This is the basics of Python and data analysis with pandas, performed by Aryan

Variable in python

```
# variable carries and contain
x = 22
# printing variable
print(x)
```

```
22
```

```
# now , let's check it's type
type(x)
```

```
int
```

```
# variable can also contain string
y = ' mint chocolate chip '
print(y)
```

```
 mint chocolate chip
```

```
# checking it's type
type(y)
```

```
str
```

```
# We can also change variables as we see necessary
y = ' vanilla '
print(y)
```

```
 vanilla
```

```
# variables are case sensitive
Y = ' choco '
print(y)
```

```
 vanilla
```

```
# We can also assign multiple values to multiple variables
x, y, z = 'ryukendo', 'jungle fury', 'dino hunter'
print(x)
print(y)
print(z)
```

```
ryukendo
jungle fury
dino hunter
```

```python
# We can also assign multiple variables to one value
x=y=z='pokemon'
print(x)
print(y)
print(z)
```

```
pokemon
pokemon
pokemon
```

```python
# We can also assign list , tuples, dictionary, and sets to variables
as well
tv_show = ['ben 10', 'oggy and the cockroaches', 'demon' ] #list
print(tv_show)
x,y,z = tv_show
print(x)
print(y)
print(z)
```

```
['ben 10', 'oggy and the cockroaches', 'demon']
ben 10
oggy and the cockroaches
demon
```

```python
# Naming variable convention ( these are best possible way to
represent variables )

# Camel Case
# Test variable Case

testVariableCase = 'vanilla swirl'

# Pascal Case
# Test variable Case

TestVariableCase = 'vanilla swirl'

# Snake Case
# Test variable Case

test_variable_case = 'vanilla swirl'

# Can write in these ways

# testvar = 'Vanilla Swirl'
# test_var = 'Vanilla Swirl'
# _test_var = 'Vanilla Swirl'
```

```
# testVar = 'Vanilla Swirl'
# TestVar = 'Vanilla Swirl'
# testVar2 = 'Vanilla Swirl'


# Cannot write in these way

# 2testVar = 'Vanilla Swirl'
# test-Var2 = 'Vanilla Swirl'
# test Var2 = 'Vanilla Swirl'
# test,Var2 = 'Vanilla Swirl'


# We can use plus sign in variables
x = 'Best ben 10 alien in alien x' + '.'
print(x)

# We cannot add string and a integer
# cannot do : x = 'Best ben 10 alien in alien x' + 2
# But we can do ,
x = 'Best ben 10 alien in alien x' + str(2)
print(x)
```

```
Best ben 10 alien in alien x.
Best ben 10 alien in alien x2
```

```
# We can add multiple variables in the print statement.
x = 'Pokemon'
y = ' is'
z = ' my fav'

print(x+y+z)
```

```
Pokemon is my fav
```

```
# we can separate string and integers with comma in print statement
x = 'ben'
y = 10
print(x,y)
```

```
ben 10
```

Data Types

```
# main datatypes: numeric, boolean,strings, lists,Tuples,
Sets,Dictionaries

# numeric: integers, float, complex
print(type(12))
```

```python
print(type(12 + 10.25))
print(type(12 + 3j))
```

```
<class 'int'>
<class 'float'>
<class 'complex'>
```

```python
# boolean: True, False
print(type(True))
print(type(False))
# ex :
print(type(1>5))
print(1>5)
```

```
<class 'bool'>
<class 'bool'>
<class 'bool'>
False
```

```python
# Strings

print('Single Quote')
print("Double Quote")
print("""
multi
line
""")
# For use case, you cannot write 'I've always wanted to eat Doritos'
# Instead of " I've always wanted to eat Dorritos"

# Strings can be indexed
a = 'Hello World'
print(a[6])
print(a[-4])
print(a[2:7])

# Strings can be multiplied by number , or could be added into one another
print(a*4)
print(a+a)
```

```
Single Quote
Double Quote

multi
line

W
o
llo W
```

```
Hello WorldHello WorldHello WorldHello World
Hello WorldHello World
```

```python
# Lists: It could be of one data type , or multiple data type
[1,2,3]
['tynewname' , 'aryanislegend' , 'gami world']
['tynewname' , 8 , [1,2,3], True ]

#It could be put into an variable

showdown_ids = ['tynewname' , 'aryanislegend' , 'gami world']
showdown_ids.append('ashserebii') # append use to add into a list
showdown_ids
print(type(showdown_ids))

# We can also change the list

showdown_ids[0] = 'butterscorch'
showdown_ids

# list could be nested and put into an variable

nested_list = ['tynewname' , 8 , [1,2,3], True ]

# Indexing within indexing within indexing

nested_list[1:3][1][2]
```

```
<class 'list'>
```

```
3
```

```python
# Tuples : It's cannot be modified or changed after being made
tuple_ids = (12 , 34 , 67 , 78)
print(type(tuple_ids))
tuple_ids[0]
# cannot do tuple_ids.append()
```

```
<class 'tuple'>
```

```
12
```

```python
# Sets : Don't have any duplicate elements . It does'nt have index ,
it's unordered

daily_rona = {1 , 34 ,56 , 34 , 5}
print(type(daily_rona))
daily_rona_hai = {1,2,3,3,4,5,6,7,8,8,8,9,0,0,1,3,4}
print(daily_rona_hai)

# generally we use sets to compare values
girl = {1,2,3,4,4,4,5,6,7}
```

```
boy = {4,5,6,6,6,7,8,9,0,0}
print(girl | boy) # | used to show combined unique values
print(girl & boy) # & shows what matches in both sets
print(girl - boy) # - shows what does'nt matches
print(girl ^ boy) # ^ shows value in one or other but not in both
```

```
<class 'set'>
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
{4, 5, 6, 7}
{1, 2, 3}
{0, 1, 2, 3, 8, 9}
```

```
# dictionaries : key/Value pair

dict_pokemon = {'name':'primape' , 'type':'fighting' , 'weak_to':
['psychic','fairy','flying']}
print(type(dict_pokemon))
print(dict_pokemon)
dict_pokemon.values() # shows values inside dictionary
dict_pokemon.keys()   # shows keys of dictionary
dict_pokemon.items()  # shows both keys and values

# can't do indexing like dict_pokemon[0] as it will not return this
'name':'primape', it will show err
# we can call dictionary by using keys
dict_pokemon['name']
# it will not work :dict_pokemon['primape']

# we can also update dictionary

dict_pokemon['name'] = 'Machoke'
print(dict_pokemon)

# we can also update all value at a time
dict_pokemon.update({'name':'primape' , 'type':'fighting' , 'strong
to' : ['steel','rock','dark'],
                     'weak to':['psychic','fairy','flying']})
print(dict_pokemon)
## please carefully see the update function
```

```
<class 'dict'>
{'name': 'primape', 'type': 'fighting', 'weak_to': ['psychic',
'fairy', 'flying']}
{'name': 'Machoke', 'type': 'fighting', 'weak_to': ['psychic',
'fairy', 'flying']}
{'name': 'primape', 'type': 'fighting', 'weak_to': ['psychic',
'fairy', 'flying'], 'strong to': ['steel', 'rock', 'dark'], 'weak to':
['psychic', 'fairy', 'flying']}
```

Comparison , logical and Membership operaator in python

```
# Comparison operator: Use to compare
# ==
print(1,10==10)
print(2,10==50)
x = 'vanilla'
y = 'chocolate'
print(3,x==y)
# !=
print(4,10!=10)
print(5,10!=50)
print(6,x!=y)
# <
print(7,10<10)
print(8,10<50)
# >
print(9,10>10)
print(10,10>50)
```

```
1 True
2 False
3 False
4 False
5 True
6 True
7 False
8 True
9 False
10 False
```

```
# Logical operators: for logical ans
# and
print(1,(10 > 50) and (50 > 10))
print(2,(70 > 50) and (50 > 10))
# or
print(3,(10 > 50) or (50 > 10))
print(4,(70 > 50) or (50 > 10))
# not
print(5,not(50>10))
```

```
1 False
2 True
3 True
4 True
5 False
```

```python
# Membership operator : If something is within something
new_pokemon = 'pokemon ZA is not coming this year'
print('is' in new_pokemon)
print('is' not in new_pokemon)
```

```
True
False
```

If-Elif-Else Statements

```python
# it's like: if this then this , if not , then this , if not , then anything else
if 25 > 10:
    print('It worked!')
```

```
It worked!
```

```python
if 25 < 10:
    print('It worked!')
else:
    print('It didnt worked!....')
```

```
It didnt worked!....
```

```python
# If we wants to try lots of different conditions , then ,
if 25 < 10:
    print('It worked!')
elif 25 < 30:
    print('nah , I would win')
else:
    print('It didnt worked!....')
```

```
nah , I would win
```

```python
# If else statement in one line
print('It worked') if 10>30 else print('It did not worked...')
```

```
It did not worked...
```

```python
# we can also do nested loop

if (25 < 10) or (1 < 3):
    print('It worked!')
    if 10 > 5:
        print('This nested if statement worked!')
elif 25 < 20:
    print('elif worked!')
elif 25 < 21:
    print('elif 2 worked!')
elif 25 < 40:
```

```
    print('elif 3 worked!')
elif 25 < 50:
    print('elif 4 worked!')
else:
    print('It did not work...')
```

```
It worked!
This nested if statement worked!
```

For loops

```
# for loops : use to iterate over a sequence . It does not need
condition.
integer = [1,2,3,4,5] # could be a list , tuple , string ...

for number in integer:
    print(number)
```

```
1
2
3
4
5
```

```
for number in integer:
    print('What a waste of talent...')
```

```
What a waste of talent...
What a waste of talent...
What a waste of talent...
What a waste of talent...
What a waste of talent...
```

```
for jeffery in integer:
    print(jeffery + jeffery)
```

```
2
4
6
8
10
```

```
# we can iterate over dictionary
new_dict = {'name':'aryan' , 'weekly_money_use':'500' ,
'fav_pokemon_poison':['toxicroak' , 'merciless']}

for pokemon in new_dict.values(): # In dictionary we have to specify
what we have to pull
    print(pokemon)
```

```
aryan
500
['toxicroak', 'merciless']
```

```python
for key , value in new_dict.items(): # In dictionary we can also do both key and values
    print(key,'->>',value)
```

```
name ->> aryan
weekly_money_use ->> 500
fav_pokemon_poison ->> ['toxicroak', 'merciless']
```

```python
# Nested for loop
pokemons_types_1 = ['fighting' , 'psychic' , 'dark']
pokemons_types_2 = ['fire' , 'water' , 'grass']
for one in pokemons_types_1:
    for two in pokemons_types_2:
        print(one,',',two)
```

```
fighting , fire
fighting , water
fighting , grass
psychic , fire
psychic , water
psychic , grass
dark , fire
dark , water
dark , grass
```

While loop

```python
# while loop : use to iterate as long as text condition is true . It need condition.
number = 1
while number<=10:
    print(number)
    number = number + 1
```

```
1
2
3
4
5
6
7
8
9
10
```

```python
number = 1
while number<=10:
    print(number)
    if number == 3:
        break          # with break statement , we can stop the loop even when while condition is true
    number = number + 1
```

```
1
2
3
```

```python
number = 1
while number<=10:
    print(number)
    if number == 11:
        break
    number = number + 1
else:   # works when condition of while statement fallen
    print(' no longer small')
```

```
1
2
3
4
5
6
7
8
9
10
 no longer small
```

```python
# continue statement : if it triggers , then it rejects all remaining statements in current iteration of loop and we go to next iteration
number = 0
while number<=10:
    number = number + 1
    if number == 7:
        continue
    print(number)
else:   # it rejected 7
    print(' no longer small')
```

```
1
2
3
4
5
6
```

```
8
9
10
11
 no longer small
```

Functions

```
# functions : it is a block of code , only runs when we calls it
def first_fun():
    print('you are not good enough')

first_fun()

you are not good enough

# taking arguments in functions
def number_cubed(number):
    print(number**3)
number_cubed(69)

328509

# we can also take multiple arguments
def number_powered(number , power):
    print(number**power)
number_powered(69,2)

4761

# arbitrary arguments : If we don't know how many arguments to pass ,
we can specify it while calling
# this is how it look like : def number_args(*args):
def number_args(*number):
    print(number[0]*number[1])
number_args(5,6,3,4,5)

30

# working while makeing tuple outside will not work
args_tuple = (67,78,5,6,7)
# number_args(args_tuple) # It will not work , we have to specify it
(unpack it)
number_args(*args_tuple)


5226

# keywords arguments
def number_powered(number , power):
```

```
    print(number**power)
number_powered(power = 5 , number = 3) # using keywords
```

```
243
```

```
# arbitrary keyword argument (we have to use **
def number_kwarg(**number):
    print('My lucky number: ' + str(number['integer']))
# if we don't call using keyword and  just  keep number[] , it will
not work
number_kwarg(integer = 2456)
```

```
My lucky number: 2456
```

Converting Data types

```
num1 , num2 = 7 , '7'
print(type(num1))
print(type(num2))
# cannot do : num1 + num2
# for adding those ,
print(num1 + int(num2))
```

```
<class 'int'>
<class 'str'>
14
```

```
# converting lists , sets and tuples
list_type = [1,2,3]
print(type(list_type))
print(type(tuple(list_type)))
print(list_type)
print(tuple(list_type))

list_type = [1,2,3,3,3,4,4,5,6,6]
print(set(list_type)) # beware while changing as it may remove
duplicates
```

```
<class 'list'>
<class 'tuple'>
[1, 2, 3]
(1, 2, 3)
{1, 2, 3, 4, 5, 6}
```

```
# for dictionaries
dict_type = {'name':'ary' , 'age':290}
print(type(dict_type))
print(dict_type.items())
print(dict_type.values())
```

```
print(dict_type.keys())
print(list(dict_type.keys()))
```

```
<class 'dict'>
dict_items([('name', 'ary'), ('age', 290)])
dict_values(['ary', 290])
dict_keys(['name', 'age'])
['name', 'age']
```

```
# also for strings
live_long = 'I like to party'
list(live_long)
```

```
['I', ' ', 'l', 'i', 'k', 'e', ' ', 't', 'o', ' ', 'p', 'a', 'r', 't',
'y']
```

Reading In files

```
import pandas as pd # importing pandas library and giving it an alias
import numpy as np # importing numpy and giving it a alias

# first we will learn how to do reading in different files type :
csv , JSON , text , .xlsx

#reading csv
df = pd.read_csv(r"C:\Users\Aryan\Desktop\New folder (2)\countries of
the world csv.csv")
# r in start as to read it as a raw text / string without \ and other
stuffs
# Note : use shift + tab to see details

# we can also specify header (main point to be noted) and their names
pd.read_csv(r"C:\Users\Aryan\Desktop\New folder (2)\countries of the
world csv.csv" , header = None ,
           names =['Count' , 'Reg'] )
```

```
               Count                                 Reg
0            Country                              Region
1        Afghanistan         ASIA (EX. NEAR EAST)
2            Albania   EASTERN EUROPE
3            Algeria   NORTHERN AFRICA
4     American Samoa   OCEANIA
..               ...                                 ...
223        West Bank   NEAR EAST
224   Western Sahara   NORTHERN AFRICA
225            Yemen   NEAR EAST
226           Zambia   SUB-SAHARAN AFRICA
227         Zimbabwe   SUB-SAHARAN AFRICA
```

```
[228 rows x 2 columns]

# reading texts
df = pd.read_csv(r"C:\Users\Aryan\Desktop\New folder (2)\countries of
the world.txt")
df # we need to use separator as \t need to be separated , so,
df = pd.read_csv(r"C:\Users\Aryan\Desktop\New folder (2)\countries of
the world.txt",sep = '\t')
df
# other way is
df = pd.read_table(r"C:\Users\Aryan\Desktop\New folder (2)\countries
of the world.txt")
df
# for reading it as a csv , change .txt into .csv and use separator ,
but do use it
# df = pd.read_csv(r"C:\Users\Aryan\Desktop\New folder (2)\countries
of the world.csv",sep = ',')

              Country                               Region
0        Afghanistan         ASIA (EX. NEAR EAST)
1            Albania   EASTERN EUROPE
2            Algeria   NORTHERN AFRICA
3     American Samoa   OCEANIA
4            Andorra   WESTERN EUROPE
..               ...                                   ...
222        West Bank   NEAR EAST
223   Western Sahara   NORTHERN AFRICA
224            Yemen   NEAR EAST
225           Zambia   SUB-SAHARAN AFRICA
226         Zimbabwe   SUB-SAHARAN AFRICA

[227 rows x 2 columns]

# JSON files
df = pd.read_json(r"C:\Users\Aryan\Desktop\New folder (2)\
json_sample.json")
df

                                            12 Strong  \
0  {'Genre': 'Action', 'Gross': '$453,173', 'IMDB...

            A Fantastic Woman (Una Mujer Fantástica)  \
0  {'popcornscore': 83, 'rating': 'R', 'tomatosco...

                          All The Money In The World  \
0  {'popcornscore': 71, 'rating': 'R', 'tomatosco...

                          Bilal: A New Breed Of Hero  \
0  {'popcornscore': 91, 'rating': 'PG13', 'tomato...
```

```
                                    Call Me By Your Name  \
0  {'popcornscore': 87, 'rating': 'R', 'tomatosco...   

                                            Darkest Hour  \
0  {'popcornscore': 84, 'rating': 'PG13', 'tomato...   

                                          Den Of Thieves  \
0  {'Genre': 'Action', 'Gross': '$491,898', 'IMDB...   

                                               Ferdinand  \
0  {'popcornscore': 49, 'rating': 'PG', 'tomatosc...   

                                      Fifty Shades Freed  \
0  {'Genre': 'Drama', 'Gross': 'unknown', 'IMDB M...   

                      Film Stars Don'T Die In Liverpool  ...  \
0  {'popcornscore': 69, 'rating': 'R', 'tomatosco...  ...   

                                      The 15:17 To Paris  \
0  {'Genre': 'Drama', 'Gross': 'unknown', 'IMDB M...   

                                            The Commuter  \
0  {'popcornscore': 48, 'rating': 'PG13', 'tomato...   

                                     The Disaster Artist  \
0  {'popcornscore': 89, 'rating': 'R', 'tomatosco...   

                                    The Greatest Showman  \
0  {'Genre': 'Biography', 'Gross': '$627,248', 'I...   

                                  The Insult (L'Insulte)  \
0  {'popcornscore': 86, 'rating': 'R', 'tomatosco...   

                                                The Post  \
0  {'Genre': 'Biography', 'Gross': '$463,228', 'I...   

                                      The Shape Of Water  \
0  {'Genre': 'Adventure', 'Gross': '$448,287', 'I...   

               Three Billboards Outside Ebbing, Missouri  \
0  {'popcornscore': 87, 'rating': 'R', 'tomatosco...   

                                Till The End Of The World  \
0  {'popcornscore': -1, 'rating': 'NR', 'tomatosc...   

                                              Winchester  
0  {'Genre': 'Biography', 'Gross': '$696,786', 'I...  

[1 rows x 38 columns]
```

```python
# Excel files : here we can specify which sheet we need to study
df = pd.read_excel(r"C:\Users\Aryan\Desktop\New folder (2)\
world_population_excel_workbook.xlsx"
                  ,sheet_name = 'Sheet1') # specifying sheet name
df
```

```
     Rank CCA3            Country           Capital
0      36  AFG        Afghanistan             Kabul
1     138  ALB            Albania            Tirana
2      34  DZA            Algeria           Algiers
3     213  ASM     American Samoa         Pago Pago
4     203  AND            Andorra  Andorra la Vella
..    ...  ...                ...               ...
229   226  WLF  Wallis and Futuna          Mata-Utu
230   172  ESH     Western Sahara         El AaiÃºn
231    46  YEM              Yemen             Sanaa
232    63  ZMB             Zambia            Lusaka
233    74  ZWE           Zimbabwe            Harare

[234 rows x 4 columns]
```

```python
# for viewing all data in the files , we can use set_option
# pd.set_option('display.max.rows' , 235)   # for watching all rows
# pd.set_option('display.max.columns' , 40) # for watching all columns

# If we want to know more about df
df.info() # It's a method so parenthesis needed
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 234 entries, 0 to 233
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Rank     234 non-null    int64 
 1   CCA3     234 non-null    object
 2   Country  234 non-null    object
 3   Capital  234 non-null    object
dtypes: int64(1), object(3)
memory usage: 7.4+ KB
```

```python
# for shape
df.shape # it's an attribute so no parenthesis needed
```

```
(234, 4)
```

```python
df.head(4) # we can get first some values
```

```
   Rank CCA3         Country    Capital
0    36  AFG     Afghanistan      Kabul
1   138  ALB         Albania     Tirana
```

```
2    34  DZA         Algeria    Algiers
3   213  ASM  American Samoa  Pago Pago
```

```
df.tail(3) # we can get last some values
```

```
     Rank CCA3   Country Capital
231    46  YEM     Yemen   Sanaa
232    63  ZMB    Zambia  Lusaka
233    74  ZWE  Zimbabwe  Harare
```

```
# for fetching a single data , use df[''] or df.
df['Rank']
```

```
0       36
1      138
2       34
3      213
4      203
      ...
229    226
230    172
231     46
232     63
233     74
Name: Rank, Length: 234, dtype: int64
```

```
# we will know about loc and iloc in later stages
```

Pandas Series and DataFrame

```
# converting a list into an array
my_list = [23 , 45 , 67]
x = np.array(my_list)
print(x)
print(type(x))
```

```
[23 45 67]
<class 'numpy.ndarray'>
```

```
# now fetching this array as a series
y = pd.Series(my_list)
print(y)
print(type(y))
```

```
0    23
1    45
2    67
dtype: int64
<class 'pandas.core.series.Series'>
```

```python
# Now let's see what happens after we add 2 series
index_value = ['best' , '2nd Best' , '3rd best']
my_list = [10,25,35]
one = pd.Series(data = my_list , index = index_value)
print(one)
print('\n')
index_value = ['best' , '2nd Best' , '4th best']
my_list = [10,25,40]
two = pd.Series(data = my_list , index = index_value)
print(two)
print('\n')
print(one + two) # It only takes common columns and add those ,
otherwise it will show NaN
print(type(one+two))
```

```
best        10
2nd Best    25
3rd best    35
dtype: int64


best        10
2nd Best    25
4th best    40
dtype: int64


2nd Best    50.0
3rd best     NaN
4th best     NaN
best        20.0
dtype: float64
<class 'pandas.core.series.Series'>
```

```python
# Creating dataframe and its different methods
new_starters = [['osshowatt','water'] , ['cyndaquil' , 'fire'] ,
['hoothoot','flying']]
starters_data = pd.DataFrame(new_starters)
starters_data # it does'nt contain a column name
```

```
           0       1
0  osshowatt   water
1  cyndaquil    fire
2   hoothoot  flying
```

```python
starters_data = pd.DataFrame(new_starters , columns=('Starters' ,
'Type'))
starters_data # Columns added but index can also be added
```

```
    Starters    Type
0  osshowatt   water
```

```
1  cyndaquil    fire
2   hoothoot  flying
```

```
starters_data = pd.DataFrame(new_starters , columns=('Starters' ,
'Type') ,
                             index = ['Pokemon 1','Pokemon 2' ,
'Pokemon 3']) # index added
starters_data
print(type(starters_data))
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
# other way to add to the dataframe is
pokemon = ['Osshowatt' , 'Bulbasaur' , 'chimchar']
ranking = [1,2,3]
df_starters = {'Starters':pokemon , 'rank':ranking}
df_starters = pd.DataFrame(df_starters , index = ['Pokemon 1' ,
'Pokemon 2' , 'Pokemon 3'])
df_starters
```

```
            Starters  rank
Pokemon 1  Osshowatt     1
Pokemon 2  Bulbasaur     2
Pokemon 3   chimchar     3
```

Filtering and Ordering

```
df = pd.read_csv(r"D:\gami\world_population (1).csv")
df
```

```
     Rank CCA3            Country           Capital Continent  \
0      36  AFG        Afghanistan             Kabul      Asia
1     138  ALB            Albania            Tirana    Europe
2      34  DZA            Algeria           Algiers    Africa
3     213  ASM     American Samoa         Pago Pago   Oceania
4     203  AND            Andorra  Andorra la Vella    Europe
..    ...  ...                ...               ...       ...
229   226  WLF  Wallis and Futuna          Mata-Utu   Oceania
230   172  ESH     Western Sahara          El Aaiún    Africa
231    46  YEM              Yemen             Sanaa      Asia
232    63  ZMB             Zambia            Lusaka    Africa
233    74  ZWE           Zimbabwe            Harare    Africa

     2022 Population  2020 Population  2015 Population  2010
Population  \
0         41128771.0       38972230.0       33753499.0
28189672.0
1          2842321.0        2866849.0        2882481.0
2913399.0
```

```
2        44903225.0       43451666.0       39543154.0
35856344.0
3           44273.0          46189.0          51368.0
54849.0
4           79824.0          77700.0          71746.0
71519.0
..              ...              ...              ...               ..
.
229         11572.0          11655.0          12182.0
13142.0
230        575986.0         556048.0         491824.0
413296.0
231      33696614.0       32284046.0       28516545.0
24743946.0
232      20017675.0       18927715.0              NaN
13792086.0
233      16320537.0       15669666.0       14154937.0
12839771.0

     2000 Population  1990 Population  1980 Population  1970
Population  \
0        19542982.0       10694796.0       12486631.0
10752971.0
1         3182021.0        3295066.0        2941651.0
2324731.0
2        30774621.0       25518074.0       18739378.0
13795915.0
3           58230.0          47818.0          32886.0
27075.0
4           66097.0          53569.0          35611.0
19860.0
..              ...              ...              ...               ..
.
229         14723.0          13454.0          11315.0
9377.0
230        270375.0         178529.0         116775.0
76371.0
231      18628700.0       13375121.0        9204938.0
6843607.0
232       9891136.0        7686401.0        5720438.0
4281671.0
233      11834676.0       10113893.0        7049926.0
5202918.0

     Area (km²)  Density (per km²)  Growth Rate  World Population
Percentage
0      652230.0            63.0587       1.0257
0.52
1       28748.0            98.8702       0.9957
```

```
0.04
2     2381741.0            18.8531       1.0164
0.56
3         199.0           222.4774       0.9831
0.00
4         468.0           170.5641       1.0100
0.00
..          ...                ...          ...
...
229       142.0            81.4930       0.9953
0.00
230    266000.0             2.1654       1.0184
0.01
231    527968.0            63.8232       1.0217
0.42
232    752612.0            26.5976       1.0280
0.25
233    390757.0            41.7665       1.0204
0.20

[234 rows x 17 columns]

# Filtering
df[df['Rank'] <= 10] # for filtering , as we can see , we have to
write condition inside df

     Rank CCA3        Country            Capital      Continent  \
16      8  BGD     Bangladesh              Dhaka           Asia
27      7  BRA         Brazil           Brasilia  South America
41      1  CHN          China            Beijing           Asia
92      2  IND          India          New Delhi           Asia
93      4  IDN      Indonesia            Jakarta           Asia
131    10  MEX         Mexico        Mexico City  North America
149     6  NGA        Nigeria              Abuja         Africa
156     5  PAK       Pakistan          Islamabad           Asia
171     9  RUS         Russia             Moscow         Europe
221     3  USA  United States  Washington, D.C.  North America

     2022 Population  2020 Population  2015 Population  2010
Population  \
16      1.711864e+08     1.674210e+08     1.578300e+08
1.483911e+08
27      2.153135e+08     2.131963e+08     2.051882e+08
1.963535e+08
41      1.425887e+09     1.424930e+09     1.393715e+09
1.348191e+09
92      1.417173e+09     1.396387e+09     1.322867e+09
1.240614e+09
93      2.755013e+08     2.718580e+08     2.590920e+08
2.440162e+08
```

```
131     1.275041e+08     1.259983e+08     1.201499e+08
1.125324e+08
149     2.185412e+08     2.083274e+08     1.839958e+08
1.609529e+08
156     2.358249e+08     2.271967e+08     2.109693e+08
1.944545e+08
171     1.447133e+08     1.456173e+08     1.446684e+08
1.432426e+08
221     3.382899e+08     3.359420e+08     3.246078e+08
3.111828e+08

     2000 Population  1990 Population  1980 Population  1970
Population  \
16      1.291933e+08     1.071477e+08       83929765.0
67541860.0
27      1.758737e+08     1.507064e+08      122288383.0
96369875.0
41      1.264099e+09     1.153704e+09      982372466.0
822534450.0
92      1.059634e+09              NaN              NaN
557501301.0
93      2.140724e+08     1.821599e+08      148177096.0
115228394.0
131     9.787344e+07     8.172043e+07       67705186.0
50289306.0
149     1.228520e+08     9.521426e+07       72951439.0
55569264.0
156     1.543699e+08     1.154141e+08       80624057.0
59290872.0
171     1.468448e+08     1.480057e+08      138257420.0
130093010.0
221     2.823986e+08     2.480837e+08      223140018.0
200328340.0

     Area (km²)  Density (per km²)  Growth Rate  World Population
Percentage
16     147570.0          1160.0350       1.0108
2.15
27    8515767.0            25.2841       1.0046
2.70
41    9706961.0           146.8933       1.0000
17.88
92    3287590.0           431.0675       1.0068
17.77
93    1904569.0           144.6529       1.0064
3.45
131   1964375.0            64.9082       1.0063
1.60
149    923768.0           236.5759       1.0241
```

```
2.74
156    881912.0           267.4018       1.0191
2.96
171  17098242.0             8.4636       0.9973
1.81
221   9372610.0            36.0935       1.0038
4.24

# we can use isin function in pandas , same as in function in SQL
specific_countries = ['India' , 'Brazil']
df[df['Country'].isin(specific_countries)] # it's case sensitive ,
please watch

    Rank CCA3 Country    Capital      Continent  2022 Population  \
27     7  BRA  Brazil   Brasilia  South America     2.153135e+08
92     2  IND   India  New Delhi           Asia     1.417173e+09

    2020 Population  2015 Population  2010 Population  2000 Population
\
27     2.131963e+08     2.051882e+08     1.963535e+08     1.758737e+08

92     1.396387e+09     1.322867e+09     1.240614e+09     1.059634e+09


    1990 Population  1980 Population  1970 Population  Area (km²)  \
27      150706446.0      122288383.0       96369875.0   8515767.0
92              NaN              NaN      557501301.0   3287590.0

    Density (per km²)  Growth Rate  World Population Percentage
27            25.2841       1.0046                         2.70
92           431.0675       1.0068                        17.77

# we can also use contain function for same task, It works like LIKE
clause in SQL
df[df['Country'].str.contains('United')]

     Rank CCA3                       Country           Capital
Continent  \
219    97  ARE          United Arab Emirates         Abu Dhabi
Asia
220    21  GBR                United Kingdom            London
Europe
221     3  USA                 United States  Washington, D.C.  North
America
222   200  VIR  United States Virgin Islands  Charlotte Amalie  North
America

     2022 Population  2020 Population  2015 Population  2010
Population  \
219        9441129.0        9287289.0        8916899.0
8481771.0
```

```
220       67508936.0       67059474.0       65224364.0
62760039.0
221      338289857.0      335942003.0      324607776.0
311182845.0
222          99465.0         100442.0         102803.0
106142.0

     2000 Population  1990 Population  1980 Population  1970
Population  \
219        3275333.0        1900151.0        1014048.0
298084.0
220       58850043.0       57210442.0       56326328.0
55650166.0
221      282398554.0      248083732.0      223140018.0
200328340.0
222         108185.0         100685.0          96640.0
63446.0

     Area (km²)  Density (per km²)  Growth Rate  World Population
Percentage
219     83600.0           112.9322       1.0081
0.12
220    242900.0           277.9289       1.0034
0.85
221   9372610.0            36.0935       1.0038
4.24
222       347.0           286.6427       0.9937
0.00

# We can also filter on the base of index , main ways : 1) filter ,
2)loc/iloc
# we can set index whenever we wants
df2 = df.set_index('Country')
df2

                   Rank CCA3           Capital Continent  2022
Population  \
Country

Afghanistan          36  AFG             Kabul      Asia
41128771.0
Albania             138  ALB            Tirana    Europe
2842321.0
Algeria              34  DZA           Algiers    Africa
44903225.0
American Samoa      213  ASM         Pago Pago   Oceania
44273.0
Andorra             203  AND  Andorra la Vella    Europe
79824.0
...                 ...  ...               ...       ...
```

```
...
Wallis and Futuna   226  WLF          Mata-Utu   Oceania
11572.0
Western Sahara      172  ESH          El Aaiún    Africa
575986.0
Yemen                46  YEM             Sanaa      Asia
33696614.0
Zambia               63  ZMB            Lusaka    Africa
20017675.0
Zimbabwe             74  ZWE            Harare    Africa
16320537.0

                   2020 Population  2015 Population  2010
Population  \
Country

Afghanistan             38972230.0       33753499.0       28189672.0

Albania                  2866849.0        2882481.0        2913399.0

Algeria                 43451666.0       39543154.0       35856344.0

American Samoa             46189.0          51368.0          54849.0

Andorra                    77700.0          71746.0          71519.0

...                            ...              ...              ...

Wallis and Futuna          11655.0          12182.0          13142.0

Western Sahara            556048.0         491824.0         413296.0

Yemen                   32284046.0       28516545.0       24743946.0

Zambia                  18927715.0              NaN       13792086.0

Zimbabwe                15669666.0       14154937.0       12839771.0


                   2000 Population  1990 Population  1980
Population  \
Country

Afghanistan             19542982.0       10694796.0       12486631.0

Albania                  3182021.0        3295066.0        2941651.0

Algeria                 30774621.0       25518074.0       18739378.0

American Samoa             58230.0          47818.0          32886.0
```

```
Andorra                        66097.0           53569.0           35611.0

...                                ...               ...               ...

Wallis and Futuna              14723.0           13454.0           11315.0

Western Sahara                270375.0          178529.0          116775.0

Yemen                       18628700.0        13375121.0         9204938.0

Zambia                       9891136.0         7686401.0         5720438.0

Zimbabwe                    11834676.0        10113893.0         7049926.0


                   1970 Population  Area (km²)  Density (per km²)  \
Country
Afghanistan             10752971.0    652230.0            63.0587
Albania                  2324731.0     28748.0            98.8702
Algeria                 13795915.0   2381741.0            18.8531
American Samoa             27075.0       199.0           222.4774
Andorra                    19860.0       468.0           170.5641
...                            ...         ...                ...
Wallis and Futuna           9377.0       142.0            81.4930
Western Sahara             76371.0    266000.0             2.1654
Yemen                    6843607.0    527968.0            63.8232
Zambia                   4281671.0    752612.0            26.5976
Zimbabwe                 5202918.0    390757.0            41.7665

                   Growth Rate  World Population Percentage
Country
Afghanistan             1.0257                         0.52
Albania                 0.9957                         0.04
Algeria                 1.0164                         0.56
American Samoa          0.9831                         0.00
Andorra                 1.0100                         0.00
...                        ...                          ...
Wallis and Futuna       0.9953                         0.00
Western Sahara          1.0184                         0.01
Yemen                   1.0217                         0.42
Zambia                  1.0280                         0.25
Zimbabwe                1.0204                         0.20

[234 rows x 16 columns]
```

```python
df2.filter(items = ['Continent' , 'CCA3'] ,axis = 0)
```

```
Empty DataFrame
Columns: [Rank, CCA3, Capital, Continent, 2022 Population, 2020
Population, 2015 Population, 2010 Population, 2000 Population, 1990
Population, 1980 Population, 1970 Population, Area (km²), Density (per
```

```
km²), Growth Rate, World Population Percentage]
Index: []

df2.filter(items = ['Continent' , 'CCA3'] ,axis = 1) # default axis =
1

                  Continent CCA3
Country
Afghanistan            Asia  AFG
Albania              Europe  ALB
Algeria              Africa  DZA
American Samoa      Oceania  ASM
Andorra              Europe  AND
...                     ...  ...
Wallis and Futuna   Oceania  WLF
Western Sahara       Africa  ESH
Yemen                  Asia  YEM
Zambia               Africa  ZMB
Zimbabwe             Africa  ZWE

[234 rows x 2 columns]

df2.filter(items = ['Zimbabwe'] ,axis = 0) # axis 0 will fetch data

          Rank CCA3 Capital Continent  2022 Population  2020
Population  \
Zimbabwe    74  ZWE  Harare    Africa       16320537.0
15669666.0

          2015 Population  2010 Population  2000 Population  1990
Population  \
Zimbabwe       14154937.0       12839771.0       11834676.0
10113893.0

          1980 Population  1970 Population  Area (km²)  Density (per
km²)  \
Zimbabwe        7049926.0        5202918.0    390757.0
41.7665

          Growth Rate  World Population Percentage
Zimbabwe       1.0204                          0.2

# we can also use like
df2.filter(like = 'United' ,axis = 0)

                              Rank CCA3          Capital
Continent  \
Country

United Arab Emirates            97  ARE        Abu Dhabi
Asia
```

```
United Kingdom                     21  GBR            London
Europe
United States                       3  USA  Washington, D.C.  North
America
United States Virgin Islands      200  VIR  Charlotte Amalie  North
America

                              2022 Population  2020 Population  \
Country
United Arab Emirates                9441129.0        9287289.0
United Kingdom                     67508936.0       67059474.0
United States                     338289857.0      335942003.0
United States Virgin Islands          99465.0         100442.0

                              2015 Population  2010 Population  \
Country
United Arab Emirates                8916899.0        8481771.0
United Kingdom                     65224364.0       62760039.0
United States                     324607776.0      311182845.0
United States Virgin Islands         102803.0         106142.0

                              2000 Population  1990 Population  \
Country
United Arab Emirates                3275333.0        1900151.0
United Kingdom                     58850043.0       57210442.0
United States                     282398554.0      248083732.0
United States Virgin Islands         108185.0         100685.0

                              1980 Population  1970 Population  Area
(km²)  \
Country

United Arab Emirates                1014048.0         298084.0
83600.0
United Kingdom                     56326328.0       55650166.0
242900.0
United States                     223140018.0      200328340.0
9372610.0
United States Virgin Islands          96640.0          63446.0
347.0

                              Density (per km²)  Growth Rate  \
Country
United Arab Emirates                   112.9322       1.0081
United Kingdom                         277.9289       1.0034
United States                           36.0935       1.0038
United States Virgin Islands           286.6427       0.9937

                              World Population Percentage
Country
```

```
United Arab Emirates                                            0.12
United Kingdom                                                  0.85
United States                                                   4.24
United States Virgin Islands                                    0.00
```

```
# basics of loc and iloc
df2.loc['United States']
```

```
Rank                                               3
CCA3                                             USA
Capital                             Washington, D.C.
Continent                              North America
2022 Population                          338289857.0
2020 Population                          335942003.0
2015 Population                          324607776.0
2010 Population                          311182845.0
2000 Population                          282398554.0
1990 Population                          248083732.0
1980 Population                          223140018.0
1970 Population                          200328340.0
Area (km²)                                 9372610.0
Density (per km²)                            36.0935
Growth Rate                                   1.0038
World Population Percentage                     4.24
Name: United States, dtype: object
```

```
# we can use iloc as thinking that their is a numerical index inside
df2.iloc[3]
```

```
Rank                                     213
CCA3                                     ASM
Capital                            Pago Pago
Continent                            Oceania
2022 Population                      44273.0
2020 Population                      46189.0
2015 Population                      51368.0
2010 Population                      54849.0
2000 Population                      58230.0
1990 Population                      47818.0
1980 Population                      32886.0
1970 Population                      27075.0
Area (km²)                             199.0
Density (per km²)                   222.4774
Growth Rate                           0.9831
World Population Percentage              0.0
Name: American Samoa, dtype: object
```

```
# Now , let's starts ordering
df[df['Rank'] < 10].sort_values(by = 'Rank' , ascending = False) #
condition -> sorting -> order by -> ascending / desending
```

```
     Rank CCA3        Country           Capital      Continent  \
171     9  RUS         Russia            Moscow         Europe
16      8  BGD     Bangladesh             Dhaka           Asia
27      7  BRA         Brazil          Brasilia  South America
149     6  NGA        Nigeria             Abuja         Africa
156     5  PAK       Pakistan         Islamabad           Asia
93      4  IDN      Indonesia           Jakarta           Asia
221     3  USA  United States  Washington, D.C.  North America
92      2  IND          India         New Delhi           Asia
41      1  CHN          China           Beijing           Asia

     2022 Population  2020 Population  2015 Population  2010
Population  \
171     1.447133e+08     1.456173e+08     1.446684e+08
1.432426e+08
16      1.711864e+08     1.674210e+08     1.578300e+08
1.483911e+08
27      2.153135e+08     2.131963e+08     2.051882e+08
1.963535e+08
149     2.185412e+08     2.083274e+08     1.839958e+08
1.609529e+08
156     2.358249e+08     2.271967e+08     2.109693e+08
1.944545e+08
93      2.755013e+08     2.718580e+08     2.590920e+08
2.440162e+08
221     3.382899e+08     3.359420e+08     3.246078e+08
3.111828e+08
92      1.417173e+09     1.396387e+09     1.322867e+09
1.240614e+09
41      1.425887e+09     1.424930e+09     1.393715e+09
1.348191e+09

     2000 Population  1990 Population  1980 Population  1970
Population  \
171     1.468448e+08     1.480057e+08      138257420.0
130093010.0
16      1.291933e+08     1.071477e+08       83929765.0
67541860.0
27      1.758737e+08     1.507064e+08      122288383.0
96369875.0
149     1.228520e+08     9.521426e+07       72951439.0
55569264.0
156     1.543699e+08     1.154141e+08       80624057.0
59290872.0
93      2.140724e+08     1.821599e+08      148177096.0
115228394.0
221     2.823986e+08     2.480837e+08      223140018.0
200328340.0
92      1.059634e+09              NaN              NaN
557501301.0
```

```
41      1.264099e+09     1.153704e+09      982372466.0
822534450.0

     Area (km²)  Density (per km²)  Growth Rate  World Population
Percentage
171  17098242.0             8.4636       0.9973
1.81
16     147570.0          1160.0350       1.0108
2.15
27    8515767.0            25.2841       1.0046
2.70
149    923768.0           236.5759       1.0241
2.74
156    881912.0           267.4018       1.0191
2.96
93    1904569.0           144.6529       1.0064
3.45
221   9372610.0            36.0935       1.0038
4.24
92    3287590.0           431.0675       1.0068
17.77
41    9706961.0           146.8933       1.0000
17.88
```

```
# we can do ordering by multiple columns too
df[df['Rank'] < 10].sort_values(by = ['Rank','Country'] , ascending =
False)
# due to order of importance , the table does'nt changed much
```

```
     Rank CCA3        Country           Capital      Continent  \
171     9  RUS         Russia            Moscow         Europe
16      8  BGD     Bangladesh             Dhaka           Asia
27      7  BRA         Brazil          Brasilia  South America
149     6  NGA        Nigeria             Abuja         Africa
156     5  PAK       Pakistan         Islamabad           Asia
93      4  IDN      Indonesia           Jakarta           Asia
221     3  USA  United States  Washington, D.C.  North America
92      2  IND          India         New Delhi           Asia
41      1  CHN          China           Beijing           Asia

     2022 Population  2020 Population  2015 Population  2010
Population  \
171     1.447133e+08     1.456173e+08     1.446684e+08
1.432426e+08
16      1.711864e+08     1.674210e+08     1.578300e+08
1.483911e+08
27      2.153135e+08     2.131963e+08     2.051882e+08
1.963535e+08
149     2.185412e+08     2.083274e+08     1.839958e+08
1.609529e+08
```

```
156     2.358249e+08     2.271967e+08     2.109693e+08
1.944545e+08
93      2.755013e+08     2.718580e+08     2.590920e+08
2.440162e+08
221     3.382899e+08     3.359420e+08     3.246078e+08
3.111828e+08
92      1.417173e+09     1.396387e+09     1.322867e+09
1.240614e+09
41      1.425887e+09     1.424930e+09     1.393715e+09
1.348191e+09

     2000 Population  1990 Population  1980 Population  1970
Population  \
171     1.468448e+08     1.480057e+08      138257420.0
130093010.0
16      1.291933e+08     1.071477e+08       83929765.0
67541860.0
27      1.758737e+08     1.507064e+08      122288383.0
96369875.0
149     1.228520e+08     9.521426e+07       72951439.0
55569264.0
156     1.543699e+08     1.154141e+08       80624057.0
59290872.0
93      2.140724e+08     1.821599e+08      148177096.0
115228394.0
221     2.823986e+08     2.480837e+08      223140018.0
200328340.0
92      1.059634e+09              NaN              NaN
557501301.0
41      1.264099e+09     1.153704e+09      982372466.0
822534450.0

     Area (km²)  Density (per km²)  Growth Rate  World Population
Percentage
171  17098242.0             8.4636       0.9973
1.81
16     147570.0          1160.0350       1.0108
2.15
27    8515767.0            25.2841       1.0046
2.70
149    923768.0           236.5759       1.0241
2.74
156    881912.0           267.4018       1.0191
2.96
93    1904569.0           144.6529       1.0064
3.45
221   9372610.0            36.0935       1.0038
4.24
92    3287590.0           431.0675       1.0068
```

```
17.77
41    9706961.0           146.8933       1.0000
17.88
```

```python
# for writing in different ways
df[df['Rank'] < 10].sort_values(by = ['Country','Rank'] , ascending =
False) # first sorting country
# then rank
```

```
     Rank CCA3         Country            Capital      Continent  \
221     3  USA  United States  Washington, D.C.  North America
171     9  RUS         Russia             Moscow         Europe
156     5  PAK       Pakistan          Islamabad           Asia
149     6  NGA        Nigeria              Abuja         Africa
93      4  IDN      Indonesia            Jakarta           Asia
92      2  IND          India          New Delhi           Asia
41      1  CHN          China            Beijing           Asia
27      7  BRA         Brazil           Brasilia  South America
16      8  BGD     Bangladesh              Dhaka           Asia

     2022 Population  2020 Population  2015 Population  2010
Population  \
221     3.382899e+08     3.359420e+08     3.246078e+08
3.111828e+08
171     1.447133e+08     1.456173e+08     1.446684e+08
1.432426e+08
156     2.358249e+08     2.271967e+08     2.109693e+08
1.944545e+08
149     2.185412e+08     2.083274e+08     1.839958e+08
1.609529e+08
93      2.755013e+08     2.718580e+08     2.590920e+08
2.440162e+08
92      1.417173e+09     1.396387e+09     1.322867e+09
1.240614e+09
41      1.425887e+09     1.424930e+09     1.393715e+09
1.348191e+09
27      2.153135e+08     2.131963e+08     2.051882e+08
1.963535e+08
16      1.711864e+08     1.674210e+08     1.578300e+08
1.483911e+08

     2000 Population  1990 Population  1980 Population  1970
Population  \
221     2.823986e+08     2.480837e+08      223140018.0
200328340.0
171     1.468448e+08     1.480057e+08      138257420.0
130093010.0
156     1.543699e+08     1.154141e+08       80624057.0
59290872.0
149     1.228520e+08     9.521426e+07       72951439.0
```

```
55569264.0
93      2.140724e+08     1.821599e+08      148177096.0
115228394.0
92      1.059634e+09              NaN              NaN
557501301.0
41      1.264099e+09     1.153704e+09      982372466.0
822534450.0
27      1.758737e+08     1.507064e+08      122288383.0
96369875.0
16      1.291933e+08     1.071477e+08       83929765.0
67541860.0

     Area (km²)  Density (per km²)  Growth Rate  World Population
Percentage
221   9372610.0            36.0935       1.0038
4.24
171  17098242.0             8.4636       0.9973
1.81
156    881912.0           267.4018       1.0191
2.96
149    923768.0           236.5759       1.0241
2.74
93    1904569.0           144.6529       1.0064
3.45
92    3287590.0           431.0675       1.0068
17.77
41    9706961.0           146.8933       1.0000
17.88
27    8515767.0            25.2841       1.0046
2.70
16     147570.0          1160.0350       1.0108
2.15
```

Indexing

```
# index : obeject that stores access label for all pandas object / No. 
or label for each row

df = pd.read_csv(r"D:\gami\world_population (1).csv")
df

     Rank CCA3            Country           Capital Continent  \
0      36  AFG        Afghanistan             Kabul      Asia
1     138  ALB            Albania            Tirana    Europe
2      34  DZA            Algeria           Algiers    Africa
3     213  ASM     American Samoa         Pago Pago   Oceania
4     203  AND            Andorra  Andorra la Vella    Europe
..    ...  ...                ...               ...       ...
```

```
229   226  WLF  Wallis and Futuna           Mata-Utu   Oceania
230   172  ESH     Western Sahara          El Aaiún    Africa
231    46  YEM              Yemen             Sanaa      Asia
232    63  ZMB             Zambia            Lusaka    Africa
233    74  ZWE           Zimbabwe            Harare    Africa

     2022 Population  2020 Population  2015 Population  2010
Population  \
0         41128771.0       38972230.0       33753499.0
28189672.0
1          2842321.0        2866849.0        2882481.0
2913399.0
2         44903225.0       43451666.0       39543154.0
35856344.0
3            44273.0          46189.0          51368.0
54849.0
4            79824.0          77700.0          71746.0
71519.0
..               ...              ...              ...              ..
.
229          11572.0          11655.0          12182.0
13142.0
230         575986.0         556048.0         491824.0
413296.0
231       33696614.0       32284046.0       28516545.0
24743946.0
232       20017675.0       18927715.0              NaN
13792086.0
233       16320537.0       15669666.0       14154937.0
12839771.0

     2000 Population  1990 Population  1980 Population  1970
Population  \
0         19542982.0       10694796.0       12486631.0
10752971.0
1          3182021.0        3295066.0        2941651.0
2324731.0
2         30774621.0       25518074.0       18739378.0
13795915.0
3            58230.0          47818.0          32886.0
27075.0
4            66097.0          53569.0          35611.0
19860.0
..               ...              ...              ...              ..
.
229          14723.0          13454.0          11315.0
9377.0
230         270375.0         178529.0         116775.0
76371.0
```

```
231       18628700.0       13375121.0        9204938.0
6843607.0
232        9891136.0        7686401.0        5720438.0
4281671.0
233       11834676.0       10113893.0        7049926.0
5202918.0

     Area (km²)  Density (per km²)  Growth Rate  World Population
Percentage
0      652230.0            63.0587       1.0257
0.52
1       28748.0            98.8702       0.9957
0.04
2     2381741.0            18.8531       1.0164
0.56
3         199.0           222.4774       0.9831
0.00
4         468.0           170.5641       1.0100
0.00
..          ...                ...          ...
...
229       142.0            81.4930       0.9953
0.00
230    266000.0             2.1654       1.0184
0.01
231    527968.0            63.8232       1.0217
0.42
232    752612.0            26.5976       1.0280
0.25
233    390757.0            41.7665       1.0204
0.20

[234 rows x 17 columns]
```

```
df = pd.read_csv(r"D:\gami\world_population (1).csv" , index_col =
'Country')
df # changing index to country
```

```
                   Rank CCA3           Capital Continent  2022
Population  \
Country

Afghanistan          36  AFG             Kabul      Asia
41128771.0
Albania             138  ALB            Tirana    Europe
2842321.0
Algeria              34  DZA           Algiers    Africa
44903225.0
American Samoa      213  ASM         Pago Pago   Oceania
44273.0
```

```
Andorra                203  AND  Andorra la Vella    Europe
79824.0
...                    ...  ...               ...       ...
...
Wallis and Futuna      226  WLF          Mata-Utu   Oceania
11572.0
Western Sahara         172  ESH          El Aaiún    Africa
575986.0
Yemen                   46  YEM             Sanaa      Asia
33696614.0
Zambia                  63  ZMB            Lusaka    Africa
20017675.0
Zimbabwe                74  ZWE            Harare    Africa
16320537.0

                   2020 Population  2015 Population  2010
Population  \
Country

Afghanistan             38972230.0       33753499.0       28189672.0

Albania                  2866849.0        2882481.0        2913399.0

Algeria                 43451666.0       39543154.0       35856344.0

American Samoa             46189.0          51368.0          54849.0

Andorra                    77700.0          71746.0          71519.0

...                            ...              ...              ...

Wallis and Futuna          11655.0          12182.0          13142.0

Western Sahara            556048.0         491824.0         413296.0

Yemen                   32284046.0       28516545.0       24743946.0

Zambia                  18927715.0              NaN       13792086.0

Zimbabwe                15669666.0       14154937.0       12839771.0


                   2000 Population  1990 Population  1980
Population  \
Country

Afghanistan             19542982.0       10694796.0       12486631.0

Albania                  3182021.0        3295066.0        2941651.0

Algeria                 30774621.0       25518074.0       18739378.0
```

```
American Samoa              58230.0            47818.0            32886.0
Andorra                     66097.0            53569.0            35611.0
...                             ...                ...                ...
Wallis and Futuna           14723.0            13454.0            11315.0
Western Sahara             270375.0           178529.0           116775.0
Yemen                    18628700.0         13375121.0          9204938.0
Zambia                    9891136.0          7686401.0          5720438.0
Zimbabwe                 11834676.0         10113893.0          7049926.0

                   1970 Population  Area (km²)  Density (per km²)  \
Country
Afghanistan             10752971.0    652230.0            63.0587
Albania                  2324731.0     28748.0            98.8702
Algeria                 13795915.0   2381741.0            18.8531
American Samoa             27075.0       199.0           222.4774
Andorra                    19860.0       468.0           170.5641
...                            ...         ...                ...
Wallis and Futuna           9377.0       142.0            81.4930
Western Sahara             76371.0    266000.0             2.1654
Yemen                    6843607.0    527968.0            63.8232
Zambia                   4281671.0    752612.0            26.5976
Zimbabwe                 5202918.0    390757.0            41.7665

                   Growth Rate  World Population Percentage
Country
Afghanistan             1.0257                         0.52
Albania                 0.9957                         0.04
Algeria                 1.0164                         0.56
American Samoa          0.9831                         0.00
Andorra                 1.0100                         0.00
...                        ...                          ...
Wallis and Futuna       0.9953                         0.00
Western Sahara          1.0184                         0.01
Yemen                   1.0217                         0.42
Zambia                  1.0280                         0.25
Zimbabwe                1.0204                         0.20

[234 rows x 16 columns]
```

```
df.reset_index(inplace = True) # Index got reseted and it happens in the same previous table due to
```

```python
#Inplace : performing an operation that directly modifies the data of an object rather than creating a new object.

# If you don't use drop = True , then it will create a new col index and level_0
# So , always first use drop or use df.drop(columns = ['level_0'] , inplace = True)
# df.drop(columns = ['index'] , inplace = True) , for removing extra col

df
```

```
                Country  Rank CCA3           Capital Continent  \
0           Afghanistan    36  AFG             Kabul      Asia
1               Albania   138  ALB            Tirana    Europe
2               Algeria    34  DZA           Algiers    Africa
3        American Samoa   213  ASM         Pago Pago   Oceania
4               Andorra   203  AND  Andorra la Vella    Europe
..                  ...   ...  ...               ...       ...
229   Wallis and Futuna   226  WLF          Mata-Utu   Oceania
230      Western Sahara   172  ESH          El Aaiún    Africa
231               Yemen    46  YEM             Sanaa      Asia
232              Zambia    63  ZMB            Lusaka    Africa
233            Zimbabwe    74  ZWE            Harare    Africa

     2022 Population  2020 Population  2015 Population  2010 
Population  \
0         41128771.0       38972230.0       33753499.0       
28189672.0
1          2842321.0        2866849.0        2882481.0       
2913399.0
2         44903225.0       43451666.0       39543154.0       
35856344.0
3            44273.0          46189.0          51368.0       
54849.0
4            79824.0          77700.0          71746.0       
71519.0
..               ...              ...              ...             ..
.
229          11572.0          11655.0          12182.0       
13142.0
230         575986.0         556048.0         491824.0       
413296.0
231       33696614.0       32284046.0       28516545.0       
24743946.0
232       20017675.0       18927715.0              NaN       
13792086.0
233       16320537.0       15669666.0       14154937.0       
12839771.0
```

```
     2000 Population  1990 Population  1980 Population  1970 
Population  \
0         19542982.0       10694796.0       12486631.0         
10752971.0
1          3182021.0        3295066.0        2941651.0          
2324731.0
2         30774621.0       25518074.0       18739378.0         
13795915.0
3            58230.0          47818.0          32886.0            
27075.0
4            66097.0          53569.0          35611.0            
19860.0
..               ...              ...              ...                
.
229          14723.0          13454.0          11315.0             
9377.0
230         270375.0         178529.0         116775.0            
76371.0
231       18628700.0       13375121.0        9204938.0          
6843607.0
232        9891136.0        7686401.0        5720438.0          
4281671.0
233       11834676.0       10113893.0        7049926.0          
5202918.0

     Area (km²)  Density (per km²)  Growth Rate  World Population 
Percentage
0      652230.0            63.0587       1.0257                       
0.52
1       28748.0            98.8702       0.9957                       
0.04
2     2381741.0            18.8531       1.0164                       
0.56
3         199.0           222.4774       0.9831                       
0.00
4         468.0           170.5641       1.0100                       
0.00
..          ...                ...          ...                        
...
229       142.0            81.4930       0.9953                       
0.00
230    266000.0             2.1654       1.0184                       
0.01
231    527968.0            63.8232       1.0217                       
0.42
232    752612.0            26.5976       1.0280                       
0.25
233    390757.0            41.7665       1.0204                       
0.20
```

```
[234 rows x 17 columns]

# in different way ,
df.set_index('Country') # It will momentrily change index

                   Rank CCA3           Capital Continent  2022
Population  \
Country

Afghanistan          36  AFG             Kabul      Asia
41128771.0
Albania             138  ALB            Tirana    Europe
2842321.0
Algeria              34  DZA           Algiers    Africa
44903225.0
American Samoa      213  ASM         Pago Pago   Oceania
44273.0
Andorra             203  AND  Andorra la Vella    Europe
79824.0
...                 ...  ...               ...       ...
...
Wallis and Futuna   226  WLF          Mata-Utu   Oceania
11572.0
Western Sahara      172  ESH          El Aaiún    Africa
575986.0
Yemen                46  YEM             Sanaa      Asia
33696614.0
Zambia               63  ZMB            Lusaka    Africa
20017675.0
Zimbabwe             74  ZWE            Harare    Africa
16320537.0

                   2020 Population  2015 Population  2010
Population  \
Country

Afghanistan             38972230.0       33753499.0       28189672.0

Albania                  2866849.0        2882481.0        2913399.0

Algeria                 43451666.0       39543154.0       35856344.0

American Samoa             46189.0          51368.0          54849.0

Andorra                    77700.0          71746.0          71519.0

...                            ...              ...              ...

Wallis and Futuna          11655.0          12182.0          13142.0
```

```
Western Sahara          556048.0       491824.0       413296.0
Yemen                 32284046.0     28516545.0     24743946.0
Zambia                18927715.0            NaN     13792086.0
Zimbabwe              15669666.0     14154937.0     12839771.0

                   2000 Population  1990 Population  1980
Population  \
Country
Afghanistan           19542982.0     10694796.0     12486631.0
Albania                3182021.0      3295066.0      2941651.0
Algeria               30774621.0     25518074.0     18739378.0
American Samoa           58230.0        47818.0        32886.0
Andorra                  66097.0        53569.0        35611.0
...                          ...            ...            ...
Wallis and Futuna        14723.0        13454.0        11315.0
Western Sahara          270375.0       178529.0       116775.0
Yemen                 18628700.0     13375121.0      9204938.0
Zambia                 9891136.0      7686401.0      5720438.0
Zimbabwe              11834676.0     10113893.0      7049926.0

                   1970 Population  Area (km²)  Density (per km²)  \
Country
Afghanistan           10752971.0    652230.0            63.0587
Albania                2324731.0     28748.0            98.8702
Algeria               13795915.0   2381741.0            18.8531
American Samoa           27075.0       199.0           222.4774
Andorra                  19860.0       468.0           170.5641
...                          ...         ...                ...
Wallis and Futuna         9377.0       142.0            81.4930
Western Sahara           76371.0    266000.0             2.1654
Yemen                  6843607.0    527968.0            63.8232
Zambia                 4281671.0    752612.0            26.5976
Zimbabwe               5202918.0    390757.0            41.7665

                   Growth Rate  World Population Percentage
```

```
Country
Afghanistan             1.0257                        0.52
Albania                 0.9957                        0.04
Algeria                 1.0164                        0.56
American Samoa          0.9831                        0.00
Andorra                 1.0100                        0.00
...                        ...                         ...
Wallis and Futuna       0.9953                        0.00
Western Sahara          1.0184                        0.01
Yemen                   1.0217                        0.42
Zambia                  1.0280                        0.25
Zimbabwe                1.0204                        0.20

[234 rows x 16 columns]
```

```
df.set_index('Country' , inplace = True)

df # now , using inplace works
```

```
                   Rank CCA3           Capital Continent  2022
Population  \
Country

Afghanistan          36  AFG             Kabul      Asia
41128771.0
Albania             138  ALB            Tirana    Europe
2842321.0
Algeria              34  DZA           Algiers    Africa
44903225.0
American Samoa      213  ASM         Pago Pago   Oceania
44273.0
Andorra             203  AND  Andorra la Vella    Europe
79824.0
...                 ...  ...               ...       ...
...
Wallis and Futuna   226  WLF          Mata-Utu   Oceania
11572.0
Western Sahara      172  ESH          El Aaiún    Africa
575986.0
Yemen                46  YEM             Sanaa      Asia
33696614.0
Zambia               63  ZMB            Lusaka    Africa
20017675.0
Zimbabwe             74  ZWE            Harare    Africa
16320537.0

                   2020 Population  2015 Population  2010
Population  \
Country
```

```
Afghanistan                38972230.0         33753499.0         28189672.0
Albania                     2866849.0          2882481.0          2913399.0
Algeria                    43451666.0         39543154.0         35856344.0
American Samoa                46189.0            51368.0            54849.0
Andorra                       77700.0            71746.0            71519.0
...                               ...                ...                ...
Wallis and Futuna             11655.0            12182.0            13142.0
Western Sahara               556048.0           491824.0           413296.0
Yemen                      32284046.0         28516545.0         24743946.0
Zambia                     18927715.0                NaN         13792086.0
Zimbabwe                   15669666.0         14154937.0         12839771.0

                   2000 Population  1990 Population  1980
Population  \
Country
Afghanistan                19542982.0         10694796.0         12486631.0
Albania                     3182021.0          3295066.0          2941651.0
Algeria                    30774621.0         25518074.0         18739378.0
American Samoa                58230.0            47818.0            32886.0
Andorra                       66097.0            53569.0            35611.0
...                               ...                ...                ...
Wallis and Futuna             14723.0            13454.0            11315.0
Western Sahara               270375.0           178529.0           116775.0
Yemen                      18628700.0         13375121.0          9204938.0
Zambia                      9891136.0          7686401.0          5720438.0
Zimbabwe                   11834676.0         10113893.0          7049926.0

                   1970 Population  Area (km²)  Density (per km²)  \
Country
```

```
Afghanistan             10752971.0    652230.0             63.0587
Albania                  2324731.0     28748.0             98.8702
Algeria                 13795915.0   2381741.0             18.8531
American Samoa             27075.0       199.0            222.4774
Andorra                    19860.0       468.0            170.5641
...                            ...         ...                 ...
Wallis and Futuna           9377.0       142.0             81.4930
Western Sahara             76371.0    266000.0              2.1654
Yemen                    6843607.0    527968.0             63.8232
Zambia                   4281671.0    752612.0             26.5976
Zimbabwe                 5202918.0    390757.0             41.7665

                   Growth Rate  World Population Percentage
Country
Afghanistan             1.0257                         0.52
Albania                 0.9957                         0.04
Algeria                 1.0164                         0.56
American Samoa          0.9831                         0.00
Andorra                 1.0100                         0.00
...                        ...                          ...
Wallis and Futuna       0.9953                         0.00
Western Sahara          1.0184                         0.01
Yemen                   1.0217                         0.42
Zambia                  1.0280                         0.25
Zimbabwe                1.0204                         0.20

[234 rows x 16 columns]
```

```python
# Searching based of the index : loc and iloc (location and integer location)
df.loc['Albania']
```

```
Rank                                       138
CCA3                                       ALB
Capital                                 Tirana
Continent                               Europe
2022 Population                      2842321.0
2020 Population                      2866849.0
2015 Population                      2882481.0
2010 Population                      2913399.0
2000 Population                      3182021.0
1990 Population                      3295066.0
1980 Population                      2941651.0
1970 Population                      2324731.0
Area (km²)                             28748.0
Density (per km²)                      98.8702
Growth Rate                             0.9957
World Population Percentage               0.04
Name: Albania, dtype: object
```

```
# Searching based of the index : loc and iloc (location and integer location)
df.iloc[1]
```

```
Rank                                          138
CCA3                                          ALB
Capital                                    Tirana
Continent                                  Europe
2022 Population                         2842321.0
2020 Population                         2866849.0
2015 Population                         2882481.0
2010 Population                         2913399.0
2000 Population                         3182021.0
1990 Population                         3295066.0
1980 Population                         2941651.0
1970 Population                         2324731.0
Area (km²)                                28748.0
Density (per km²)                         98.8702
Growth Rate                                0.9957
World Population Percentage                  0.04
Name: Albania, dtype: object
```

```
# Multi indexing
df.reset_index(inplace = True)
df
```

```
               Country  Rank CCA3           Capital Continent  \
0          Afghanistan    36  AFG             Kabul      Asia
1              Albania   138  ALB            Tirana    Europe
2              Algeria    34  DZA           Algiers    Africa
3       American Samoa   213  ASM         Pago Pago   Oceania
4              Andorra   203  AND  Andorra la Vella    Europe
..                 ...   ...  ...               ...       ...
229  Wallis and Futuna   226  WLF          Mata-Utu   Oceania
230     Western Sahara   172  ESH          El Aaiún    Africa
231              Yemen    46  YEM             Sanaa      Asia
232             Zambia    63  ZMB            Lusaka    Africa
233           Zimbabwe    74  ZWE            Harare    Africa

     2022 Population  2020 Population  2015 Population  2010 
Population  \
0         41128771.0       38972230.0       33753499.0
28189672.0
1          2842321.0        2866849.0        2882481.0
2913399.0
2         44903225.0       43451666.0       39543154.0
35856344.0
3            44273.0          46189.0          51368.0
54849.0
4            79824.0          77700.0          71746.0
```

```
71519.0
..               ...              ...              ...              ..
.
229          11572.0          11655.0          12182.0
13142.0
230         575986.0         556048.0         491824.0
413296.0
231       33696614.0       32284046.0       28516545.0
24743946.0
232       20017675.0       18927715.0              NaN
13792086.0
233       16320537.0       15669666.0       14154937.0
12839771.0

     2000 Population  1990 Population  1980 Population  1970
Population  \
0         19542982.0       10694796.0       12486631.0
10752971.0
1          3182021.0        3295066.0        2941651.0
2324731.0
2         30774621.0       25518074.0       18739378.0
13795915.0
3            58230.0          47818.0          32886.0
27075.0
4            66097.0          53569.0          35611.0
19860.0
..               ...              ...              ...              ..
.
229          14723.0          13454.0          11315.0
9377.0
230         270375.0         178529.0         116775.0
76371.0
231       18628700.0       13375121.0        9204938.0
6843607.0
232        9891136.0        7686401.0        5720438.0
4281671.0
233       11834676.0       10113893.0        7049926.0
5202918.0

     Area (km²)  Density (per km²)  Growth Rate  World Population
Percentage
0      652230.0            63.0587       1.0257
0.52
1       28748.0            98.8702       0.9957
0.04
2     2381741.0            18.8531       1.0164
0.56
3         199.0           222.4774       0.9831
0.00
```

```
4        468.0          170.5641      1.0100
0.00
..          ...                ...          ...
...
229      142.0           81.4930      0.9953
0.00
230   266000.0            2.1654      1.0184
0.01
231   527968.0           63.8232      1.0217
0.42
232   752612.0           26.5976      1.0280
0.25
233   390757.0           41.7665      1.0204
0.20

[234 rows x 17 columns]
```

```
df.set_index(['Continent' , 'Country'] , inplace = True)

df
```

```
                             Rank CCA3           Capital  2022
Population  \
Continent Country

Asia      Afghanistan          36  AFG             Kabul
41128771.0
Europe    Albania             138  ALB            Tirana
2842321.0
Africa    Algeria              34  DZA           Algiers
44903225.0
Oceania   American Samoa      213  ASM         Pago Pago
44273.0
Europe    Andorra             203  AND  Andorra la Vella
79824.0
...                           ...  ...               ...
...
Oceania   Wallis and Futuna   226  WLF          Mata-Utu
11572.0
Africa    Western Sahara      172  ESH          El Aaiún
575986.0
Asia      Yemen                46  YEM             Sanaa
33696614.0
Africa    Zambia               63  ZMB            Lusaka
20017675.0
          Zimbabwe             74  ZWE            Harare
16320537.0

                             2020 Population  2015 Population  \
Continent Country
```

```
Asia      Afghanistan                38972230.0       33753499.0
Europe    Albania                     2866849.0        2882481.0
Africa    Algeria                    43451666.0       39543154.0
Oceania   American Samoa                46189.0          51368.0
Europe    Andorra                       77700.0          71746.0
...                                         ...              ...
Oceania   Wallis and Futuna             11655.0          12182.0
Africa    Western Sahara               556048.0         491824.0
Asia      Yemen                      32284046.0       28516545.0
Africa    Zambia                     18927715.0              NaN
          Zimbabwe                   15669666.0       14154937.0

                             2010 Population  2000 Population  \
Continent Country
Asia      Afghanistan                28189672.0       19542982.0
Europe    Albania                     2913399.0        3182021.0
Africa    Algeria                    35856344.0       30774621.0
Oceania   American Samoa                54849.0          58230.0
Europe    Andorra                       71519.0          66097.0
...                                         ...              ...
Oceania   Wallis and Futuna             13142.0          14723.0
Africa    Western Sahara               413296.0         270375.0
Asia      Yemen                      24743946.0       18628700.0
Africa    Zambia                     13792086.0        9891136.0
          Zimbabwe                   12839771.0       11834676.0

                             1990 Population  1980 Population  \
Continent Country
Asia      Afghanistan                10694796.0       12486631.0
Europe    Albania                     3295066.0        2941651.0
Africa    Algeria                    25518074.0       18739378.0
Oceania   American Samoa                47818.0          32886.0
Europe    Andorra                       53569.0          35611.0
...                                         ...              ...
Oceania   Wallis and Futuna             13454.0          11315.0
Africa    Western Sahara               178529.0         116775.0
Asia      Yemen                      13375121.0        9204938.0
Africa    Zambia                      7686401.0        5720438.0
          Zimbabwe                   10113893.0        7049926.0

                             1970 Population  Area (km²)  Density (per
km²)  \
Continent Country

Asia      Afghanistan                10752971.0    652230.0
63.0587
Europe    Albania                     2324731.0     28748.0
98.8702
Africa    Algeria                    13795915.0   2381741.0
18.8531
```

```
Oceania   American Samoa                27075.0       199.0
222.4774
Europe    Andorra                       19860.0       468.0
170.5641
...                                         ...         ...
...
Oceania   Wallis and Futuna              9377.0       142.0
81.4930
Africa    Western Sahara                76371.0    266000.0
2.1654
Asia      Yemen                       6843607.0    527968.0
63.8232
Africa    Zambia                      4281671.0    752612.0
26.5976
          Zimbabwe                    5202918.0    390757.0
41.7665

                             Growth Rate  World Population Percentage

Continent Country

Asia      Afghanistan             1.0257                         0.52

Europe    Albania                 0.9957                         0.04

Africa    Algeria                 1.0164                         0.56

Oceania   American Samoa          0.9831                         0.00

Europe    Andorra                 1.0100                         0.00

...                                  ...                          ...

Oceania   Wallis and Futuna       0.9953                         0.00

Africa    Western Sahara          1.0184                         0.01

Asia      Yemen                   1.0217                         0.42

Africa    Zambia                  1.0280                         0.25

          Zimbabwe                1.0204                         0.20


[234 rows x 15 columns]
```

```
df.sort_index(ascending = True)
```

```
                            Rank CCA3      Capital  2022 Population  \
Continent     Country
Africa        Algeria         34  DZA      Algiers       44903225.0
```

```
                 Angola           42  AGO       Luanda       35588987.0
                 Benin            77  BEN   Porto-Novo       13352864.0
                 Botswana        144  BWA     Gaborone        2630296.0
                 Burkina Faso     58  BFA  Ouagadougou       22673762.0
...                              ...  ...          ...              ...
South America Paraguay          109  PRY     Asunción        6780744.0
                 Peru             44  PER         Lima       34049588.0
                 Suriname        170  SUR   Paramaribo         618040.0
                 Uruguay         133  URY   Montevideo        3422794.0
                 Venezuela        51  VEN      Caracas       28301696.0

                            2020 Population  2015 Population  2010
Population  \
Continent     Country

Africa        Algeria              43451666.0       39543154.0
35856344.0
              Angola               33428485.0       28127721.0
23364185.0
              Benin                12643123.0       10932783.0
9445710.0
              Botswana              2546402.0        2305171.0
2091664.0
              Burkina Faso         21522626.0       18718019.0
16116845.0
...                                       ...              ...
...
South America Paraguay              6618695.0        6177950.0
5768613.0
              Peru                 33304756.0       30711863.0
29229572.0
              Suriname               607065.0         575475.0
546080.0
              Uruguay               3429086.0        3402818.0
3352651.0
              Venezuela            28490453.0       30529716.0
28715022.0

                            2000 Population  1990 Population  1980
Population  \
Continent     Country

Africa        Algeria              30774621.0       25518074.0
18739378.0
              Angola               16394062.0       11828638.0
8330047.0
              Benin                 6998023.0        5133419.0
3833939.0
              Botswana              1726985.0        1341474.0
938578.0
```

```
               Burkina Faso       11882888.0        9131361.0
6932967.0
...                                      ...              ...
...
South America Paraguay             5123819.0        4059195.0
3078912.0
               Peru               26654439.0       22109099.0
17492406.0
               Suriname             478998.0         412756.0
375112.0
               Uruguay             3292224.0        3117012.0
2953750.0
               Venezuela                 NaN       19750579.0
15210443.0

                            1970 Population  Area (km²)  Density (per
km²)  \
Continent     Country

Africa        Algeria             13795915.0   2381741.0
18.8531
              Angola               6029700.0   1246700.0
28.5466
              Benin                3023443.0    112622.0
118.5635
              Botswana              592244.0    582000.0
4.5194
              Burkina Faso         5611666.0    272967.0
83.0641
...                                      ...         ...
...
South America Paraguay             2408787.0    406752.0
16.6705
              Peru                13562371.0   1285216.0
26.4933
              Suriname              379918.0    163820.0
3.7727
              Uruguay              2790265.0    181034.0
18.9069
              Venezuela           11355475.0         NaN
30.8820

                            Growth Rate  World Population Percentage
Continent     Country
Africa        Algeria            1.0164                         0.56
              Angola             1.0315                         0.45
              Benin              1.0274                         0.17
              Botswana           1.0162                         0.03
              Burkina Faso       1.0259                         0.28
...                                 ...                          ...
```

```
South America Paraguay          1.0115                         0.09
              Peru              1.0099                         0.43
              Suriname          1.0082                         0.01
              Uruguay           0.9990                         0.04
              Venezuela         1.0036                         0.35

[234 rows x 15 columns]
```

```python
# df.loc['Angola'] # not gonna work properly
                 # It's searching in first index / string which is Continent
df.loc['Africa' , 'Angola']
```

```
Rank                                   42
CCA3                                  AGO
Capital                            Luanda
2022 Population                35588987.0
2020 Population                33428485.0
2015 Population                28127721.0
2010 Population                23364185.0
2000 Population                16394062.0
1990 Population                11828638.0
1980 Population                 8330047.0
1970 Population                 6029700.0
Area (km²)                      1246700.0
Density (per km²)                 28.5466
Growth Rate                        1.0315
World Population Percentage          0.45
Name: (Africa, Angola), dtype: object
```

```python
df.iloc[1] # it does'nt go on the basis of multi indexing , it go on as base of original indexing
```

```
Rank                                  138
CCA3                                  ALB
Capital                            Tirana
2022 Population                 2842321.0
2020 Population                 2866849.0
2015 Population                 2882481.0
2010 Population                 2913399.0
2000 Population                 3182021.0
1990 Population                 3295066.0
1980 Population                 2941651.0
1970 Population                 2324731.0
Area (km²)                        28748.0
Density (per km²)                 98.8702
Growth Rate                        0.9957
World Population Percentage          0.04
Name: (Europe, Albania), dtype: object
```

Group by and Aggregating

```
# group by : groups values in a column , use to perform agg fun

df = pd.read_csv(r"D:\gami\Flavors (1).csv")
df

                   Flavor Base Flavor Liked  Flavor Rating  Texture
Rating  \
0     Mint Chocolate Chip     Vanilla   Yes           10.0
8.0
1               Chocolate   Chocolate   Yes            8.8
7.6
2                 Vanilla     Vanilla    No            4.7
5.0
3            Cookie Dough     Vanilla   Yes            6.9
6.5
4              Rocky Road   Chocolate   Yes            8.2
7.0
5               Pistachio     Vanilla    No            2.3
3.4
6             Cake Batter     Vanilla   Yes            6.5
6.0
7              Neapolitan     Vanilla    No            3.8
5.0
8  Chocolte Fudge Brownie   Chocolate   Yes            8.2
7.1

   Total Rating
0          18.0
1          16.6
2           9.7
3          13.4
4          15.2
5           5.7
6          12.5
7           8.8
8          15.3

group_by_frame = df.groupby('Base Flavor') # grouping based on base
flavor

group_by_frame.mean(numeric_only = True) # use numeric only otherwise
it can't find mean of liked, etc

             Flavor Rating  Texture Rating  Total Rating
Base Flavor
Chocolate              8.4        7.233333         15.70
Vanilla                5.7        5.650000         11.35
```

```
# popular aggergate functions
print(group_by_frame.count())
print('\n')
print(group_by_frame.min())  # c in chocolate is very min value in
string in chocolate
print('\n')
print(group_by_frame.max())  # R in rocky Road is very max value of
string in chocolate
print('\n')
print(group_by_frame.sum(numeric_only = True))
print('\n')

             Flavor  Liked  Flavor Rating  Texture Rating  Total
Rating
Base Flavor

Chocolate         3      3              3               3
3
Vanilla           6      6              6               6
6


                  Flavor Liked  Flavor Rating  Texture Rating  Total
Rating
Base Flavor

Chocolate      Chocolate   Yes            8.2             7.0
15.2
Vanilla      Cake Batter    No            2.3             3.4
5.7


                 Flavor Liked  Flavor Rating  Texture Rating  Total
Rating
Base Flavor

Chocolate    Rocky Road   Yes            8.8             7.6
16.6
Vanilla         Vanilla   Yes           10.0             8.0
18.0


             Flavor Rating  Texture Rating  Total Rating
Base Flavor
Chocolate             25.2            21.7          47.1
Vanilla               34.2            33.9          68.1



# agg : It is used to groupby a column , and can have multiple agg
values
```

```
group_by_frame.agg({'Flavor Rating' : ['mean' , 'max' ,'count' ,
'sum']})

             Flavor Rating
                      mean   max count   sum
Base Flavor
Chocolate              8.4   8.8     3  25.2
Vanilla                5.7  10.0     6  34.2

# It could be multiple rated too
group_by_frame.agg({'Flavor Rating' : ['mean' , 'max' ,'count' ,
'sum'] ,
                   'Texture Rating' : ['mean' , 'max' ,'count' ,
'sum']})

             Flavor Rating                   Texture Rating

                      mean   max count   sum           mean  max count
sum
Base Flavor

Chocolate              8.4   8.8     3  25.2       7.233333  7.6     3
21.7
Vanilla                5.7  10.0     6  34.2       5.650000  8.0     6
33.9

# we can also group on multiple column
df

                    Flavor Base Flavor Liked  Flavor Rating  Texture
Rating  \
0     Mint Chocolate Chip     Vanilla   Yes           10.0
8.0
1               Chocolate   Chocolate   Yes            8.8
7.6
2                 Vanilla     Vanilla    No            4.7
5.0
3           Cookie Dough     Vanilla   Yes            6.9
6.5
4             Rocky Road   Chocolate   Yes            8.2
7.0
5              Pistachio     Vanilla    No            2.3
3.4
6            Cake Batter     Vanilla   Yes            6.5
6.0
7            Neapolitan     Vanilla    No            3.8
5.0
8  Chocolte Fudge Brownie   Chocolate   Yes            8.2
7.1

   Total Rating
```

```
0          18.0
1          16.6
2           9.7
3          13.4
4          15.2
5           5.7
6          12.5
7           8.8
8          15.3
```

```
new_groupby = df.groupby(['Base Flavor','Liked'])
```

```
new_groupby.mean(numeric_only = True)
```

```
                   Flavor Rating  Texture Rating  Total Rating
Base Flavor Liked
Chocolate   Yes              8.4        7.233333     15.700000
Vanilla     No               3.6        4.466667      8.066667
            Yes              7.8        6.833333     14.633333
```

```
# shortcut function to get all these things really quickly
df.groupby('Base Flavor').describe()
```

```
            Flavor
Rating                                                \
                    count mean       std  min    25%  50%  75%   max

Base Flavor

Chocolate             3.0  8.4  0.346410  8.2  8.200  8.2  8.5   8.8

Vanilla               6.0  5.7  2.710719  2.3  4.025  5.6  6.8  10.0


            Texture Rating            ...             Total Rating
\
                     count      mean  ...    75%  max        count
mean
Base Flavor                           ...

Chocolate              3.0  7.233333  ...  7.350  7.6          3.0
15.70
Vanilla                6.0  5.650000  ...  6.375  8.0          6.0
11.35


                  std   min     25%   50%     75%   max
Base Flavor
Chocolate    0.781025  15.2  15.250  15.3  15.950  16.6
Vanilla      4.263684   5.7   9.025  11.1  13.175  18.0
```

```
[2 rows x 24 columns]
```

Merge , Join and Concatenate

```
df1 = pd.read_csv(r"D:\gami\LOTR (1).csv")
df2 = pd.read_csv(r"D:\gami\LOTR 2 (1).csv")

print(df1)
print('\n')
print(df2)

   FellowshipID FirstName     Skills
0          1001     Frodo     Hiding
1          1002   Samwise  Gardening
2          1003   Gandalf     Spells
3          1004    Pippin  Fireworks


   FellowshipID FirstName   Age
0          1001     Frodo    50
1          1002   Samwise    39
2          1006   Legolas  2931
3          1007    Elrond  6520
4          1008  Barromir    51

# merging 1st dataframe to 2nd
df1.merge(df2)

   FellowshipID FirstName     Skills  Age
0          1001     Frodo     Hiding   50
1          1002   Samwise  Gardening   39

# we did'nt explicitly ssays join using fellowshipID
df1.merge(df2,how = 'inner') # inner is default

   FellowshipID FirstName     Skills  Age
0          1001     Frodo     Hiding   50
1          1002   Samwise  Gardening   39

df1.merge(df2 , how = 'inner' , on = 'FellowshipID')
# if we does'nt join them on basis of firstname , then it will give
two diff columns

   FellowshipID FirstName_x     Skills FirstName_y  Age
0          1001       Frodo     Hiding       Frodo   50
1          1002     Samwise  Gardening     Samwise   39

df1.merge(df2 , how = 'inner' , on = ['FellowshipID','FirstName'])
```

```
   FellowshipID FirstName     Skills  Age
0          1001     Frodo     Hiding   50
1          1002   Samwise  Gardening   39
```

```
df1.merge(df2 , how = 'outer' ) # their are left and right joins too
```

```
   FellowshipID FirstName     Skills     Age
0          1001     Frodo     Hiding    50.0
1          1002   Samwise  Gardening    39.0
2          1003   Gandalf     Spells     NaN
3          1004    Pippin  Fireworks     NaN
4          1006   Legolas        NaN  2931.0
5          1007    Elrond        NaN  6520.0
6          1008  Barromir        NaN    51.0
```

```
# their is also a cross join
df1.merge(df2 , how = 'cross')
```

```
    FellowshipID_x FirstName_x     Skills  FellowshipID_y FirstName_y
Age
0             1001       Frodo     Hiding            1001       Frodo
50
1             1001       Frodo     Hiding            1002     Samwise
39
2             1001       Frodo     Hiding            1006     Legolas
2931
3             1001       Frodo     Hiding            1007      Elrond
6520
4             1001       Frodo     Hiding            1008    Barromir
51
5             1002     Samwise  Gardening            1001       Frodo
50
6             1002     Samwise  Gardening            1002     Samwise
39
7             1002     Samwise  Gardening            1006     Legolas
2931
8             1002     Samwise  Gardening            1007      Elrond
6520
9             1002     Samwise  Gardening            1008    Barromir
51
10            1003     Gandalf     Spells            1001       Frodo
50
11            1003     Gandalf     Spells            1002     Samwise
39
12            1003     Gandalf     Spells            1006     Legolas
2931
13            1003     Gandalf     Spells            1007      Elrond
6520
14            1003     Gandalf     Spells            1008    Barromir
51
```

```
15              1004      Pippin  Fireworks            1001       Frodo
50
16              1004      Pippin  Fireworks            1002     Samwise
39
17              1004      Pippin  Fireworks            1006     Legolas
2931
18              1004      Pippin  Fireworks            1007      Elrond
6520
19              1004      Pippin  Fireworks            1008    Barromir
51

# let see join function
# merge works better with columns
# joins works better with indexes
# df1.join(df2) :  shows error , cannot distinguished which is which
(['FellowshipID', 'FirstName'])
df1.join(df2 , on = 'FellowshipID' , how = 'outer' , lsuffix  =
'_left',rsuffix = '_right')

     FellowshipID  FellowshipID_left FirstName_left     Skills  \
0.0          1001             1001.0          Frodo     Hiding
1.0          1002             1002.0        Samwise  Gardening
2.0          1003             1003.0        Gandalf     Spells
3.0          1004             1004.0         Pippin  Fireworks
NaN             0                NaN            NaN        NaN
NaN             1                NaN            NaN        NaN
NaN             2                NaN            NaN        NaN
NaN             3                NaN            NaN        NaN
NaN             4                NaN            NaN        NaN

     FellowshipID_right FirstName_right     Age
0.0                 NaN             NaN     NaN
1.0                 NaN             NaN     NaN
2.0                 NaN             NaN     NaN
3.0                 NaN             NaN     NaN
NaN              1001.0           Frodo    50.0
NaN              1002.0         Samwise    39.0
NaN              1006.0         Legolas  2931.0
NaN              1007.0          Elrond  6520.0
NaN              1008.0        Barromir    51.0

# doing two steps at a time , joining and setting index
df4 =
df1.set_index('FellowshipID').join(df2.set_index('FellowshipID'),lsuff
ix  = '_left',rsuffix = '_right')

df4

             FirstName_left     Skills FirstName_right   Age
FellowshipID
```

```
1001                      Frodo     Hiding            Frodo  50.0
1002                    Samwise  Gardening          Samwise  39.0
1003                    Gandalf     Spells              NaN   NaN
1004                     Pippin  Fireworks              NaN   NaN

# concatenate : putting one dataframe above other
pd.concat([df1,df2])

   FellowshipID FirstName     Skills     Age
0          1001     Frodo     Hiding     NaN
1          1002   Samwise  Gardening     NaN
2          1003   Gandalf     Spells     NaN
3          1004    Pippin  Fireworks     NaN
0          1001     Frodo        NaN    50.0
1          1002   Samwise        NaN    39.0
2          1006   Legolas        NaN  2931.0
3          1007    Elrond        NaN  6520.0
4          1008  Barromir        NaN    51.0

# can also do inner , outer ... etc joins
pd.concat([df1,df2] , join = 'inner')

   FellowshipID FirstName
0          1001     Frodo
1          1002   Samwise
2          1003   Gandalf
3          1004    Pippin
0          1001     Frodo
1          1002   Samwise
2          1006   Legolas
3          1007    Elrond
4          1008  Barromir

# can also change based on axis
pd.concat([df1,df2] , join = 'inner' , axis = 1)

   FellowshipID FirstName     Skills  FellowshipID FirstName   Age
0          1001     Frodo     Hiding          1001     Frodo    50
1          1002   Samwise  Gardening          1002   Samwise    39
2          1003   Gandalf     Spells          1006   Legolas  2931
3          1004    Pippin  Fireworks          1007    Elrond  6520

# append is outdated
```

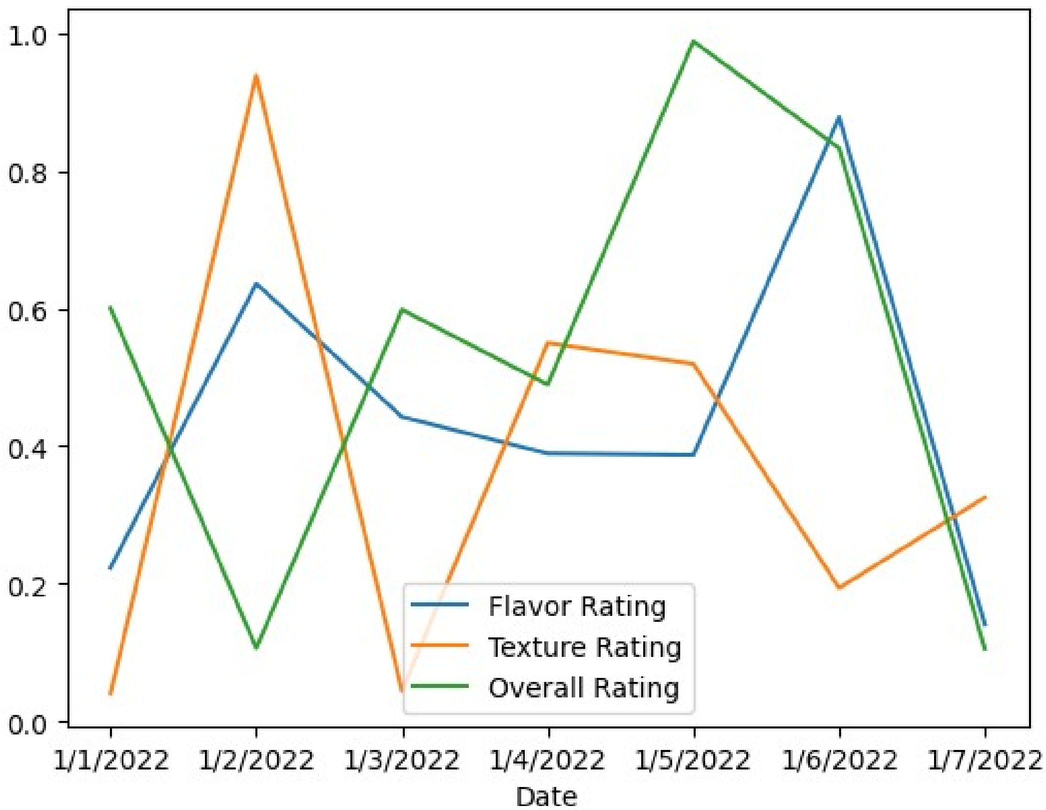
Pandas Visualization

```
import matplotlib.pyplot as plt
import seaborn as sns
```

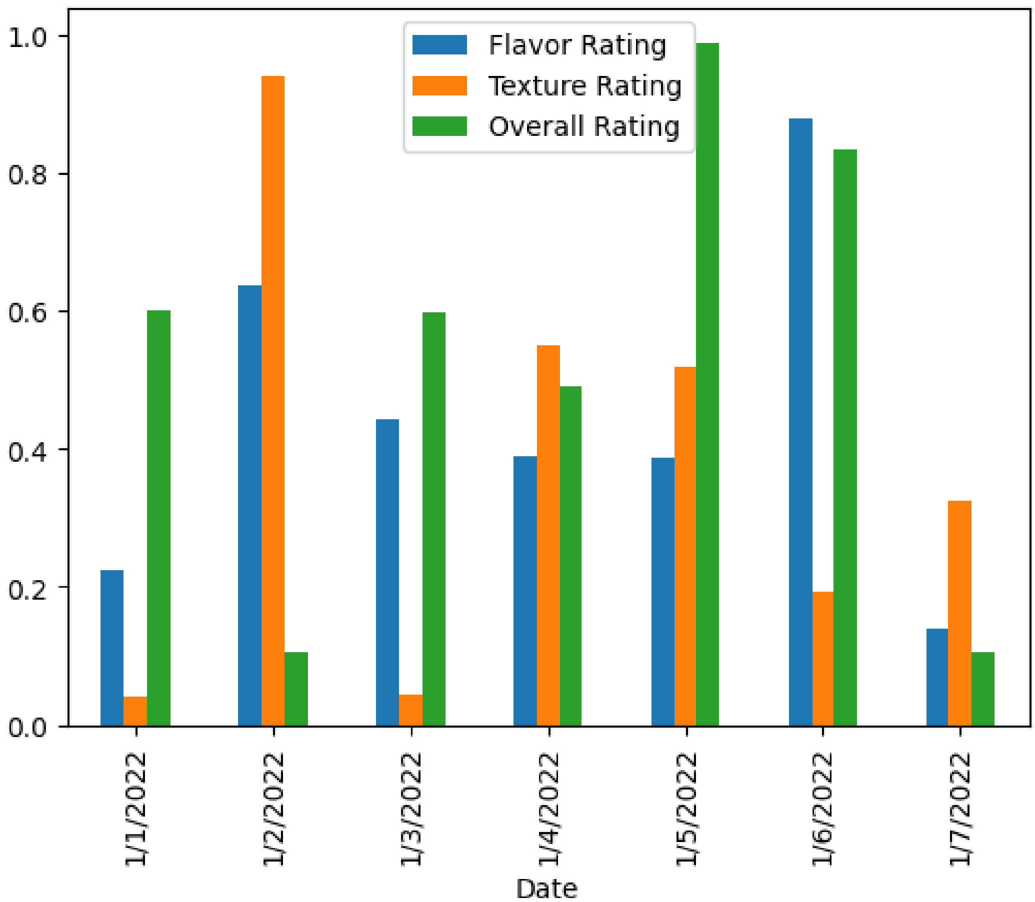
```
df = pd.read_csv(r"D:\gami\Ice Cream Ratings (1).csv")
df = df.set_index('Date')
df
```

Date	Flavor Rating	Texture Rating	Overall Rating
1/1/2022	0.223090	0.040220	0.600129
1/2/2022	0.635886	0.938476	0.106264
1/3/2022	0.442323	0.044154	0.598112
1/4/2022	0.389128	0.549676	0.489353
1/5/2022	0.386887	0.519439	0.988280
1/6/2022	0.877984	0.193588	0.832827
1/7/2022	0.140995	0.325110	0.105147

```
df.plot(); # plotting using plot()
```

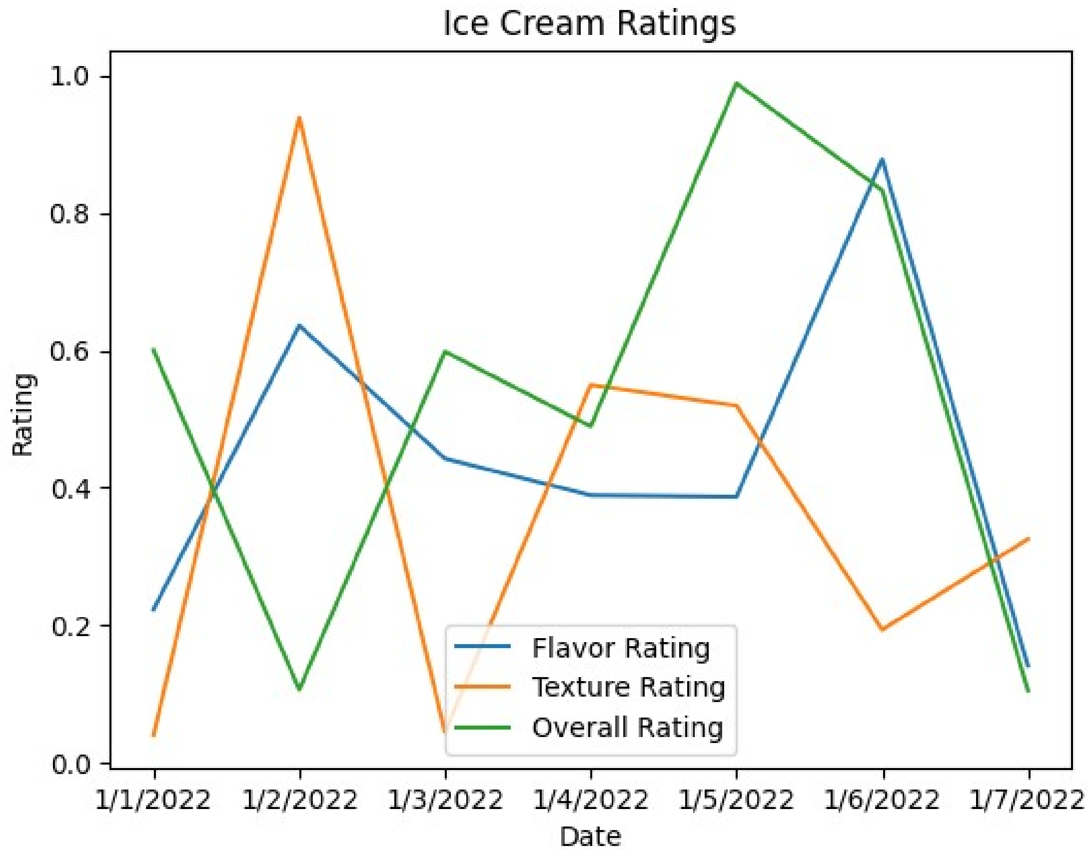



```
# we can use kind to change type of graph
df.plot(kind = 'bar');
```

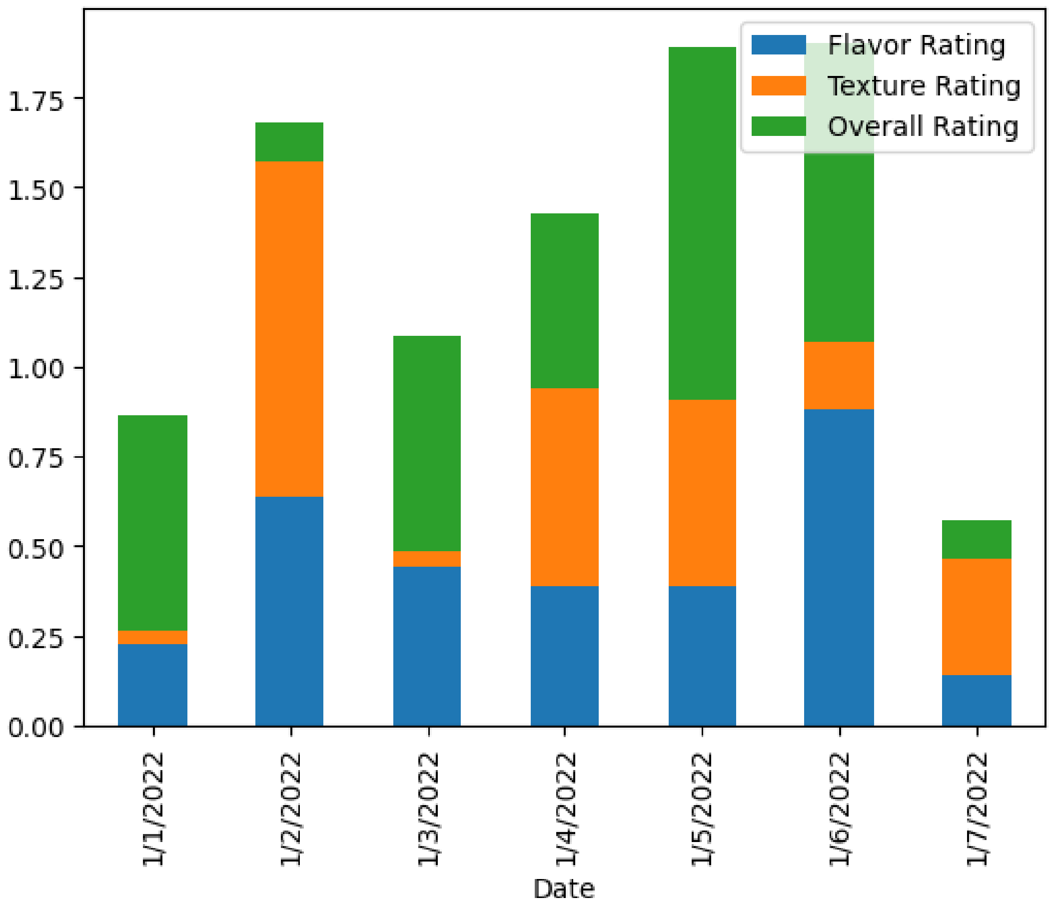



```
# we can also divide into subplots
df.plot(kind = 'bar' , subplots = True); # can also do df.plot.bar();
```

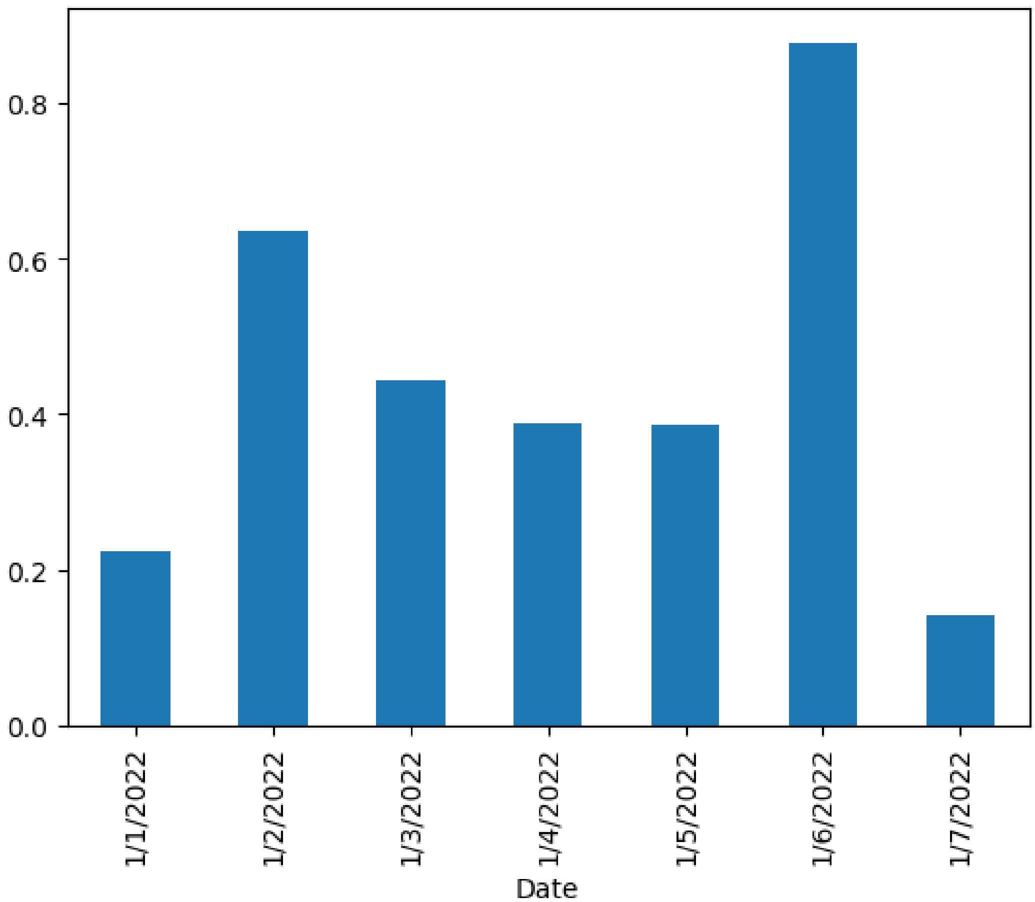

```
# can add title and labels
df.plot(kind = 'line' , title = 'Ice Cream Ratings' ,xlabel = 'Date' ,
ylabel = 'Rating');
```



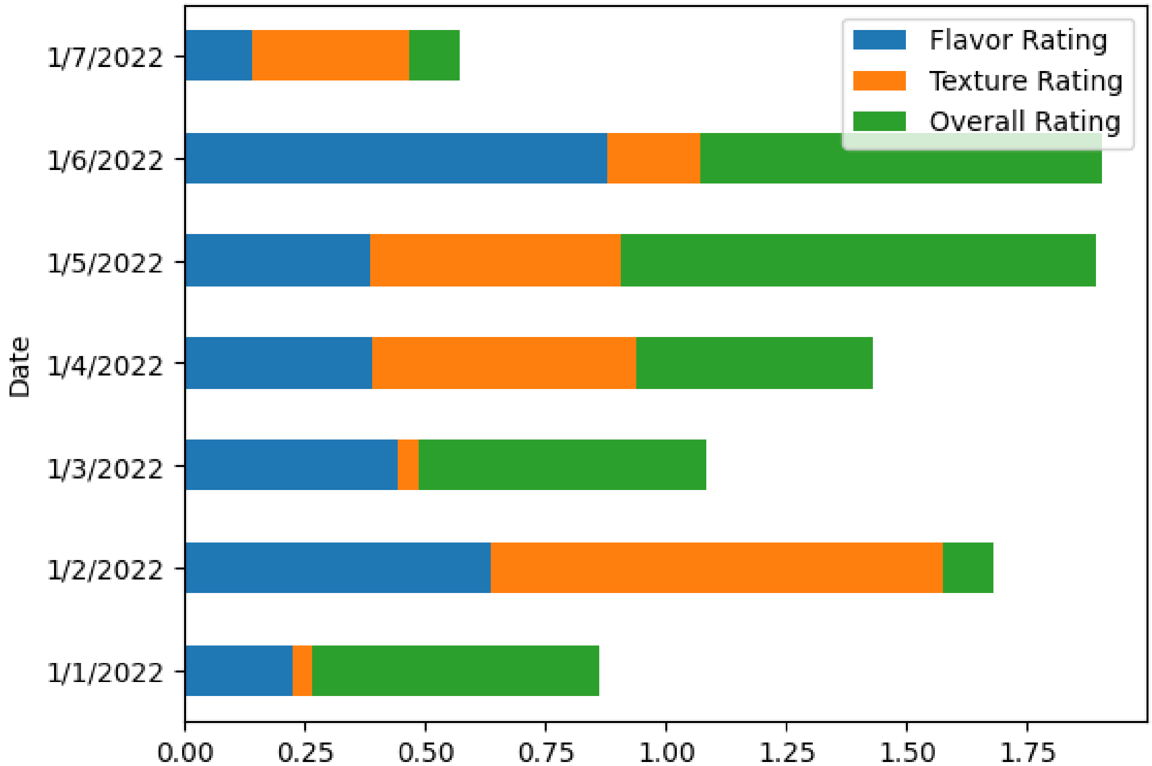

```
# for a stacked barchart
df.plot(kind = 'bar' , stacked = True);
```



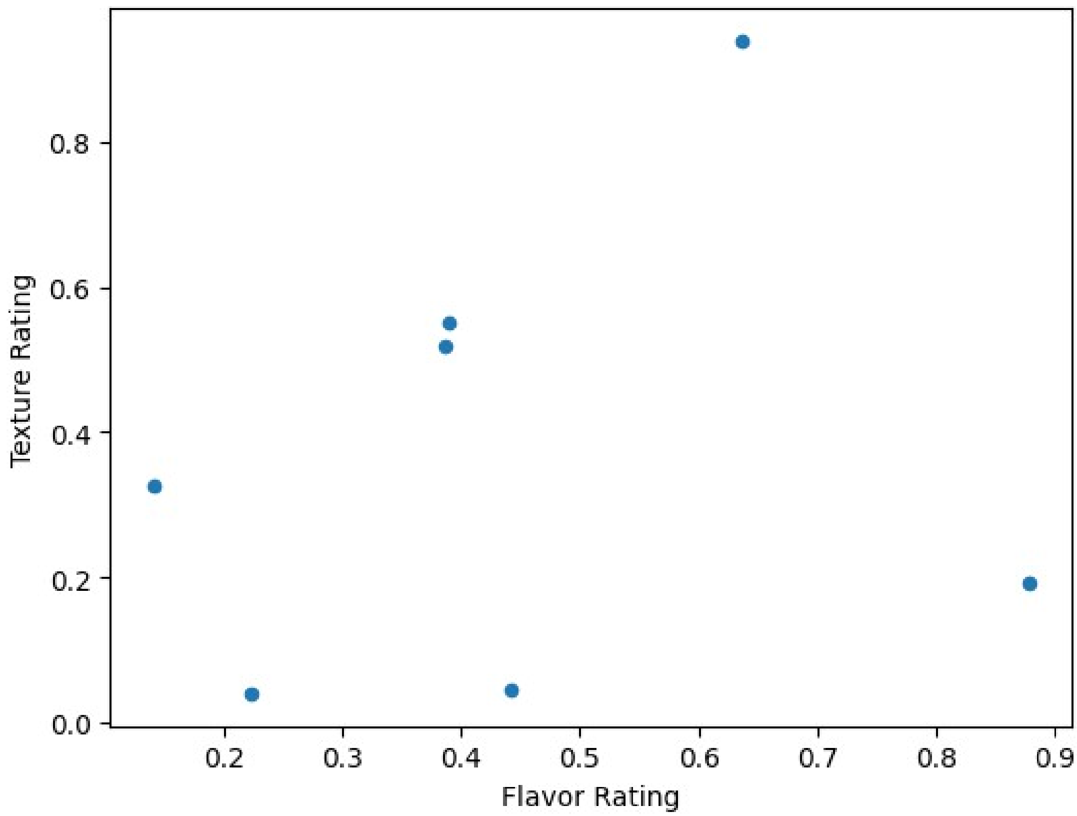

```
# for specifying column
df['Flavor Rating'].plot(kind = 'bar' , stacked = True);
```



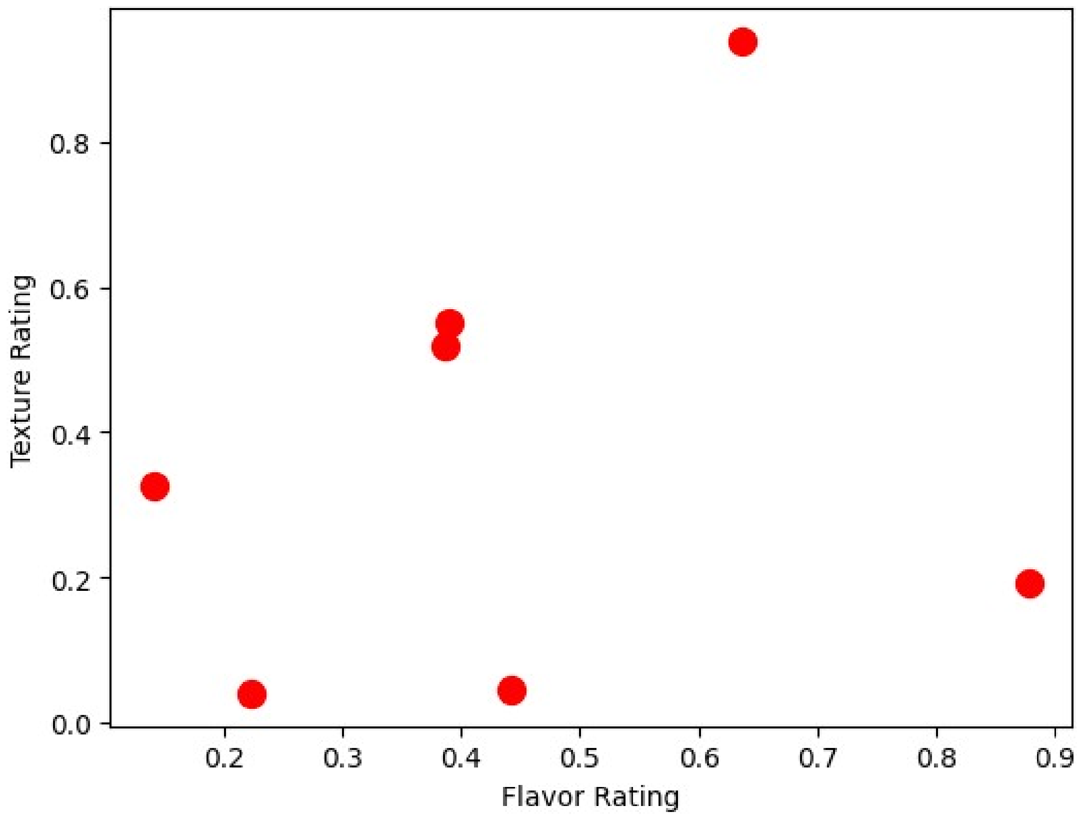

```python
# for changing into horzontal barchart
df.plot.barh(stacked = True);
```



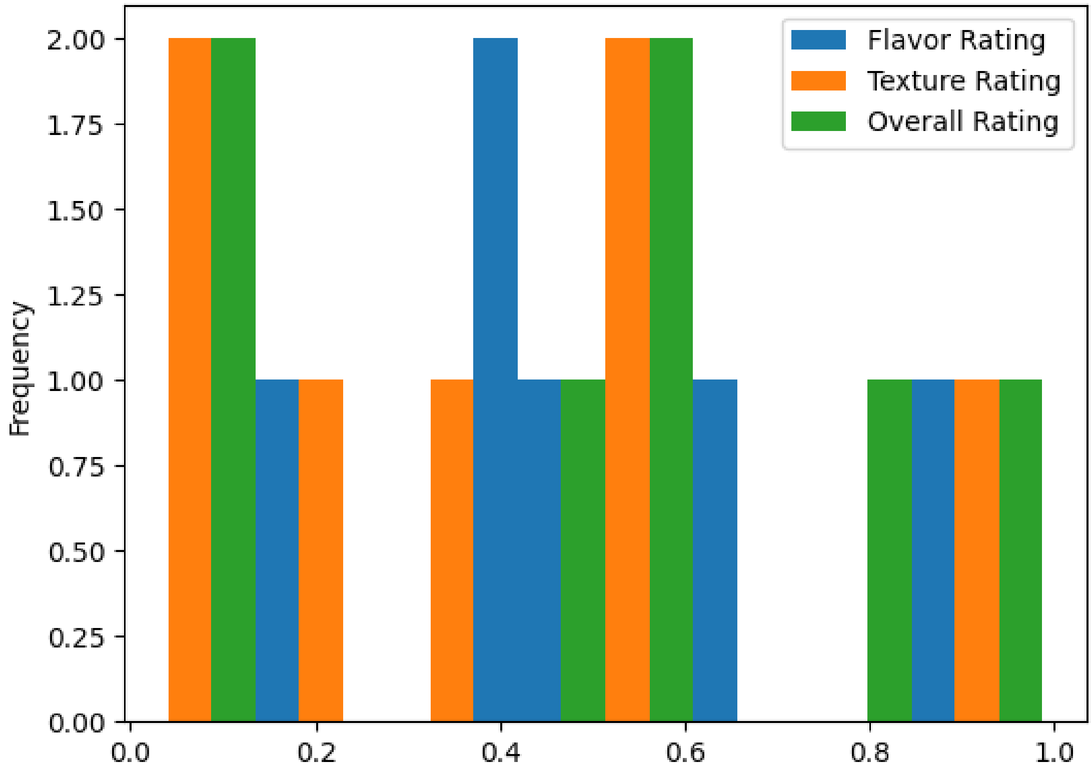

```
# for scatter plot
df.plot.scatter(x = 'Flavor Rating' , y = 'Texture Rating');
```




```
# can also change size of dots and color of dots
df.plot.scatter(x = 'Flavor Rating' , y = 'Texture Rating' , s = 100 ,
c = 'Red');
```

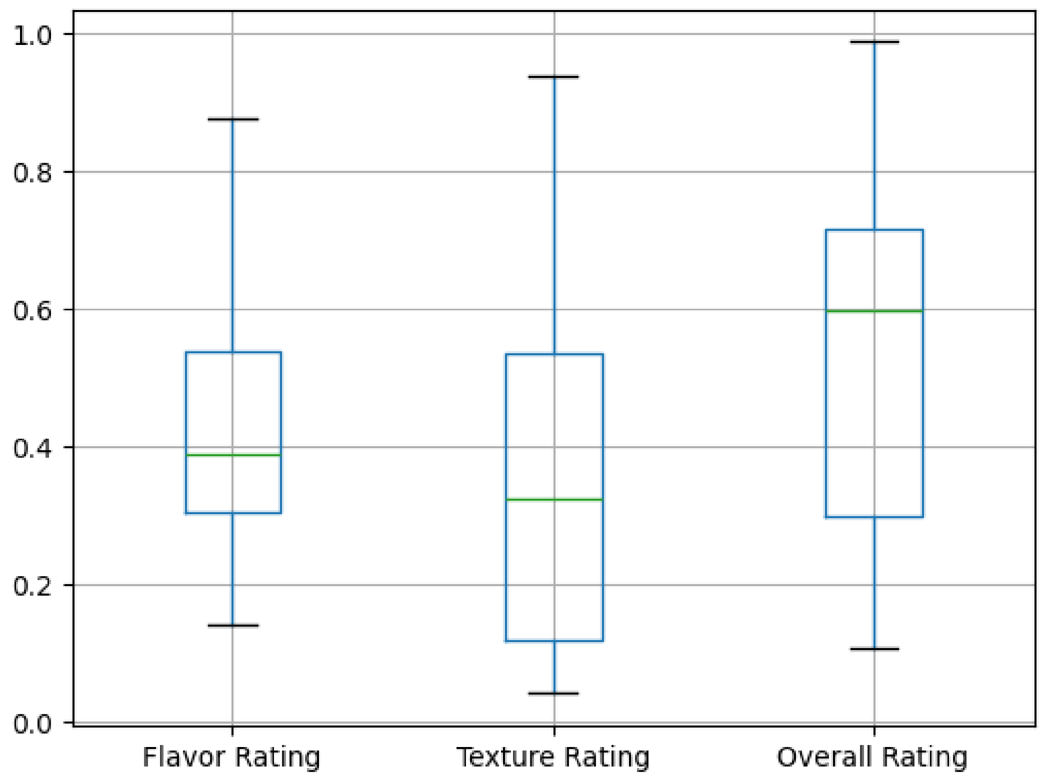



```
# histograms and bins
df.plot.hist(bins = 20);
```



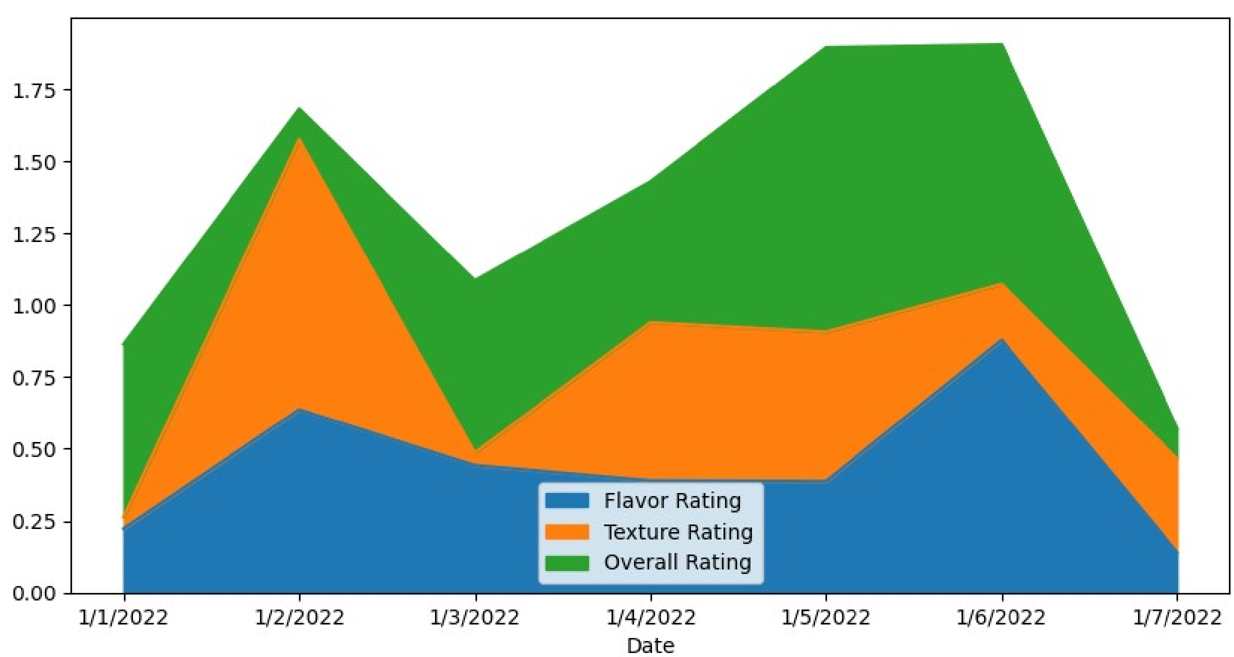

```
# boxplot
df.boxplot() # plot function not needed
```

```
<Axes: >
```




```
# area chart
df.plot.area(figsize = (10,5)) # can change size using figsize

<Axes: xlabel='Date'>
```



```python
# pie chart
#df.plot.pie() will show error as we need to specify what col we are working with
df.plot.pie(y = 'Flavor Rating' , figsize = (10,10))
```

```
<Axes: ylabel='Flavor Rating'>
```