Colab: https://colab.research.google.com/drive/15Gh2h7I2erNPqfXd7e-Ok8pldEG9kf0N?usp=sharing

```
import pandas as pd
```

```
import numpy as np
```

```
# GDP per capita V/S Life Expectancy
```

```
!wget "https://drive.google.com/uc?export=download&id=1E3bwvYGf1ig32RmcYiWc0IXPN-mD
```

```
·2022-11-30 15:52:34--  https://drive.google.com/uc?export=download&id=1E3bwvYG
:solving drive.google.com (drive.google.com)... 142.251.2.102, 142.251.2.101, 1
)nnecting to drive.google.com (drive.google.com)|142.251.2.102|:443... connecte
'TP request sent, awaiting response... 303 See Other
)cation: https://doc-0s-68-docs.googleusercontent.com/docs/securesc/ha0ro937gcu
ırning: wildcards not supported in HTTP.
·2022-11-30 15:52:35--  https://doc-0s-68-docs.googleusercontent.com/docs/secur
:solving doc-0s-68-docs.googleusercontent.com (doc-0s-68-docs.googleusercontent
)nnecting to doc-0s-68-docs.googleusercontent.com (doc-0s-68-docs.googleusercon
'TP request sent, awaiting response... 200 OK
:ngth: 83785 (82K) [text/csv]
ıving to: ‘gapminder.csv’

==========>]  81.82K  --.-KB/s    in 0.001s

- ‘gapminder.csv’ saved [83785/83785]
```

Saving...

```
df = pd.read_csv("gapminder.csv")
df # structured data, tabular data
```

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314

```
type(df)
```

```
pandas.core.frame.DataFrame
```

3	Afghanistan	1967	11537966	Asia	34.020	836.197138

```
df.shape
```

```
(1704, 6)
```

```
df["country"]
```

```
0       Afghanistan
1       Afghanistan
2       Afghanistan
3       Afghanistan
4       Afghanistan
           ...
1699       Zimbabwe
1700       Zimbabwe
1701       Zimbabwe
1702       Zimbabwe
1703       Zimbabwe
Name: country, Length: 1704, dtype: object
```

Saving...

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1704 entries, 0 to 1703
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   country     1704 non-null   object 
 1   year        1704 non-null   int64  
 2   population  1704 non-null   int64  
 3   continent   1704 non-null   object 
 4   life_exp    1704 non-null   float64
 5   gdp_cap     1704 non-null   float64
dtypes: float64(2), int64(2), object(2)
memory usage: 80.0+ KB
```

```
df.head(7)
```

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
2	Afghanistan	1962	10267083	Asia	31.997	853.100710
3	Afghanistan	1967	11537966	Asia	34.020	836.197138

```
df.tail(6)
```

	country	year	population	continent	life_exp	gdp_cap
1698	Zimbabwe	1982	7636524	Africa	60.363	788.855041
1699	Zimbabwe	1987	9216418	Africa	62.351	706.157306
1700	Zimbabwe	1992	10704340	Africa	60.377	693.420786
1701	Zimbabwe	1997	11404948	Africa	46.809	792.449960
1702	Zimbabwe	2002	11926563	Africa	39.989	672.038623
1703	Zimbabwe	2007	12311143	Africa	43.487	469.709298

```
# row orineted approach - lists of lists

pd.DataFrame([['Afghanistan',1952, 8425333, 'Asia', 28.801, 779.445314 ],
                                , 9240934, 'Asia', 30.332, 820.853030 ],
                                , 102267083, 'Asia', 31.997, 853.100710 ]],
                          ,'year','population','continent','life_exp','gdp_c
```


	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
2	Afghanistan	1962	102267083	Asia	31.997	853.100710

```
pd.DataFrame([['Afghanistan',1952, 8425333, 'Asia', 28.801, 779.445314 ]],
             columns = ['country','year','population','continent','life_exp','gdp_c
```

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314

```
# column oriented approach

pd.DataFrame({'country':['Afghanistan', 'Afghanistan'], 'year':[1952,1957],
              'population':[842533, 9240934], 'continent':['Asia', 'Asia'],
              'life_exp':[28.801, 30.332], 'gdp_cap':[779.445314, 820.853030]})
```

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	842533	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030

```
# learner's doubt - columns with same names, gets replaced
pd.DataFrame({'country':['Afghanistan', 'Afghanistan'], 'year':[1952,1957],
              'population':[842533, 9240934], 'continent':['Asia', 'Asia'],
              'life_exp':[28.801, 30.332], 'continent':[779.445314, 820.853030]})
```

	country	year	population	continent	life_exp
0	Afghanistan	1952	842533	779.445314	28.801
1	Afghanistan	1957	9240934	820.853030	30.332

```
df.columns
```

```
Index(['country', 'year', 'population', 'continent', 'life_exp', 'gdp_cap'],
dtype='object')
```

```
df.keys() # df works like a "specialised" dictionary
```

```
Index(['country', 'year', 'population', 'continent', 'life_exp', 'gdp_cap'],
dtype='object')
```

Saving...

```
ract subset of columns
```

	country	life_exp
0	Afghanistan	28.801
1	Afghanistan	30.332
2	Afghanistan	31.997
3	Afghanistan	34.020
4	Afghanistan	36.088
...	...	...
1699	Zimbabwe	62.351
1700	Zimbabwe	60.377
1701	Zimbabwe	46.809
1702	Zimbabwe	39.989
1703	Zimbabwe	43.487

1704 rows × 2 columns

```
df["country"] # series
```

```
0       Afghanistan
1       Afghanistan
2       Afghanistan
3       Afghanistan
4       Afghanistan
           ...
1699       Zimbabwe
1700       Zimbabwe
1701       Zimbabwe
1702       Zimbabwe
1703       Zimbabwe
Name: country, Length: 1704, dtype: object
```

```
df[["country"]] # returns as dataframe because of double brackers
```

	country
0	Afghanistan
1	Afghanistan
2	Afghanistan
3	Afghanistan
4	Afghanistan
...	...
1701	Zimbabwe
1702	Zimbabwe
1703	Zimbabwe

1704 rows × 1 columns


```
df["country"].unique() #np.unique(df["country"])
```

```
array(['Afghanistan', 'Albania', 'Algeria', 'Angola', 'Argentina',
       'Australia', 'Austria', 'Bahrain', 'Bangladesh', 'Belgium',
       'Benin', 'Bolivia', 'Bosnia and Herzegovina', 'Botswana', 'Brazil',
       'Bulgaria', 'Burkina Faso', 'Burundi', 'Cambodia', 'Cameroon',
       'Canada', 'Central African Republic', 'Chad', 'Chile', 'China',
       'Colombia', 'Comoros', 'Congo, Dem. Rep.', 'Congo, Rep.',
       'Costa Rica', "Cote d'Ivoire", 'Croatia', 'Cuba', 'Czech Republic',
       'Denmark', 'Djibouti', 'Dominican Republic', 'Ecuador', 'Egypt',
       'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Ethiopia',
       'Finland', 'France', 'Gabon', 'Gambia', 'Germany', 'Ghana',
       'Greece', 'Guatemala', 'Guinea', 'Guinea-Bissau', 'Haiti',
       'Honduras', 'Hong Kong, China', 'Hungary', 'Iceland', 'India',
       'Indonesia', 'Iran', 'Iraq', 'Ireland', 'Israel', 'Italy',
       'Jamaica', 'Japan', 'Jordan', 'Kenya', 'Korea, Dem. Rep.',
       'Korea, Rep.', 'Kuwait', 'Lebanon', 'Lesotho', 'Liberia', 'Libya',
```

```
       'Madagascar', 'Malawi', 'Malaysia', 'Mali', 'Mauritania',
       'Mauritius', 'Mexico', 'Mongolia', 'Montenegro', 'Morocco',
       'Mozambique', 'Myanmar', 'Namibia', 'Nepal', 'Netherlands',
       'New Zealand', 'Nicaragua', 'Niger', 'Nigeria', 'Norway', 'Oman',
       'Pakistan', 'Panama', 'Paraguay', 'Peru', 'Philippines', 'Poland',
       'Portugal', 'Puerto Rico', 'Reunion', 'Romania', 'Rwanda',
       'Sao Tome and Principe', 'Saudi Arabia', 'Senegal', 'Serbia',
       'Sierra Leone', 'Singapore', 'Slovak Republic', 'Slovenia',
       'Somalia', 'South Africa', 'Spain', 'Sri Lanka', 'Sudan',
       'Swaziland', 'Sweden', 'Switzerland', 'Syria', 'Taiwan',
       'Tanzania', 'Thailand', 'Togo', 'Trinidad and Tobago', 'Tunisia',
       'Turkey', 'Uganda', 'United Kingdom', 'United States', 'Uruguay',
       'Venezuela', 'Vietnam', 'West Bank and Gaza', 'Yemen, Rep.',
       'Zambia', 'Zimbabwe'], dtype=object)
```

```
df["country"].nunique()
```

```
142
```

```
df["country"].value_counts()
```

```
Afghanistan          12
Pakistan             12
New Zealand          12
Nicaragua            12
Niger                12
                     ..
Eritrea              12
                     ..
Zimbabwe             12
Name: country, Length: 142, dtype: int64
```



```
df.rename({"population":"Population", "country":"Country"}, axis=1, inplace=True)
```

```
df
```

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030

```
df
```

	Country	year	Population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
2	Afghanistan	1962	10267083	Asia	31.997	853.100710
3	Afghanistan	1967	11537966	Asia	34.020	836.197138
4	Afghanistan	1972	13079460	Asia	36.088	739.981106
...	...	...	...	...	...	...
1699	Zimbabwe	1987	9216418	Africa	62.351	706.157306
1700	Zimbabwe	1992	10704340	Africa	60.377	693.420786
1701	Zimbabwe	1997	11404948	Africa	46.809	792.449960
1702	Zimbabwe	2002	11926563	Africa	39.989	672.038623
1703	Zimbabwe	2007	12311143	Africa	43.487	469.709298


```
df.rename({"continent":"Continent"}) # default axis=0
```

	Country	year	Population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
2	Afghanistan	1962	10267083	Asia	31.997	853.100710
3	Afghanistan	1967	11537966	Asia	34.020	836.197138
4	Afghanistan	1972	13079460	Asia	36.088	739.981106
...	...	...	...	...	...	...
1699	Zimbabwe	1987	9216418	Africa	62.351	706.157306
1700	Zimbabwe	1992	10704340	Africa	60.377	693.420786
1701	Zimbabwe	1997	11404948	Africa	46.809	792.449960
1702	Zimbabwe	2002	11926563	Africa	39.989	672.038623
1703	Zimbabwe	2007	12311143	Africa	43.487	469.709298

1704 rows × 6 columns

```
df["Country"] # dict like style
```

```
0       Afghanistan
1       Afghanistan
2       Afghanistan
3       Afghanistan
4       Afghanistan
           ...
1699       Zimbabwe
1700       Zimbabwe
1701       Zimbabwe
1702       Zimbabwe
1703       Zimbabwe
Name: Country, Length: 1704, dtype: object
```

```
df.Country # attribute, DONT USE THIS
```

```
0       Afghanistan
1       Afghanistan
2       Afghanistan
3       Afghanistan
4       Afghanistan
           ...
1699       Zimbabwe
1700       Zimbabwe
1701       Zimbabwe
1702       Zimbabwe
1703       Zimbabwe
Name: Country, Length: 1704, dtype: object
```



```
# HOMEWORK - WHY I SHOULD NOT USE THIS STYLE
```

```
df.drop("continent", axis=1) # for permanent changes use inplace=True
```

	Country	year	Population	life_exp	gdp_cap
3	Afghanistan	1967	11537966	34.020	836.197138

```
df["year+7"] = df["year"] + 7
```

```
df["gdp"] = df["gdp_cap"] * df["Population"]
```

```
df
```

	Country	year	Population	continent	life_exp	gdp_cap	gdp
0	Afghanistan	1952	8425333	Asia	28.801	779.445314	6.567086e+09
1	Afghanistan	1957	9240934	Asia	30.332	820.853030	7.585449e+09
2	Afghanistan	1962	10267083	Asia	31.997	853.100710	8.758856e+09
3	Afghanistan	1967	11537966	Asia	34.020	836.197138	9.648014e+09
4	Afghanistan	1972	13079460	Asia	36.088	739.981106	9.678553e+09
...	...	...	...	...	...	...	...
1699	Zimbabwe	1987	9216418	Africa	62.351	706.157306	6.508241e+09
1700	Zimbabwe	1992	10704340	Africa	60.377	693.420786	7.422612e+09
1701	Zimbabwe	1997	11404948	Africa	46.809	792.449960	9.037851e+09
1702	Zimbabwe	2002	11926563	Africa	39.989	672.038623	8.015111e+09
1703	Zimbabwe	2007	12311143	Africa	43.487	469.709298	5.782658e+09

1704 rows × 7 columns

Saving...

```
df["Own"] = [i for i in range(len(df))]
```

```
df
```

	Country	year	Population	continent	life_exp	gdp_cap	gdp
0	Afghanistan	1952	8425333	Asia	28.801	779.445314	6.567086e+09
1	Afghanistan	1957	9240934	Asia	30.332	820.853030	7.585449e+09

```
# df.insert() that helps in adding a column at a particular location
```

3	Afghanistan	1967	11537966	Asia	34.020	836.197138	9.648014e+09

```
# Working with Rows


ser = df["Country"]
ser
```

```
0       Afghanistan
1       Afghanistan
2       Afghanistan
3       Afghanistan
4       Afghanistan
           ...
1699       Zimbabwe
1700       Zimbabwe
1701       Zimbabwe
1702       Zimbabwe
1703       Zimbabwe
Name: Country, Length: 1704, dtype: object
```

```
ser[0]
```

Saving...

```
ser[5:14]
```

```
5     Afghanistan
6     Afghanistan
7     Afghanistan
8     Afghanistan
9     Afghanistan
10    Afghanistan
11    Afghanistan
12        Albania
13        Albania
Name: Country, dtype: object
```

```
df[0] # indexing a row like doesnt happan in dataframe
# df["country"] #looks for this index along axis=1
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.7/dist-packages/pandas/core/indexes/base.py in
get_loc(self, key, method, tolerance)
   3360             try:
-> 3361                 return self._engine.get_loc(casted_key)
   3362             except KeyError as err:

                                4 frames

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 0

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.7/dist-packages/pandas/core/indexes/base.py in
get_loc(self, key, method, tolerance)
   3361                 return self._engine.get_loc(casted_key)
```

```
df[5:14]
```

	Country	year	Population	continent	life_exp	gdp_cap	gdp	O
5	Afghanistan	1977	14880372	Asia	38.438	786.113360	1.169766e+10	
			316	Asia	39.854	978.011439	1.259856e+10	
			957	Asia	40.822	852.395945	1.182099e+10	
8	Afghanistan	1992	16317921	Asia	41.674	649.341395	1.059590e+10	
9	Afghanistan	1997	22227415	Asia	41.763	635.341351	1.412200e+10	
10	Afghanistan	2002	25268405	Asia	42.129	726.734055	1.836341e+10	
11	Afghanistan	2007	31889923	Asia	43.828	974.580338	3.107929e+10	
12	Albania	1952	1282697	Europe	55.230	1601.056136	2.053670e+09	
13	Albania	1957	1476505	Europe	59.280	1942.284244	2.867792e+09	

Saving...

```
# ===> Indexing doesnt work to index rows because of similar syntax for columns
# ===> Slicing works for slicing the rows
```

```
df.index.values
```

```
array([   0,    1,    2, ..., 1701, 1702, 1703])
```

```
df.columns
```

```
Index(['Country', 'year', 'Population', 'continent', 'life_exp', 'gdp_cap',
       'gdp', 'Own'],
      dtype='object')
```

```
df.index = range(1, df.shape[0]+1)
```

```
df
```

	Country	year	Population	continent	life_exp	gdp_cap	gdp
1	Afghanistan	1952	8425333	Asia	28.801	779.445314	6.567086e+09
2	Afghanistan	1957	9240934	Asia	30.332	820.853030	7.585449e+09
3	Afghanistan	1962	10267083	Asia	31.997	853.100710	8.758856e+09
4	Afghanistan	1967	11537966	Asia	34.020	836.197138	9.648014e+09
5	Afghanistan	1972	13079460	Asia	36.088	739.981106	9.678553e+09
...	...	...	...	...	...	...	...
1700	Zimbabwe	1987	9216418	Africa	62.351	706.157306	6.508241e+09
1701	Zimbabwe	1992	10704340	Africa	60.377	693.420786	7.422612e+09
1702	Zimbabwe	1997	11404948	Africa	46.809	792.449960	9.037851e+09
1703	Zimbabwe	2002	11926563	Africa	39.989	672.038623	8.015111e+09
1704	Zimbabwe	2007	12311143	Africa	43.487	469.709298	5.782658e+09

1704 rows × 8 columns


```
2
```

```
df.index = np.arange(1, df.shape[0]+1, dtype='float')
df
```

	Country	year	Population	continent	life_exp	gdp_cap	gdp
1.0	Afghanistan	1952	8425333	Asia	28.801	779.445314	6.567086e+09

```
sample = df.head()
```

```
sample
```

	Country	year	Population	continent	life_exp	gdp_cap	gdp	Ow
1.0	Afghanistan	1952	8425333	Asia	28.801	779.445314	6.567086e+09	
2.0	Afghanistan	1957	9240934	Asia	30.332	820.853030	7.585449e+09	
3.0	Afghanistan	1962	10267083	Asia	31.997	853.100710	8.758856e+09	
4.0	Afghanistan	1967	11537966	Asia	34.020	836.197138	9.648014e+09	
5.0	Afghanistan	1972	13079460	Asia	36.088	739.981106	9.678553e+09	

```
sample.index = ["a", "b", "c", "d", "e"]
```

```
sample
```

	Country	year	Population	continent	life_exp	gdp_cap	gdp	Own
			33	Asia	28.801	779.445314	6.567086e+09	0
			34	Asia	30.332	820.853030	7.585449e+09	1
c	Afghanistan	1962	10267083	Asia	31.997	853.100710	8.758856e+09	2
d	Afghanistan	1967	11537966	Asia	34.020	836.197138	9.648014e+09	3
e	Afghanistan	1972	13079460	Asia	36.088	739.981106	9.678553e+09	4


```
sample.index = ["a", "b", "c", "d", "d"]
```

```
sample
```

	Country	year	Population	continent	life_exp	gdp_cap	gdp	Own
a	Afghanistan	1952	8425333	Asia	28.801	779.445314	6.567086e+09	0
b	Afghanistan	1957	9240934	Asia	30.332	820.853030	7.585449e+09	1
c	Afghanistan	1962	10267083	Asia	31.997	853.100710	8.758856e+09	2
d	Afghanistan	1967	11537966	Asia	34.020	836.197138	9.648014e+09	3
d	Afghanistan	1972	13079460	Asia	36.088	739.981106	9.678553e+09	4

```
df.columns
```

```
Index(['Country', 'year', 'Population', 'continent', 'life_exp', 'gdp_cap',
       'gdp', 'Own'],
      dtype='object')
```

```
df.columns = ['Country', 'year', 'Population', 'continent', 'life_exp', 'gdp_cap',
       'gdp', 'Country']
```

```
df
```

	Country	year	Population	continent	life_exp	gdp_cap	gdp
1.0	Afghanistan	1952	8425333	Asia	28.801	779.445314	6.567086e+09
2.0	Afghanistan	1957	9240934	Asia	30.332	820.853030	7.585449e+09
3.0	Afghanistan	1962	10267083	Asia	31.997	853.100710	8.758856e+09
4.0	Afghanistan	1967	11537966	Asia	34.020	836.197138	9.648014e+09
5.0	Afghanistan	1972	13079460	Asia	36.088	739.981106	9.678553e+09
...	...	...	...	...	...	...	...
1700.0	Zimbabwe	1987	9216418	Africa	62.351	706.157306	6.508241e+09
1701.0	Zimbabwe	1992	10704340	Africa	60.377	693.420786	7.422612e+09
1702.0	Zimbabwe	1997	11404948	Africa	46.809	792.449960	9.037851e+09
			926563	Africa	39.989	672.038623	8.015111e+09
			2311143	Africa	43.487	469.709298	5.782658e+09

1704 rows × 8 columns


```
# duplication of explciit indexes is ok
```

```
df["Country"] # you can basically group the data, classification using same explcit
```

	Country	Country
1.0	Afghanistan	0
2.0	Afghanistan	1
3.0	Afghanistan	2

```
df.reset_index() # drop=True would have dropped the colum
```

	index	Country	year	Population	continent	life_exp	gdp_cap	
0	1.0	Afghanistan	1952	8425333	Asia	28.801	779.445314	6.56708
1	2.0	Afghanistan	1957	9240934	Asia	30.332	820.853030	7.58544
2	3.0	Afghanistan	1962	10267083	Asia	31.997	853.100710	8.75885
3	4.0	Afghanistan	1967	11537966	Asia	34.020	836.197138	9.64801
4	5.0	Afghanistan	1972	13079460	Asia	36.088	739.981106	9.67855
...	...	...	...	...	...	...	...	
1699	1700.0	Zimbabwe	1987	9216418	Africa	62.351	706.157306	6.50824
1700	1701.0	Zimbabwe	1992	10704340	Africa	60.377	693.420786	7.42261
1701	1702.0	Zimbabwe	1997	11404948	Africa	46.809	792.449960	9.03785
1702	1703.0	Zimbabwe	2002	11926563	Africa	39.989	672.038623	8.01511
1703	1704.0	Zimbabwe	2007	12311143	Africa	43.487	469.709298	5.78265

Saving...

```
# Learner's personal doubt
```

Double-click (or enter) to edit

```
x = pd.DataFrame({'veclocity':[100, -11, -16, 13, 14]})

def find_closest(x):
  return np.argmin(np.abs(np.array([3, 30, -5, -15]) - x))

x["veclocity"].apply(find_closest)
```

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