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

```
print(type(12 + 10.25))
print(type(12 + 3j))
<class 'int'>
<class 'float'>
<class 'complex'>
# boolean: True, False
print(type(True))
print(type(False))
# ex :
print(type(1>5))
print(1>5)
<class 'bool'>
<class 'bool'>
<class 'bool'>
False
# 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
llo W
```

```
Hello WorldHello WorldHello World
Hello WorldHello World
# 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
# 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
# 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) \text{ and } (50 > 10))
print(2,(70 > 50)) and (50 > 10)
# or
print(3,(10 > 50) \text{ or } (50 > 10))
print(4,(70 > 50) \text{ or } (50 > 10))
# not
print(5, not(50>10))
1 False
2 True
3 True
4 True
5 False
```

```
# 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

```
# it's like: if this then this , if not , then this , if not , then
anything else
if 25 > 10:
    print('It worked!')
It worked!
if 25 < 10:
    print('It worked!')
else:
    print('It didnt worked!....')
It didnt worked!....
# 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
# If else statement in one line
print('It worked') if 10>30 else print('It did not worked...')
It did not worked...
# 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!</pre>
```

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
3
4
5
for number in integer:
    print('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']
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']
# 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

```
# 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
```

```
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
2
3
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
# 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'>
# 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
                      EASTERN EUROPE
            Albania
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
           Zimbabwe
                      SUB-SAHARAN AFRICA
227
```

```
[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
                            ASIA (EX. NEAR EAST)
0
        Afghanistan
1
           Albania
                     EASTERN EUROPE
2
            Algeria
                      NORTHERN AFRICA
3
     American Samoa
                      OCEANIA
                      WESTERN EUROPE
4
            Andorra
222
         West Bank
                      NEAR EAST
223 Western Sahara
                      NORTHERN AFRICA
224
             Yemen
                      NEAR EAST
                      SUB-SAHARAN AFRICA
225
             Zambia
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 \
   {'Genre': 'Action', 'Gross': '$491,898', 'IMDB...
                                           Ferdinand
   {'popcornscore': 49, 'rating': 'PG', 'tomatosc...
                                  Fifty Shades Freed
  {'Genre': 'Drama', 'Gross': 'unknown', 'IMDB M...
                   Film Stars Don'T Die In Liverpool
   {'popcornscore': 69, 'rating': 'R', 'tomatosco... ...
                                  The 15:17 To Paris \
   {'Genre': 'Drama', 'Gross': 'unknown', 'IMDB M...
                                        The Commuter \
   {'popcornscore': 48, 'rating': 'PG13', 'tomato...
                                 The Disaster Artist \
   {'popcornscore': 89, 'rating': 'R', 'tomatosco...
                                The Greatest Showman \
  {'Genre': 'Biography', 'Gross': '$627,248', 'I...
                              The Insult (L'Insulte) \
  {'popcornscore': 86, 'rating': 'R', 'tomatosco...
                                            The Post \
  {'Genre': 'Biography', 'Gross': '$463,228', 'I...
                                  The Shape Of Water \
  {'Genre': 'Adventure', 'Gross': '$448,287', 'I...
           Three Billboards Outside Ebbing, Missouri \
  {'popcornscore': 87, 'rating': 'R', 'tomatosco...
                           Till The End Of The World \
  {'popcornscore': -1, 'rating': 'NR', 'tomatosc...
                                          Winchester
0 {'Genre': 'Biography', 'Gross': '$696,786', 'I...
[1 rows x 38 columns]
```

```
# 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
                                             Capital
                          Country
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 Aaiún
231
       46
          YEM
                            Yemen
                                               Sanaa
232
       63
           ZMB
                           Zambia
                                              Lusaka
233
       74 ZWE
                         Zimbabwe
                                              Harare
[234 rows x 4 columns]
# 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):
              Non-Null Count Dtype
 #
     Column
     _ _ _ _ _
                              _ _ _ _ _
     Rank
              234 non-null
                              int64
 1
     CCA3
              234 non-null
                              object
 2
     Country 234 non-null
                              object
     Capital 234 non-null
 3
                              object
dtypes: int64(1), object(3)
memory usage: 7.4+ KB
# for shape
df.shape # it's an attribute so no parenthesis needed
(234, 4)
df.head(4) # we can get first some values
   Rank CCA3
                                Capital
                     Country
0
     36 AFG
                 Afghanistan
                                  Kabul
    138 ALB
                     Albania
1
                                 Tirana
```

```
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
           ZMB
                  Zambia Lusaka
       63
       74 ZWE Zimbabwe Harare
233
# 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
        63
232
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))
     23
1
     45
     67
dtype: int64
<class 'pandas.core.series.Series'>
```

```
# 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
             NaN
3rd best
4th best
             NaN
            20.0
best
dtype: float64
<class 'pandas.core.series.Series'>
# 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
                   1
0 osshowatt
               water
1 cyndaguil
                fire
2 hoothoot flying
starters data = pd.DataFrame(new starters , columns=('Starters' ,
'Type'))
starters data # Columns added but index can also be added
    Starters
                Type
0 osshowatt
               water
```

```
fire
1 cyndaguil
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
                         2
           Bulbasaur
Pokemon 3
                         3
            chimchar
```

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
      138 ALB
1
                                             Tirana
                          Albania
                                                        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
                                                       Africa
                                             Harare
     2022 Population 2020 Population 2015 Population 2010
Population
          41128771.0
                           38972230.0
                                            33753499.0
28189672.0
           2842321.0
                            2866849.0
                                             2882481.0
2913399.0
```

2 35856344.	44903225.0	43451666.0	9 39543154.0	
3	44273.0	46189.0	51368.0	
54849.0 4	79824.0	77700.0	9 71746.0	
71519.0				
229	11572.0	11655.0	0 12182.0	
13142.0				
230 413296.0	575986.0	556048.0	9 491824.0	
231 24743946.	33696614.0	32284046.0	0 28516545.0	
232 13792086.	20017675.0	18927715.0	0 NaN	
233 12839771.	16320537.0	15669666.0	0 14154937.0	
		1990 Population	n 1980 Population	1970
Populatio 0	19542982.0	10694796.0	0 12486631.0	
10752971.	3182021.0	3295066.0	0 2941651.0	
2324731.0	30774621.0	25518074.0	0 18739378.0	
13795915. 3	0 58230.0	47818.0	9 32886.0	
27075.0 4 19860.0	66097.0	53569.0	9 35611.0	
229	14723.0	13454.0	0 11315.0	
9377.0 230	270375.0	178529.0	0 116775.0	
76371.0 231	18628700.0	13375121.0	9204938.0	
6843607.0				
232 4281671.0		7686401.0		
233 5202918.0	11834676.0	10113893.0	0 7049926.0	
		ity (per km²) (Growth Rate World	Population
	e 2230.0	63.0587	1.0257	
0.52 1 2	8748.0	98.8702	0.9957	

0.04			
2	2381741.0	18.8531	1.0164
0.56			
3	199.0	222.4774	0.9831
0.00	400.0		
4	468.0	170.5641	1.0100
0.00			
220	142.0	01 4020	0 0053
229	142.0	81.4930	0.9953
0.00	266000 0	2 1654	1 0104
230	266000.0	2.1654	1.0184
0.01	507060 0	62, 6222	1 0017
231	527968.0	63.8232	1.0217
0.42	752612.0	26 5076	1 0000
232	752612.0	26.5976	1.0280
0.25	200757 0	41 7665	1 0204
233	390757.0	41.7665	1.0204
0.20			

[234 rows x 17 columns]

Filtering

 $df[df['Rank'] \le 10]$ # for filtering , as we can see , we have to write condition inside df

	Rank	CCA3		Country		Capital	Co	ontinent
16	8	BGD	Ba	ngladesh		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	I	ndonesia		Jakarta		Asia
131	10	MEX		Mexico	Me	xico City	North	America
149	6	NGA		Nigeria		Abuja		Africa
156		PAK		Pakistan		Islamabad		Asia
171	9	RUS		Russia		Moscow		Europe
221	3	USA	Unite	d States	Washing	iton, D.C.	North	America
	2022			2020 5		2015 5		2010
D			ation	2020 Pop	ulation	2015 Popu	lation	2010
	ılatior	-	4 00	1 674	210 00	1 5700	00 00	
16			4e+08	1.6/4	210e+08	1.5/83	00e+08	
	3911e+		F 00	2 121	062 00	2 0510	00 00	
27			5e+08	2.131	.963e+08	2.0518	82e+08	
	3535e+		700	1 424	02000	1 2027	1500	
41			7e+09	1.424	930e+09	1.3937	15e+09	
	ŀ8191e+		22 00	1 200	207 00	1 2220		
92			3e+09	1.396	387e+09	1.3228	67e+09	
	ŀ0614e+		200	2 710	E00 00	2 5000	20 00	
93			.3e+08	2./18	580e+08	2.5909	20e+08	
2.44	ŀ0162e+	-08						

	1.275041	Le+08	1.259983e+	98	1.20149	99e+08		
149	324e+08 2.185412	2e+08	2.083274e+	98	1.83995	8e+08		
	529e+08 2.358249	e+08	2.271967e+	98 2	2.10969	3e+08		
	545e+08 1.447133	8e+08	1.456173e+	98	1.44668	34e+08		
1.432	426e+08		3.359420e+					
	828e+08	7E+00	3.3394200+	. vo	3.24007	00+00		
			90 Populati	on 1980	0 Popul	ation	1970	
	ation \ 1.291933		1.071477e+	98	83929	765.0		
67541	860.0 1.758737		1.507064e+	คล	122288	1383 A		
96369	875.0							
82253	4450.0		1.153704e+		982372	2400.0		
	1.059634 1301.0	le+09	Na	aN		NaN		
93		le+08	1.821599e+	98	148177	7096.0		
131	9.787344	le+07	8.172043e+	97	67705	186.0		
	306.0 1.228520	e+08	9.521426e+	97	72951	L439.0		
	264.0 1.543699	0e+08	1.154141e+	98	80624	1057.0		
59290	872.0 1.468448		1.480057e+	คล	138257	7420 0		
13009	3010.0							
	8340.0	e+08	2.480837e+	98	223140	0.8100		
		Density	(per km²)	Growth	Rate	World	Population	
Perce 16	ntage 147570.0		1160.0350	1	.0108			
2.15 27	8515767.0		25.2841	1	.0046			
2.70								
41 17.88			146.8933		.0000			
92 17.77	3287590.0		431.0675	1	.0068			
93 3.45	1904569.0		144.6529	1	.0064			
131	1964375.0		64.9082	1	.0063			
1.60 149	923768.0		236.5759	1	.0241			

```
2.74
156
                         267.4018
                                        1.0191
      881912.0
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
                                     Continent 2022 Population \
                        Capital
         BRA Brazil
                        Brasilia South America
27
      7
                                                   2.153135e+08
      2 IND
92
               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
                    1980 Population 1970 Population Area (km²) \
   1990 Population
        150706446.0
                         122288383.0
                                          96369875.0
27
                                                       8515767.0
92
                                         557501301.0
               NaN
                                NaN
                                                       3287590.0
   Density (per km<sup>2</sup>) Growth Rate World Population Percentage
27
             25.2841
                                                          2.70
                           1.0046
92
            431.0675
                           1.0068
                                                         17.77
# we can also use contain function for same task, It works like LIKE
clause in SOL
df[df['Country'].str.contains('United')]
    Rank CCA3
                                                      Capital
                                    Country
Continent
          1
                       United Arab Emirates
219
      97 ARE
                                                    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
                            57210442.0
                                              56326328.0
220
          58850043.0
55650166.0
221
         282398554.0
                           248083732.0
                                             223140018.0
200328340.0
222
            108185.0
                              100685.0
                                                 96640.0
63446.0
     Area (km²)
                 Density (per km<sup>2</sup>) Growth Rate World Population
Percentage
                           112.9322
219
        83600.0
                                          1.0081
0.12
220
                           277.9289
                                          1.0034
       242900.0
0.85
      9372610.0
                            36.0935
                                          1.0038
221
4.24
222
                                          0.9937
          347.0
                           286.6427
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
                    213 ASM
American Samoa
                                      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 33696614.0	46	YEM	Sanaa	Asia	
Zambia 20017675.0	63	ZMB	Lusaka	Africa	
Zimbabwe 16320537.0	74	ZWE	Harare	Africa	
Population \ Country	2020	Population	2015 Populat	ion 2010	
Afghanistan		38972230.0	3375349	9.0	28189672.0
Albania		2866849.0	288248	1.0	2913399.0
Algeria		43451666.0	3954315	4.0	35856344.0
American Samoa		46189.0	5136	8.0	54849.0
Andorra		77700.0	7174	6.0	71519.0
Wallis and Futuna		11655.0	1218	2.0	13142.0
Western Sahara		556048.0	49182	4.0	413296.0
Yemen		32284046.0	2851654	5.0	24743946.0
Zambia		18927715.0		NaN	13792086.0
Zimbabwe		15669666.0	1415493	7.0	12839771.0
Population \ Country	2000	Population	1990 Populat	ion 1980	
Afghanistan		19542982.0	1069479	6.0	12486631.0
Albania		3182021.0	329506	6.0	2941651.0
Algeria		30774621.0	2551807	4.0	18739378.0
American Samoa		58230.0	4781	8.0	32886.0

Andorra	66097.0	53569.0	35611.0
Alluot i a	00097.0	33309.0	33011.0
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	y (per km²) \
Country Afghanistan Albania Algeria American Samoa Andorra	10752971.0 2324731.0 13795915.0 27075.0 19860.0	652230.0 28748.0 2381741.0 199.0 468.0	63.0587 98.8702 18.8531 222.4774 170.5641
Wallis and Futuna Western Sahara Yemen Zambia Zimbabwe	9377.0 76371.0 6843607.0 4281671.0 5202918.0	142.0 266000.0 527968.0 752612.0 390757.0	81.4930 2.1654 63.8232 26.5976 41.7665
	Growth Rate Wor	ld Population Perce	ntage
Country Afghanistan Albania Algeria American Samoa Andorra Wallis and Futuna Western Sahara	1.0257 0.9957 1.0164 0.9831 1.0100 0.9953 1.0184		0.52 0.04 0.56 0.00 0.00 0.00 0.01
Yemen Zambia Zimbabwe	1.0217 1.0280 1.0204		0.42 0.25 0.20
[234 rows x 16 colu	ımns]		
<pre>df2.filter(items =</pre>	['Continent' , '	CCA3'], $axis = 0$)	
Population, 2015 Po	pulation, 2010 Po	inent, 2022 Populat opulation, 2000 Pop opulation, Area (km	ulation, 1990

```
km<sup>2</sup>), Growth Rate, World Population Percentagel
Index: []
df2.filter(items = ['Continent' , 'CCA3'] ,axis = 1) # default axis =
                 Continent CCA3
Country
Afghanistan
                      Asia AFG
Albania
                    Europe ALB
Algeria
                    Africa
                            DZA
                   Oceania ASM
American Samoa
Andorra
                    Europe AND
Wallis and Futuna
                           WLF
                   Oceania
Western Sahara
                    Africa ESH
Yemen
                      Asia
                           YEM
                    Africa
Zambia
                            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 \
              14154937.0 12839771.0
Zimbabwe
                                                11834676.0
10113893.0
         1980 Population 1970 Population Area (km<sup>2</sup>) Density (per
km<sup>2</sup>) \
                                5202918.0
Zimbabwe
               7049926.0
                                             390757.0
41.7665
         Growth Rate World Population Percentage
Zimbabwe
              1.0204
# 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		L	ondon		
Europe United States				Machin			Nost	
America		3	USA	Wasniin	ig con ,	D.C.	NOTE	1
United States Virgin I America	[slands	200	VIR	Charlo	otte A	malie	North	1
	2	2022 I	Popula	ation	2020	Popula	tion	\
Country United Arab Emirates United Kingdom United States United States Virgin I	Islands		675089 382898	936.0 357.0	3	670594 359420	74.0 03.0	
Country	2	2015 I	Popula	ation	2010	Popula	tion	\
Country United Arab Emirates United Kingdom United States United States Virgin I	Islands		652243 246077	364.0 776.0		627600 111828	39.0 45.0	
	2	2000 I	Popula	ation	1990	Popula	tion	\
Country United Arab Emirates United Kingdom United States United States Virgin I	Islands		588500 823985	943.0 554.0	2	572104 480837	42.0 32.0	
]	L980 I	Popula	ation	1970	Popula	tion	Area
(km²) \ Country			·			·		
United Arab Emirates			10140	948.0		2980	84.0	
83600.0 United Kingdom 242900.0		ļ	563263	328.0		556501	66.0	
United States 9372610.0		22	231400	918.0	2	003283	40.0	
United States Virgin I 347.0	[slands		966	540.0		634	46.0	
	D)ensi	ty (pe	er km²)	Gro	wth Ra	te \	
Country United Arab Emirates United Kingdom United States United States Virgin I	Islands		27	77.9289 36.0935) 5	$1.00 \\ 1.00$	34 38	
Country	### Table Stands 200 VIR Charlotte Amalie North							

```
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 (km<sup>2</sup>)
                                         9372610.0
Density (per km<sup>2</sup>)
                                           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 (km<sup>2</sup>)
                                     199.0
                                  222.4774
Density (per km<sup>2</sup>)
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) #</pre>
condition -> sorting -> order by -> ascending / desending
```

171 9	BRA NGA PAK IDN USA U IND	Rus Banglad Bra Nige Pakis Indone nited Sta	zil ria tan	Capita Mosco Dhak Brasili Abuj Islamaba Jakart Ington, D.O New Delk Beijir	ka ia South ja ad ta C. North	Europe Asia America Africa Asia	
2022 Population 171	Populate 1.447133e 1.447133e 1.4684 1.711864e 1.408 1.153135e 1.408 1.358249e 1.408 1.358249e 1.408 1.417173e 1.468448e 1.46844e 1.4684e 1.4684e 1.4684e 1.4684e 1.4684e 1.	ion 2020 1+08 1 1+08 2 1+08 2 1+08 2 1+08 2 1+08 3 1+09 1 1+09 1 1+09 1 1+09 1 1+09 1 1+09 1 1+09 1	Population .456173e+0 .674210e+0 .131963e+0 .083274e+0 .271967e+0 .718580e+0 .359420e+0 .396387e+0 .424930e+0	on 2015 Po 98	pulation 46684e+08 78300e+08 51882e+08 89958e+08 99693e+08 90920e+08 46078e+08 22867e+09 93715e+09	2010	
59290872. 93 2 115228394 221 2 200328340	0 .140724e .0 .823986e .0 .059634e	+08 1 +08 2	.821599e+0	98 148	3177096.0 3140018.0 NaN		

```
41
        1.264099e+09
                         1.153704e+09
                                            982372466.0
822534450.0
     Area (km²)
                 Density (per km<sup>2</sup>) 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 =</pre>
False)
# due to order of importance , the table does'nt changed much
     Rank CCA3
                      Country
                                         Capital
                                                       Continent \
171
        9
           RUS
                        Russia
                                          Moscow
                                                          Europe
        8
           BGD
                   Bangladesh
16
                                           Dhaka
                                                            Asia
27
        7
           BRA
                       Brazil
                                        Brasilia South America
                      Nigeria
                                                          Africa
149
        6
           NGA
                                           Abuja
                                       Islamabad
156
        5
           PAK
                     Pakistan
                                                            Asia
        4
93
           IDN
                    Indonesia
                                         Jakarta
                                                            Asia
221
        3
           USA United States
                                Washington, D.C.
                                                  North America
        2
92
           IND
                        India
                                       New Delhi
                                                            Asia
41
        1
          CHN
                        China
                                         Beijing
                                                            Asia
     2022 Population 2020 Population 2015 Population 2010
Population \
        1.447133e+08
                          1.456173e+08
                                           1.446684e+08
171
1.432426e+08
        1.711864e+08
                          1.674210e+08
                                           1.578300e+08
16
1.483911e+08
        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
```

	2.3582496	+08	2.271967e+	08	2.10969	93e+08		
93	545e+08 2.755013€	e+08	2.718580e+	08	2.59092	20e+08		
221	162e+08 3.382899e	e+08	3.359420e+	08	3.2460	78e+08		
92	828e+08 1.417173e	e+09	1.396387e+	09	1.3228	67e+09		
41	614e+09 1.425887∈ 191e+09	e+09	1.424930e+	09	1.3937	15e+09		
	2000 Populat	ion 199	00 Populati	on 198	0 Popu	lation	1970	
	ation \ 1.468448∈	00	1.480057e+	00	12025	7420.0		
	3010.0	:+00	1.4000576+	00	13023	7420.0		
	1.291933e 860.0	e+08	1.071477e+	80	83929	9765.0		
	1.758737e	+08	1.507064e+	08	122288	8383.0		
	875.0	00	0 501406	0.7	7205	1420 0		
	1.228520∈ 264.0	9+08	9.521426e+	07	7295	1439.0		
156	1.543699e	e+08	1.154141e+	08	8062	4057.0		
	872.0 2.140724∈	+08	1.821599e+	08	14817	7096.0		
	8394.0	00	2 400027	00	222144	0010 0		
	2.823986e 8340.0	9+98	2.480837e+	08	223140	9018.0		
92	1.059634e	e+09	N	aN		NaN		
	1301.0 1.264099e	·+09	1.153704e+	09	982372	2466.0		
	4450.0							
	Area (km²)	Density	(per km²)	Growth	Rate	World	Population	
171	ntage 17098242.0		8.4636	Θ	.9973			
1.81								
16 2.15	147570.0		1160.0350	1	.0108			
27	8515767.0		25.2841	1	.0046			
2.70 149	923768.0		236.5759	1	.0241			
2.74								
156 2.96	881912.0		267.4018	1	.0191			
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
      9706961.0
                          146.8933
                                        1.0000
41
17.88
# for writing in different ways
df[df['Rank'] < 10].sort values(by = ['Country', 'Rank'] , ascending = ['Rank']  
False) # first sorting country
# then rank
    Rank CCA3
                                                     Continent \
                      Country
                                        Capital
221
        3
           USA United States Washington, D.C.
                                                North America
171
           RUS
                       Russia
                                        Moscow
                                                        Europe
156
        5
           PAK
                     Pakistan
                                      Islamabad
                                                          Asia
149
        6
           NGA
                      Nigeria
                                          Abuja
                                                        Africa
93
           IDN
        4
                    Indonesia
                                        Jakarta
                                                          Asia
92
        2 IND
                        India
                                      New Delhi
                                                          Asia
41
       1 CHN
                        China
                                        Beijing
                                                          Asia
27
                       Brazil
       7
           BRA
                                       Brasilia South America
16
        8
           BGD
                   Bangladesh
                                          Dhaka
    2022 Population 2020 Population 2015 Population 2010
Population \
                        3.359420e+08
221
        3.382899e+08
                                          3.246078e+08
3.111828e+08
        1.447133e+08
                        1.456173e+08
                                          1.446684e+08
171
1.432426e+08
                        2.271967e+08
156
        2.358249e+08
                                         2.109693e+08
1.944545e+08
        2.185412e+08
                        2.083274e+08
                                         1.839958e+08
1.609529e+08
                        2.718580e+08
93
        2.755013e+08
                                          2.590920e+08
2.440162e+08
                        1.396387e+09
                                          1.322867e+09
        1.417173e+09
1.240614e+09
        1.425887e+09
                        1.424930e+09
                                          1.393715e+09
41
1.348191e+09
                        2.131963e+08
                                          2.051882e+08
        2.153135e+08
1.963535e+08
        1.711864e+08
                         1.674210e+08
                                          1.578300e+08
16
1.483911e+08
    2000 Population 1990 Population 1980 Population 1970
Population \
221
        2.823986e+08
                        2.480837e+08
                                           223140018.0
200328340.0
        1.468448e+08
                        1.480057e+08
                                           138257420.0
130093010.0
       1.543699e+08
                        1.154141e+08
                                            80624057.0
156
59290872.0
149
        1.228520e+08
                        9.521426e+07
                                            72951439.0
```

```
55569264.0
        2.140724e+08
                          1.821599e+08
                                             148177096.0
93
115228394.0
92
        1.059634e+09
                                    NaN
                                                      NaN
557501301.0
        1.264099e+09
                          1.153704e+09
                                             982372466.0
822534450.0
27
        1.758737e+08
                          1.507064e+08
                                             122288383.0
96369875.0
        1.291933e+08
                          1.071477e+08
                                              83929765.0
67541860.0
                 Density (per km<sup>2</sup>) Growth Rate World Population
     Area (km²)
Percentage
221
      9372610.0
                            36.0935
                                           1.0038
4.24
                                           0.9973
171
     17098242.0
                             8.4636
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
                          1160.0350
16
       147570.0
                                           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
                                             Capital Continent \
     Rank CCA3
                          Country
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
                          Andorra Andorra la Vella
      203
           AND
                                                        Europe
      . . .
```

232	226 172 46 63 74	WLF ESH YEM ZMB ZWE	We	stern	Futuna Sahara Yemen Zambia imbabwe		Mata-Utu El Aaiún Sanaa Lusaka Harare			
Ponul	2022 Lation		ation	2020	Populatio	on	2015 Populati	.on	2010	
0	9672.0	41128	771.0		38972230.	. 0	33753499	0.0		
1 29133			321.0		2866849.	. 0	2882481	0		
2	5344.0		225.0		43451666.	. 0	39543154	1.0		
3 54849			273.0		46189.	. 0	51368	3.0		
4 71519		79	824.0		77700.	. 0	71746	6.0		
229 13142	2 0	11	.572.0		11655.	. 0	12182	2.0		
230 41329		575	986.0		556048.	. 0	491824	1.0		
231			614.0		32284046.	. 0	28516545	5.0		
232		20017	675.0		18927715.	. 0	N	laN		
233		16320	537.0		15669666.	. 0	14154937	7.0		
	2000 Lation		ation	1990	Populatio	on	1980 Populati	.on	1970	
0	2971.0	19542	982.0		10694796.	. 0	12486631	0		
1 23247			021.0		3295066.	. 0	2941651	0		
2	5915.0		621.0		25518074.	. 0	18739378	3.0		
3 27075			230.0		47818.	. 0	32886	6.0		
4 19860		66	097.0		53569.	. 0	35611	0		
	,10									
229 9377.	0	14	723.0		13454.	. 0	11315	5.0		
230 76371		270	375.0		178529.	. 0	116775	5.0		

231 1862870 6843607.0	9.0 133	375121.0	9204	1938.0	
232 9891130 4281671.0	6.0 76	586401.0	5720	9438.0	
233 1183467 5202918.0	6.0 101	113893.0	7049	9926.0	
Area (km²) Percentage	Density (per	km²) Growt	h Rate	World Popu	ılation
0 652230.0 0.52	63.	. 0587	1.0257		
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 0.00	170	. 5641	1.0100		
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 co	_				_
<pre>df = pd.read_csv('Country') df # changing index</pre>		_	on (1).d	csv" , inde	ex_col =
	Rank CCA3		npital Co	ontinent 2	2022
Population \ Country	name cens		ipitat co	one incline in	.022
Afghanistan	36 AFG		Kabul	Asia	
41128771.0 Albania	138 ALB	Т	irana	Europe	

Algiers

Pago Pago

Africa

Oceania

34 DZA

213 ASM

2842321.0 Algeria

44903225.0 American Samoa

44273.0

Andorra 79824.0	203	AND	Andor	ra la	Vella	Eu	rope		
Wallis and Futuna	226	WLF		Ma	ta-Utu	0ce	ania		
Western Sahara 575986.0	172	ESH		Εl	Aaiún	Af	rica		
Yemen 33696614.0	46	YEM			Sanaa	ı	Asia		
Zambia 20017675.0	63	ZMB		l	∟usaka	Af	rica		
Zimbabwe 16320537.0	74	ZWE		ŀ	Harare	Af	rica		
Population \ Country	2020	Popula	ation	2015	Populat	ion	2010		
Afghanistan		38972	230.0		3375349	99.0		28189672.6	9
Albania		2866	849.0		288248	31.0		2913399.0)
Algeria		43451	666.0		3954315	54.0		35856344.6	9
American Samoa		46	189.0		5136	68.0		54849.0)
Andorra		77	700.0		7174	16.0		71519.0)
Wallis and Futuna		11	655.0		1218	32.0		13142.0	9
Western Sahara		556	048.0		49182	24.0		413296.0	9
Yemen		32284	046.0		2851654	15.0		24743946.6)
Zambia		18927	715.0			NaN		13792086.0)
Zimbabwe		15669	666.0		1415493	37.0		12839771.6	9
	2000	Daniel	. 4	1000	Dan1 a.4		1000		
Population \ Country	2000	Popula	ation	1990	Populat	cion	1980		
Afghanistan		19542	982.0		1069479	96.0		12486631.6)
Albania		3182	021.0		329506	6.0		2941651.0)
Algeria		30774	621.0		2551807	74.0		18739378.0	9

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 Algeria	2324731.0 13795915.0	28748.0 2381741.0	98.8702 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 Wor	ld Population Percer	ntage
Country	1 0057		0.50
Afghanistan	1.0257		0.52
Albania Algeria	0.9957 1.0164		0.04 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 colu	mns]		
<pre>df.reset_index(inpl the same previous t</pre>		dex got reseted and	it happens in

```
#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
               Albania
1
                         138 ALB
                                             Tirana
                                                        Europe
2
               Algeria
                         34 DZA
                                            Algiers
                                                        Africa
3
        American Samoa
                                           Pago Pago
                         213 ASM
                                                       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 \
                           38972230.0
          41128771.0
                                            33753499.0
28189672.0
           2842321.0
                            2866849.0
1
                                             2882481.0
2913399.0
          44903225.0
                           43451666.0
                                             39543154.0
35856344.0
             44273.0
                              46189.0
                                                51368.0
54849.0
             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
```

		tion 19	990 Populatio	n 1980 Popu	lation	1970	
	ation \						
0	195429	82.0	10694796.	0 1248	6631.0		
	971.0						
1	31820	21.0	3295066.	0 294	1651.0		
23247							
2	307746	21.0	25518074.	0 1873	9378.0		
13795		20.0	47010		2006 0		
3		30.0	47818.	9 3.	2886.0		
27075		07.0	F2F60	0 0	F.C.1.1.0		
4		97.0	53569.	U 3.	5611.0		
19860	. 0						
· •				•			• •
229	1.47	23.0	13454.	0 1	1315.0		
9377.		23.0	13434.	0 1	1313.0		
230	2703	75 A	178529.	n 11	6775.0		
76371		73.0	170323.		0113.0		
231	186287	00 0	13375121.	920	4938.0		
68436		0010	133731211	520	133010		
232		36.0	7686401.	0 572	0438.0		
42816							
233		76.0	10113893.	0 704	9926.0		
52029	18.0						
	Area (km²)	Density	y (per km²)	Growth Rate	World H	Population	
Perce	ntage	Density			World A	Population	
Perce 0		Density	y (per km²) (63.0587	Growth Rate	World I	Population	
Perce 0 0.52	ntage 652230.0	Density	63.0587	1.0257	World I	Population	
Perce 0 0.52 1	ntage	Density			World A	Population	
Perce 0 0.52 1 0.04	ntage 652230.0 28748.0	Density	63.0587 98.8702	1.0257 0.9957	World H	Population	
Perce 0 0.52 1 0.04 2	ntage 652230.0	Density	63.0587	1.0257	World F	Population	
Perce 0 0.52 1 0.04 2 0.56	ntage 652230.0 28748.0 2381741.0	Density	63.0587 98.8702 18.8531	1.0257 0.9957 1.0164	World H	Population	
Perce 0 0.52 1 0.04 2 0.56 3	ntage 652230.0 28748.0	Density	63.0587 98.8702	1.0257 0.9957	World I	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00	28748.0 2381741.0	Density	63.0587 98.8702 18.8531 222.4774	1.0257 0.9957 1.0164 0.9831	World F	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00 4	ntage 652230.0 28748.0 2381741.0	Density	63.0587 98.8702 18.8531	1.0257 0.9957 1.0164	World F	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00 4 0.00	ntage 652230.0 28748.0 2381741.0 199.0 468.0	Density	63.0587 98.8702 18.8531 222.4774	1.0257 0.9957 1.0164 0.9831 1.0100	World I	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00 4 0.00	28748.0 2381741.0	Density	63.0587 98.8702 18.8531 222.4774	1.0257 0.9957 1.0164 0.9831	World	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00 4 0.00	ntage 652230.0 28748.0 2381741.0 199.0 468.0	Density	63.0587 98.8702 18.8531 222.4774	1.0257 0.9957 1.0164 0.9831 1.0100	World F	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00 4 0.00	199.0 468.0	Density	63.0587 98.8702 18.8531 222.4774 170.5641	1.0257 0.9957 1.0164 0.9831 1.0100	World F	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00 4 0.00 	199.0 468.0	Density	63.0587 98.8702 18.8531 222.4774 170.5641	1.0257 0.9957 1.0164 0.9831 1.0100	World I	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00 4 0.00 229 0.00	ntage 652230.0 28748.0 2381741.0 199.0 468.0 	Density	63.0587 98.8702 18.8531 222.4774 170.5641 81.4930	1.0257 0.9957 1.0164 0.9831 1.0100 	World F	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00 4 0.00 229 0.00 230	ntage 652230.0 28748.0 2381741.0 199.0 468.0 	Density	63.0587 98.8702 18.8531 222.4774 170.5641 81.4930	1.0257 0.9957 1.0164 0.9831 1.0100 	World F	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00 4 0.00 229 0.00 230 0.01 231 0.42	ntage 652230.0 28748.0 2381741.0 199.0 468.0 142.0 266000.0 527968.0	Density	63.0587 98.8702 18.8531 222.4774 170.5641 81.4930 2.1654 63.8232	1.0257 0.9957 1.0164 0.9831 1.0100 0.9953 1.0184 1.0217	World I	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00 4 0.00 229 0.00 230 0.01 231 0.42 232	142.0 2652230.0 28748.0 2381741.0 199.0 468.0	Density	63.0587 98.8702 18.8531 222.4774 170.5641 81.4930 2.1654	1.0257 0.9957 1.0164 0.9831 1.0100 0.9953 1.0184	World I	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00 4 0.00 230 0.01 231 0.42 232 0.25	ntage 652230.0 28748.0 2381741.0 199.0 468.0 142.0 266000.0 527968.0 752612.0	Density	63.0587 98.8702 18.8531 222.4774 170.5641 81.4930 2.1654 63.8232 26.5976	1.0257 0.9957 1.0164 0.9831 1.0100 0.9953 1.0184 1.0217 1.0280	World F	Population	
Perce 0 0.52 1 0.04 2 0.56 3 0.00 4 0.00 229 0.00 230 0.01 231 0.42 232	ntage 652230.0 28748.0 2381741.0 199.0 468.0 142.0 266000.0 527968.0	Density	63.0587 98.8702 18.8531 222.4774 170.5641 81.4930 2.1654 63.8232	1.0257 0.9957 1.0164 0.9831 1.0100 0.9953 1.0184 1.0217	World I	Population	

[234 rows x 17 columns]

in different way ,
df set index('Country') # It will momentail

df.set_index('Coun	try')	# It v	vill m	omentrily	/ cha	nge .	index	
	Rank	CCA3		Capit	tal C	onti	nent	2022
Population \ Country								
Afghanistan 41128771.0	36	AFG		Kal	oul	ı	Asia	
Albania 2842321.0	138	ALB		Tira	ana	Eu	rope	
Algeria 44903225.0	34	DZA		Algie	ers	Af	rica	
American Samoa 44273.0	213	ASM		Pago Pa	ago	0ce	ania	
Andorra 79824.0	203	AND	Andor	ra la Ve	lla	Eu	rope	
Wallis and Futuna	226	WLF		Mata-l	Jtu	0ce	ania	
Western Sahara 575986.0	172	ESH		El Aa:	iún	Af	rica	
Yemen 33696614.0	46	YEM		Sar	naa	1	Asia	
Zambia 20017675.0	63	ZMB		Lusa	aka	Af	rica	
Zimbabwe 16320537.0	74	ZWE		Hara	are	Af	rica	
Population \ Country	2020	Popula	ation	2015 Pop	oulat	ion	2010	
Afghanistan		389722	230.0	337	75349	9.0		28189672.0
Albania		28668	349.0	28	38248	1.0		2913399.0
Algeria		434516	666.0	395	54315	4.0		35856344.0
American Samoa		461	L89.0		5136	8.0		54849.0
Andorra		777	700.0		7174	6.0		71519.0
Wallis and Futuna		116	555.0		1218	2.0		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 D 1 1	1000 5 1	1000
Population \ Country	2000 Population	1990 Population	1980
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
Country	1970 Population	Area (km²) Dens	ity (per km²) \
Country	10752971.0	652230.0	63.0587
Afghanistan Albania	2324731.0	28748.0	98.8702
Algeria	13795915.0	2381741.0	18.8531
Andrican Samoa	27075.0	199.0	222.4774
Andorra	19860.0	468.0	170.5641
Wallic and Eutuna	9377.0	142.0	01 4020
Wallis and Futuna		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 Wor	ld Population Per	centage

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
Yemen Zambia	1.0217 1.0280	0.42 0.25

[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 2842321.0	138	ALB	Tirana	Europe	
Algeria 44903225.0	34	DZA	Algiers	Africa	
American Samoa	213	ASM	Pago Pago	Oceania	
Andorra 79824.0	203	AND	Andorra la Vella	Europe	
Wallis and Futuna	226	WLF	Mata-Utu	0ceania	
Western Sahara 575986.0	172	ESH	El Aaiún	Africa	
Yemen	46	YEM	Sanaa	Asia	
33696614.0 Zambia	63	ZMB	Lusaka	Africa	
20017675.0 Zimbabwe 16320537.0	74	ZWE	Harare	Africa	
	2020	Popul	ation 2015 Popula	ation 2010	
Population \ Country		- 12 - 21			

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 Parulation	1000 Danielatian	1000
Population \ Country	2000 Population	1990 Population	1980
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
Country	1970 Population	Area (km²) Dens	ity (per km²) \

Afghanistan Albania Algeria American Samoa Andorra Wallis and Futuna Western Sahara Yemen Zambia	10752971.0 2324731.0 13795915.0 27075.0 19860.0 9377.0 76371.0 6843607.0 4281671.0	652230.0 28748.0 2381741.0 199.0 468.0 142.0 266000.0 527968.0 752612.0	63.0587 98.8702 18.8531 222.4774 170.5641 81.4930 2.1654 63.8232 26.5976
Zimbabwe	5202918.0	390757.0	41.7665
Grow	th Rate World	d Population Pe	rcentage
Country Afghanistan Albania Algeria American Samoa Andorra Wallis and Futuna Western Sahara Yemen Zambia Zimbabwe	1.0257 0.9957 1.0164 0.9831 1.0100 0.9953 1.0184 1.0217 1.0280 1.0204		0.52 0.04 0.56 0.00 0.00 0.00 0.01 0.42 0.25 0.20
[234 rows x 16 columns]			
<pre># Searching based of the location) df.loc['Albania']</pre>	e index : loc	and iloc (loca	tion and integer
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 Percent	Eur 284232 286684 288248 291339 318202 329506 294169 232473 2874 98.8 0.9	49.0 81.0 99.0 21.0 66.0 51.0	

```
# 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 km<sup>2</sup>)
                                   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
                           36 AFG
           Afghanistan
                                                 Kabul
                                                            Asia
                          138
1
                Albania
                               ALB
                                               Tirana
                                                          Europe
2
                Algeria
                           34
                               DZA
                                              Algiers
                                                          Africa
3
        American Samoa
                          213
                               ASM
                                            Pago Pago
                                                         Oceania
4
                                     Andorra la Vella
                Andorra
                          203
                               AND
                                                          Europe
                          . . .
     Wallis and Futuna
229
                          226
                               WLF
                                             Mata-Utu
                                                         Oceania
230
        Western Sahara
                          172
                               ESH
                                              El Aaiún
                                                          Africa
231
                  Yemen
                           46
                               YEM
                                                 Sanaa
                                                            Asia
                               ZMB
232
                 Zambia
                           63
                                               Lusaka
                                                          Africa
233
                               ZWE
              Zimbabwe
                           74
                                               Harare
                                                          Africa
     2022 Population 2020 Population 2015 Population 2010
Population \
          41128771.0
                            38972230.0
                                              33753499.0
28189672.0
           2842321.0
                             2866849.0
                                               2882481.0
1
2913399.0
          44903225.0
                            43451666.0
                                              39543154.0
35856344.0
              44273.0
                               46189.0
                                                  51368.0
54849.0
4
             79824.0
                                77700.0
                                                  71746.0
```

71519.0								
229 13142.0	115	72.0	116	555.0	1	2182.0		
230 413296.0	5759	86.0	5566	048.0	49	1824.0		
	336966	14.0	322840)46.0	2851	6545.0		
	200176	75.0	189277	15.0		NaN		
233 12839771.0	163205	37.0	156696	66.0	1415	4937.0		
2000 Population	Popula	tion 1	1990 Popula	ation	1980 Popu	lation	1970	
•	195429	82.0	106947	96.0	1248	6631.0		
1 2324731.0	31820	21.0	32950	066.0	294	1651.0		
	307746	21.0	255186	74.0	1873	9378.0		
3 27075.0	582	30.0	478	318.0	3	2886.0		
4 19860.0	660	97.0	535	69.0	3	5611.0		
229 9377.0	147	23.0	134	154.0	1	1315.0		
230 76371.0	2703	75.0	1785	529.0	11	6775.0		
	186287	00.0	133751	21.0	920	4938.0		
232 4281671.0	98911	36.0	76864	101.0	572	0438.0		
	118346	76.0	101138	393.0	704	9926.0		
Area Percentage		Densit	y (per km²) Gro	owth Rate	World	Population	
	230.0		63.058	37	1.0257			
	748.0		98.876)2	0.9957			
	741.0		18.853	31	1.0164			
	199.0		222.477	' 4	0.9831			

4	468.0	170.5641	<u>L</u>	1.0100	
0.00					
229 0.00	142.0	81.4930)	0.9953	
230	266000.0	2.1654	ļ	1.0184	
0.01 231	527968.0	63.8232)	1.0217	
0.42					
232 0.25	752612.0	26.5976	j	1.0280	
233 0.20	390757.0	41.7665	5	1.0204	
[234 ro	ws x 17 columns]				
df.set	index(['Continer	nt' , 'Count	rv'l	, inplace = True)
df	, , , , , , , , , , , , , , , , , , , ,	, 553	, .	, , , , , , , , , , , , , , , , , , , ,	
u i		.	6645		
Populat	ion \	Rank	CCA3	Capita	l 2022
	nt Country				
Asia	Afghanistan	36	AFG	Kabu ⁻	U
4112877	1.0				
Europe 2842321	Albania .0	138	ALB	Tirana	9
Africa	Algeria	34	DZA	Algiers	5
4490322 Oceania		oa 213	ASM	Pago Pago	0
44273.0					
Europe 79824.0	Andorra	203	AND	Andorra la Vella	a
 Oceania	Wallis and Fu	utuna 226	WLF	Mata-Uti	J
11572.0					
Africa 575986.	Western Sahai	ra 172	ESH	El Aaiún	1
Asia	Yemen	46	YEM	Sanaa	a
3369661		63	71/10		
Africa 2001767	Zambia 5.0	63	ZMB	Lusaka	3
	Zimbabwe	74	ZWE	Harare	2
1632053	7.0				
		2020	Popul	ation 2015 Popu	lation \
Contine	nt Country		·	•	

Asia Europe Africa Oceania Europe Oceania Africa Asia Africa	Afghanistan Albania Algeria American Samoa Andorra Wallis and Futuna Western Sahara Yemen Zambia Zimbabwe		38972230.0 2866849.0 43451666.0 46189.0 77700.0 11655.0 556048.0 32284046.0 18927715.0 15669666.0		3375349 288248 3954315 5136 7174 1218 49182 2851654	31.0 54.0 58.0 46.0 32.0 24.0 15.0 NaN		
Continent Asia Europe Africa Oceania Europe	Afghanistan	2010	Population 28189672.0 2913399.0 35856344.0 54849.0 71519.0	2000	Populat 1954298 318202 3077462 5823 6609	32.0 21.0 21.0 30.0	\	
Oceania Africa Asia Africa	Wallis and Futuna Western Sahara Yemen Zambia Zimbabwe		13142.0 413296.0 24743946.0 13792086.0 12839771.0		1472 27037 1862876 989113 1183467	75.0 00.0 86.0		
		1990	Population	1980	Populat	ion	\	
Continent Asia Europe Africa Oceania Europe	Country Afghanistan Albania Algeria American Samoa Andorra		10694796.0 3295066.0 25518074.0 47818.0 53569.0		1248663 294165 1873937 3288 3561	51.0 78.0 86.0		
Oceania Africa Asia Africa	Wallis and Futuna Western Sahara Yemen Zambia Zimbabwe		13454.0 178529.0 13375121.0 7686401.0 10113893.0		1131 11677 920493 572043 704992	75.0 88.0 88.0		
km²) \		1970	Population	Area	(km²)	Dens	ity	(per
Continent	Country							
Asia 63.0587	Afghanistan		10752971.0	652	2230.0			
Europe 98.8702	Albania		2324731.0	28	3748.0			
Africa 18.8531	Algeria		13795915.0	2383	1741.0			

Oceania 222.4774	American Samoa	2707	5.0	199.0	
Europe	Andorra	1986	0.0	468.0	
170.5641					
 Oceania	Wallis and Futuna	937	7 0	142.0	
81.4930					
Africa 2.1654	Western Sahara	7637	1.0	266000.0	
Asia	Yemen	684360	7.0	527968.0	
63.8232 Africa	Zambia	428167	1.0	752612.0	
26.5976			-		
41.7665	Zimbabwe	520291	8.0	390757.0	
		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_i	ndex(ascending = Tr	rue)			
		Rank CCA3	Capit	al 2022 P	opulation \
Continent Africa	Country Algeria	34 DZA	Algie	rs 4	4903225.0

 South America	Angola Benin Botswana Burkina Faso Paraguay Peru Suriname Uruguay Venezuela	42 77 144 58 109 44 170 133 51	AGO BEN BWA BFA PRY PER SUR URY VEN	Port Ga Ouaga As Para Mont	Luanda o-Novo borone dougou unción Lima maribo evideo aracas	1335 263 2267 678 3404 61 342	8987.0 2864.0 0296.0 3762.0 0744.0 9588.0 8040.0 2794.0 1696.0
Population \ Continent	Country	2020	Popul	ation	2015 F	Population	2010
Africa	Algeria		43451	666.0	3	39543154.0	
35856344.0	Angola		33/128	485.0	7	28127721.0	
23364185.0	Aligota		33420	403.0	2	2012//21.0	
0445710 0	Benin		12643	123.0	1	10932783.0	
9445710.0	Botswana		2546	402.0		2305171.0	
2091664.0							
16116845.0	Burkina Faso		21522	626.0]	18718019.0	
				• • •			
South America 5768613.0				695.0		6177950.0	
29229572.0	Peru		33304	756.0	3	30711863.0	
	Suriname		607	065.0		575475.0	
546080.0	Uruguay		3/120	086.0		3402818.0	
3352651.0			3723	000.0		3402010.0	
28715022.0	Venezuela		28490	453.0	3	30529716.0	
20/15022.0							
Danulatian)		2000	Popul	ation	1990 F	Population	1980
Population \ Continent	Country						
Africa 18739378.0	Algeria			621.0		25518074.0	
8330047.0	Angola		16394	062.0	1	11828638.0	
0330047.0	Benin		6998	023.0		5133419.0	
3833939.0	Potevens		1720	00F 0		12/11/74 0	
938578.0	Botswana		1/20	985.0		1341474.0	

6932967.0	Burkina Faso	11882888.0	9131361	.0
South America 3078912.0	Paraguay	5123819.0	4059195	.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	Θ
15210443.0	Venezueta	Naiv	13730373	
		1970 Population	Area (km²) D	ensity (per
km²) \ Continent	Country			
Africa 18.8531	Algeria	13795915.0	2381741.0	
	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	Dui Killa 1 a50	3011000.0	272907.0	
South America 16.6705	Paraguay	2408787.0	406752.0	
26.4933	Peru	13562371.0	1285216.0	
	Suriname	379918.0	163820.0	
3.7727	Uruguay	2790265.0	181034.0	
18.9069				
30.8820	Venezuela	11355475.0	NaN	
		Growth Rate Wor	ld Population	Percentage
Continent	Country			_
Africa	Algeria Angola	1.0164 1.0315		0.56 0.45
	Benin	1.0274		0.17
	Botswana Burkina Faso	1.0162 1.0259		0.03 0.28

```
South America Paraguay
                                                                    0.09
                                   1.0115
                                                                    0.43
               Peru
                                   1.0099
              Suriname
                                   1.0082
                                                                    0.01
                                   0.9990
                                                                    0.04
              Uruquav
              Venezuela
                                   1.0036
                                                                    0.35
[234 rows x 15 columns]
# df.loc['Angola'] # not gonna work properly
                  # It's searching in first index / string which is
Continent
df.loc['Africa' , 'Angola']
                                         42
Rank
CCA3
                                        AG0
Capital
                                     Luanda
2022 Population
                                 35588987.0
2020 Population
                                 33428485.0
2015 Population
                                 28127721.0
                                 23364185.0
2010 Population
2000 Population
                                16394062.0
1990 Population
                                11828638.0
1980 Population
                                 8330047.0
1970 Population
                                6029700.0
Area (km<sup>2</sup>)
                                 1246700.0
Density (per km<sup>2</sup>)
                                    28.5466
Growth Rate
                                     1.0315
World Population Percentage
                                       0.45
Name: (Africa, Angola), dtype: object
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
                                2941651.0
1980 Population
1970 Population
                                2324731.0
Area (km<sup>2</sup>)
                                   28748.0
Density (per km<sup>2</sup>)
                                   98.8702
Growth Rate
                                    0.9957
World Population Percentage
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 \
      Mint Chocolate Chip
                               Vanilla
                                                        10.0
                                          Yes
8.0
                             Chocolate
1
                Chocolate
                                          Yes
                                                         8.8
7.6
                                                         4.7
2
                  Vanilla
                               Vanilla
                                           No
5.0
3
             Cookie Dough
                               Vanilla
                                         Yes
                                                         6.9
6.5
4
               Rocky Road
                             Chocolate
                                          Yes
                                                         8.2
7.0
                Pistachio
                               Vanilla
                                                         2.3
5
                                           No
3.4
              Cake Batter
                               Vanilla
                                          Yes
                                                         6.5
6
6.0
7
               Neapolitan
                               Vanilla
                                           No
                                                         3.8
5.0
8 Chocolte Fudge Brownie
                             Chocolate
                                                         8.2
                                         Yes
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
                                   7.233333
                                                     15.70
                        8.4
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
                                                       3
Chocolate
                        3
                                       3
3
Vanilla
                        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
                                                   47.1
                     25.2
                                     21.7
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
                            8.8
                                     3 25.2
                      8.4
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
                            max count
                                                       mean max count
                     mean
                                         sum
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 \
     Mint Chocolate Chip
                                                       10.0
                              Vanilla
                                        Yes
8.0
                Chocolate
                            Chocolate
                                                        8.8
1
                                         Yes
7.6
                  Vanilla
                              Vanilla
                                                        4.7
2
                                          No
5.0
                              Vanilla
                                                        6.9
3
             Cookie Dough
                                         Yes
6.5
4
               Rocky Road
                            Chocolate
                                        Yes
                                                        8.2
7.0
                                                        2.3
5
                Pistachio
                              Vanilla
                                          No
3.4
              Cake Batter
                              Vanilla
                                                        6.5
                                        Yes
6
6.0
7
                              Vanilla
                                                        3.8
               Neapolitan
                                          No
5.0
8 Chocolte Fudge Brownie
                            Chocolate
                                                        8.2
                                       Yes
7.1
   Total Rating
```

```
0
           18.0
           16.6
1
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
                             8.4
                                        7.233333
                                                     15.700000
            Yes
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
                                                25%
                                                     50% 75%
                    count mean
                                    std min
                                                                max
Base Flavor
                      3.0 8.4 0.346410 8.2 8.200 8.2 8.5
Chocolate
                                                                8.8
Vanilla
                      6.0 5.7 2.710719 2.3 4.025 5.6 6.8 10.0
            Texture Rating
                                                     Total Rating
                                            75% max
                     count
                               mean ...
                                                            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
                               25%
                                     50%
                                             75%
                  std
                       min
                                                   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
```

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
                    Frodo
                              50
           1001
1
           1002
                  Samwise
                              39
2
           1006
                  Legolas
                          2931
3
                   Elrond 6520
           1007
           1008
                 Barromir
# merging 1st dataframe to 2nd
df1.merge(df2)
   FellowshipID FirstName
                               Skills
                                       Age
0
                                        50
           1001
                    Frodo
                               Hidina
1
           1002
                  Samwise Gardening
# we did'nt explicitly ssays join using fellowshipID
df1.merge(df2,how = 'inner') # inner is default
   FellowshipID FirstName
                               Skills
                                       Age
0
                                        50
           1001
                    Frodo
                               Hidina
1
           1002
                  Samwise Gardening
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
                                 Hiding
0
           1001
                      Frodo
                                              Frodo
                                                      50
           1002
                                            Samwise
1
                    Samwise Gardening
                                                      39
df1.merge(df2 , how = 'inner' , on = ['FellowshipID','FirstName'])
```

```
FellowshipID FirstName
                                Skills
                                        Age
0
                     Frodo
                                Hiding
                                         50
           1001
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
                                        2931.0
           1006
                   Legolas
                                   NaN
5
           1007
                    Elrond
                                   NaN
                                        6520.0
6
           1008
                  Barromir
                                   NaN
                                          51.0
# their is also a cross join
df1.merge(df2 , how = 'cross')
                                              FellowshipID y FirstName y
    FellowshipID x FirstName x
                                     Skills
Age
               1001
                          Frodo
                                     Hiding
                                                                    Frodo
0
                                                        1001
50
               1001
                          Frodo
                                     Hiding
                                                        1002
                                                                  Samwise
1
39
2
               1001
                                                        1006
                          Frodo
                                     Hiding
                                                                  Legolas
2931
               1001
                          Frodo
                                     Hiding
                                                        1007
                                                                   Elrond
6520
               1001
                                                                 Barromir
                          Frodo
                                     Hiding
                                                        1008
51
               1002
                        Samwise
                                  Gardening
                                                        1001
                                                                    Frodo
5
50
               1002
                        Samwise
                                  Gardening
                                                        1002
                                                                  Samwise
6
39
7
               1002
                        Samwise
                                  Gardening
                                                        1006
                                                                  Legolas
2931
               1002
                        Samwise
                                  Gardening
                                                        1007
                                                                   Elrond
8
6520
               1002
                        Samwise
                                  Gardening
                                                        1008
                                                                 Barromir
51
               1003
                        Gandalf
                                     Spells
                                                        1001
                                                                    Frodo
10
50
11
               1003
                        Gandalf
                                                        1002
                                     Spells
                                                                  Samwise
39
12
               1003
                        Gandalf
                                     Spells
                                                        1006
                                                                  Legolas
2931
13
               1003
                        Gandalf
                                     Spells
                                                        1007
                                                                   Elrond
6520
14
               1003
                        Gandalf
                                     Spells
                                                        1008
                                                                 Barromir
51
```

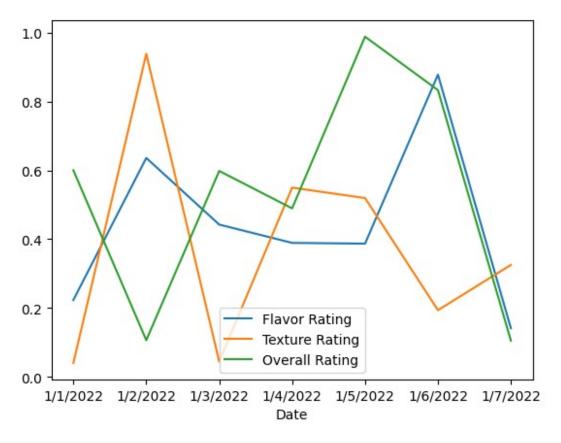
```
15
              1004
                         Pippin Fireworks
                                                       1001
                                                                   Frodo
50
16
              1004
                         Pippin Fireworks
                                                       1002
                                                                Samwise
39
17
              1004
                         Pippin Fireworks
                                                       1006
                                                                 Legolas
2931
                                                                 Elrond
18
              1004
                         Pippin Fireworks
                                                       1007
6520
19
              1004
                         Pippin Fireworks
                                                       1008
                                                               Barromir
51
# let see join function
# merge works better with columns
# joins works better with indexes
# dfl.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
                3
NaN
                                  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
                 1008.0
NaN
                                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
                                Hiding
            1001
                     Frodo
                                            NaN
1
            1002
                   Samwise
                             Gardening
                                            NaN
2
            1003
                   Gandalf
                                Spells
                                            NaN
3
            1004
                    Pippin
                             Fireworks
                                            NaN
0
                     Frodo
                                   NaN
                                           50.0
            1001
1
            1002
                                           39.0
                   Samwise
                                   NaN
2
            1006
                   Legolas
                                   NaN
                                         2931.0
3
                    Elrond
            1007
                                   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
                   Gandalf
            1003
3
            1004
                    Pippin
0
            1001
                     Frodo
1
            1002
                   Samwise
2
            1006
                   Legolas
3
                    Elrond
            1007
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
```

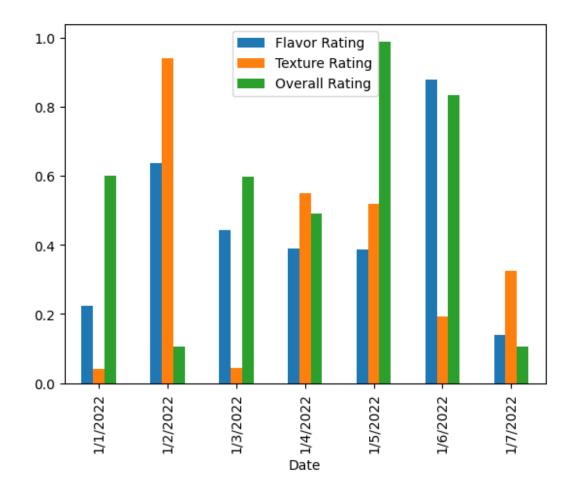
Pandas Visualization

```
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
          Flavor Rating Texture Rating
                                          Overall Rating
Date
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
               0.389128
                                0.549676
1/4/2022
                                                 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()
```



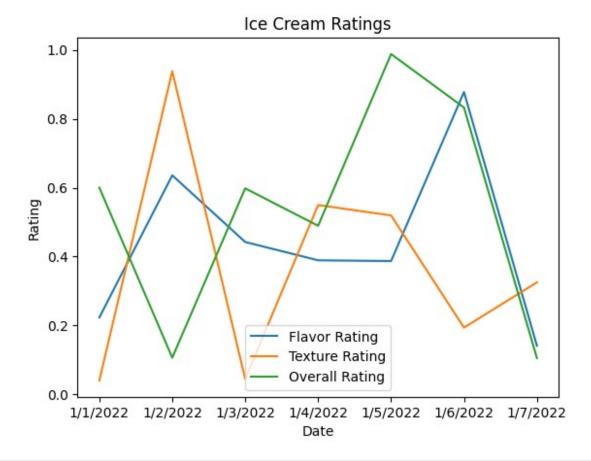
```
# we can use kind to change type of graph
df.plot(kind = 'bar');
```



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



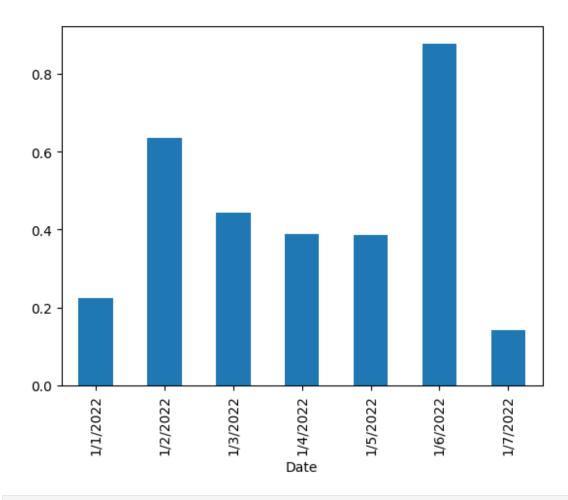
```
# can add title and labels
df.plot(kind = 'line' , title = 'Ice Cream Ratings' ,xlabel = 'Date' ,
ylabel = 'Rating');
```



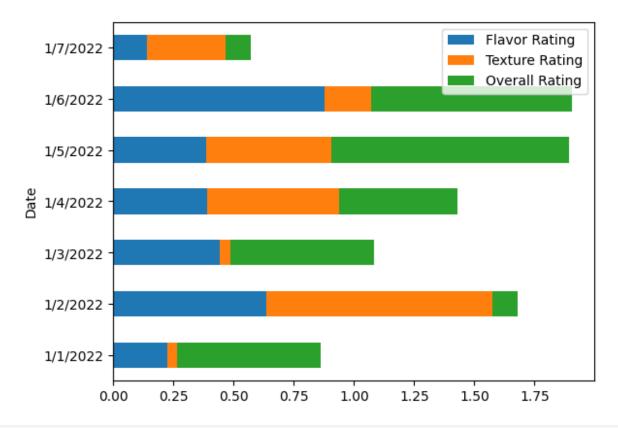
```
# for a stacked barchart
df.plot(kind = 'bar' , stacked = True);
```



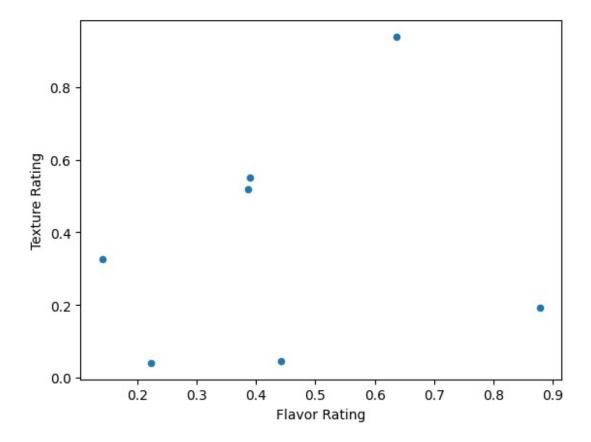
```
# for specifying column
df['Flavor Rating'].plot(kind = 'bar' , stacked = True);
```



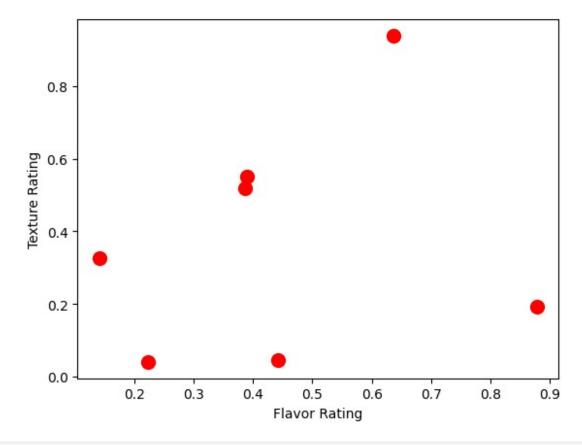
for changing into horzontal barchart
df.plot.barh(stacked = True);



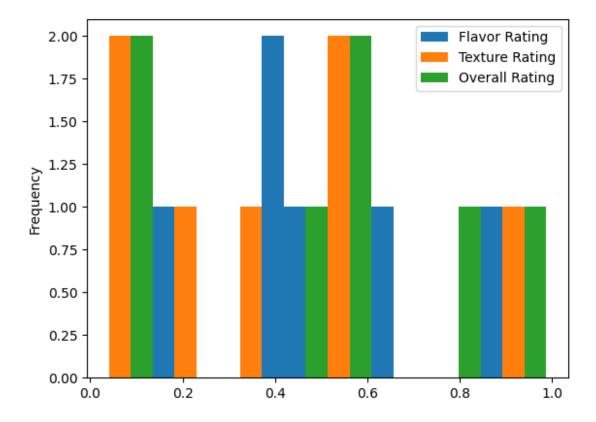
```
# for scatter plot
df.plot.scatter(x = 'Flavor Rating' , y = 'Texture Rating');
```



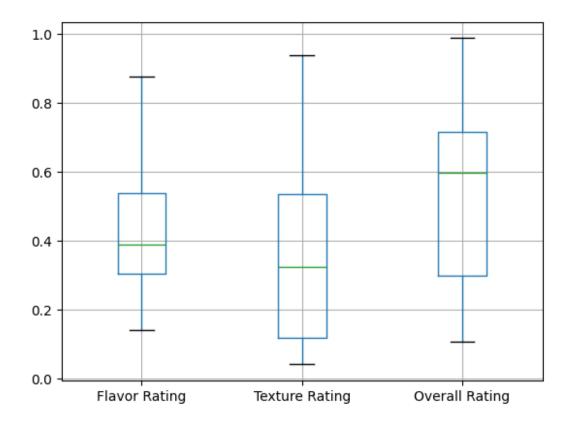
```
# can also change size of dots and color of dots df.plot.scatter(x = 'Flavor Rating' , y = 'Texture Rating' , s = 100 , c = 'Red');
```



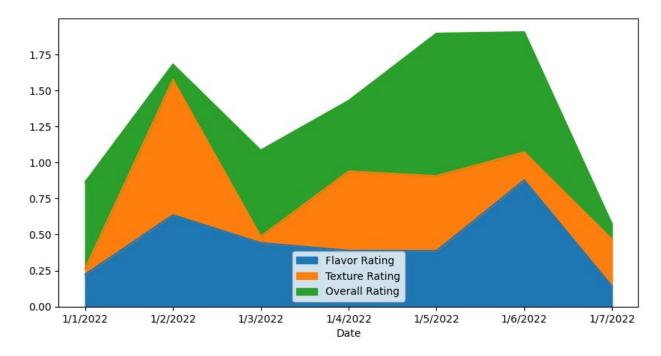
```
# histograms and bins
df.plot.hist(bins = 20);
```



boxplot
df.boxplot() # plot function not needed
<Axes: >



area chart
df.plot.area(figsize = (10,5)) # can change size using figsize
<Axes: xlabel='Date'>



```
# 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'>
```

