticks_on()
plt.grid(b=True, which='minor', color='#999999', linestyle='-', alpha=0.2)
plt.margins(0.01)
plt.show()
```



Q146. Write a python programming to display a bar chart of the Popularity of programming languages. Increase bottom margin.

```
In [22]: import matplotlib.pyplot as plt
import numpy as np

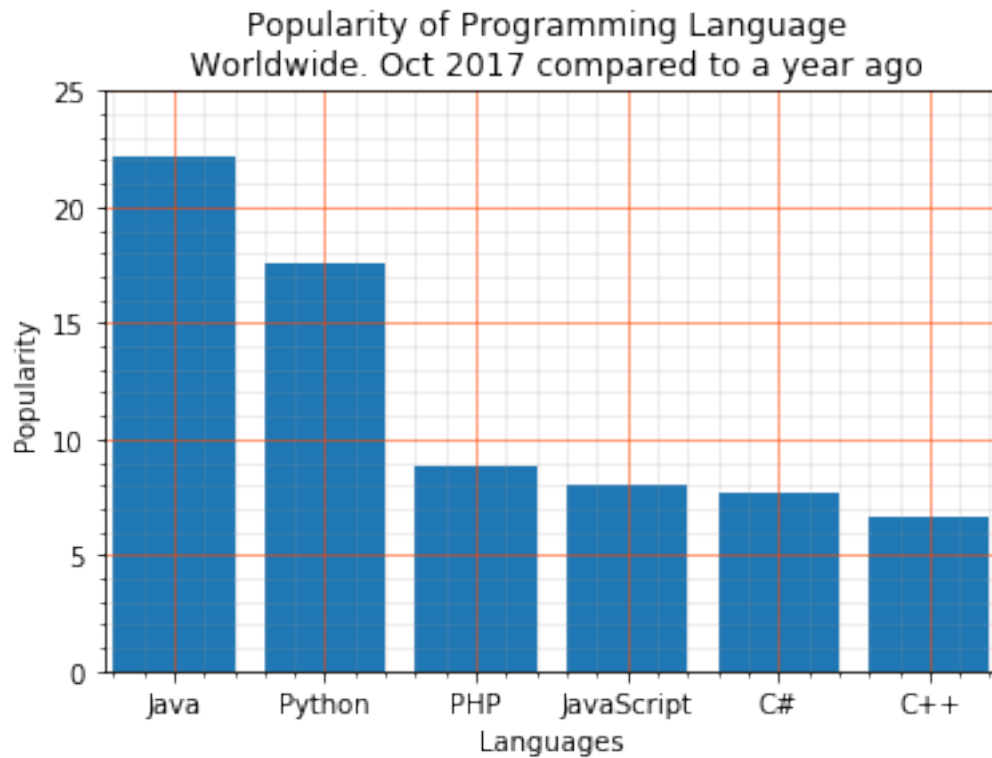
Languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
Popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

index = np.arange(len(Languages))
plt.bar(index, Popularity)
plt.xlabel('Languages', fontsize=10)
plt.ylabel('Popularity', fontsize=10)
plt.ylim(0, 25)
plt.xticks(index, Languages, fontsize=10, rotation=0)
plt.title('Popularity of Programming Language \n Worldwide. Oct 2017 compared to a year ago')
# Show the major grid lines with dark grey lines
plt.grid(b=True, which='major', color='#FF4500', linestyle='-', linewidth=0.5) #linestyle='solid'

#plt.grid(b=True, which='major', color='#666666', linestyle='-')

# Show the minor grid lines with very faint and almost transparent grey lines
plt.minorticks_on()
```

```
plt.grid(b=True, which='minor', color='#999999', linestyle='-', alpha=0.2)
plt.margins(0.01)
plt.show()
```

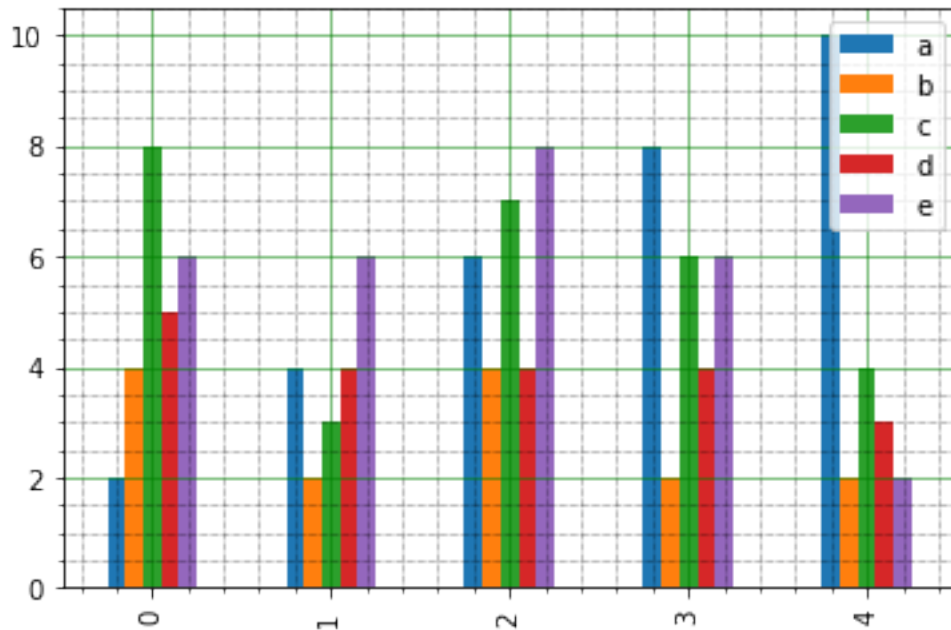


Q147. Write a python program to create the bar plot from a DataFrame.

```
In [25]: import matplotlib.pyplot as plt
df = pd.DataFrame({'a': [2,4,6,8,10],
                   'b': [4,2,4,2,2],
                   'c': [8,3,7,6,4],
                   'd': [5,4,4,4,3],
                   'e': [6,6,8,6,2]})

df
df.plot(kind='bar')
plt.minorticks_on()
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')

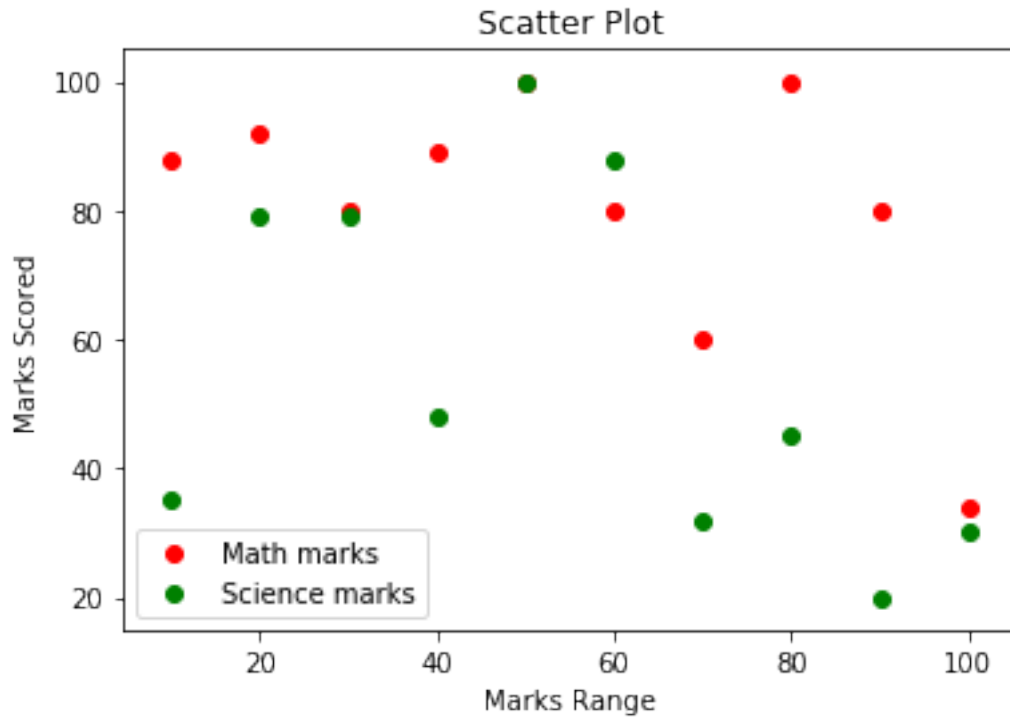
plt.show()
```

Q148. Write a python program to draw the scatter plot comparing two subject marks of Mathematics and Science. Use marks of 10 students.

```
In [26]: import matplotlib.pyplot as plt
import pandas as pd
math_marks = [88, 92 , 80, 89, 100, 80, 60, 100, 80, 34]
science_marks = [35, 79, 79 , 48, 100, 88, 32, 45, 20, 30]
marks_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

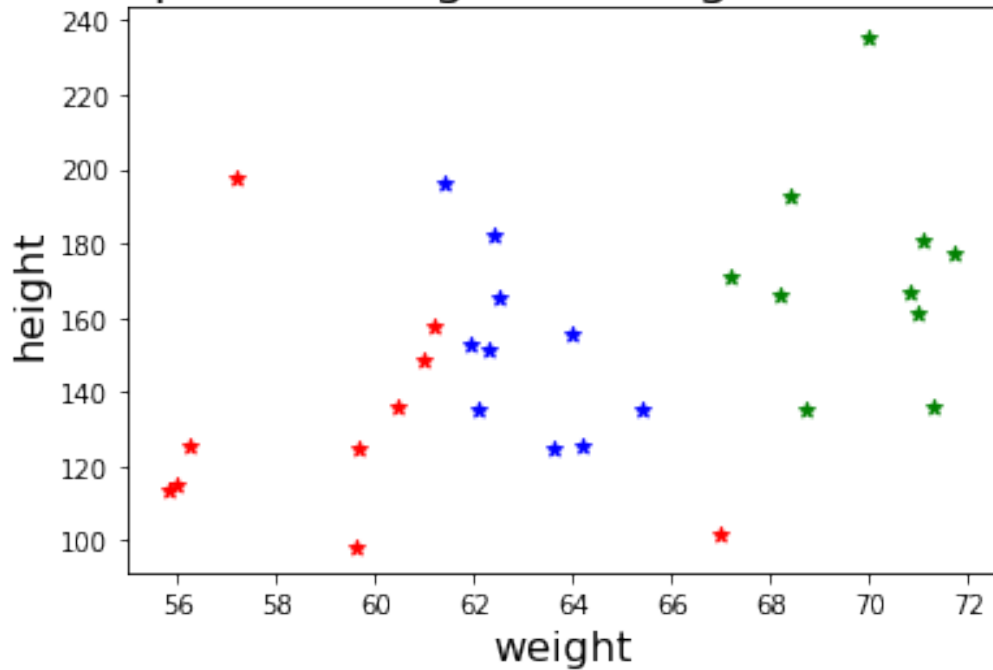
plt.scatter(marks_range, math_marks, label='Math marks', color='r')
plt.scatter(marks_range, science_marks, label='Science marks', color='g')
plt.title('Scatter Plot')
plt.xlabel('Marks Range')
plt.ylabel('Marks Scored')
plt.legend()
plt.show()
```



Q149. Write a python program to draw the scatter plot for three different groups comparing weights and heights.

```
In [4]: import matplotlib.pyplot as plt
import numpy as np
colormap = ['red', 'green', 'blue']
weight1=[67,57.2,59.6,59.64,55.8,61.2,60.45,61,56.23,56]
height1=[101.7,197.6,98.3,125.1,113.7,157.7,136,148.9,125.3,114.9]
weight2=[61.9,64,62.1,64.2,62.3,65.4,62.4,61.4,62.5,63.6]
height2=[152.8,155.3,135.1,125.2,151.3,135,182.2,195.9,165.1,125.1]
weight3=[68.2,67.2,68.4,68.7,71,71.3,70.8,70,71.1,71.7]
height3=[165.8,170.9,192.8,135.4,161.4,136.1,167.1,235.1,181.1,177.3]
weight=np.concatenate((weight1,weight2,weight3))
height=np.concatenate((height1,height2,height3))
plt.scatter(weight1, height1, marker='*', color = 'r')
plt.scatter(weight2, height2, marker='*', color = 'b')
plt.scatter(weight3, height3, marker='*', color = 'g')
plt.xlabel('weight', fontsize=16)
plt.ylabel('height', fontsize=16)
plt.title('Group wise Weight vs Height scatter plot',fontsize=20)
plt.show()
```

Group wise Weight vs Height scatter plot



Q150. Write a python program to draw a scatter plot to find sea-level rise in past 100 years.

```
In [37]: d = pd.read_csv('sealevel.csv')
Year = d['Year']
sea_levels = d['CSIRO - Adjusted sea level (inches)']
plt.scatter(Year, sea_levels, edgecolors='r')
plt.xlabel('Year')
plt.ylabel('Sea Level (inches)')
plt.title('Rise in Sealevel')
plt.show()
```

