

Colab Link:

<https://colab.research.google.com/drive/1JDpZgAO5OK1XDzXoPupwsN1ucLOHRNXC?usp=sharing>

Data Link: [https://drive.google.com/file/d/1E3bwvYGf1ig32RmcYiWc0IXPN-mD\\_bI\\_/view?usp=sharing](https://drive.google.com/file/d/1E3bwvYGf1ig32RmcYiWc0IXPN-mD_bI_/view?usp=sharing)

## ▼ Imagine that you are working as a Data Consultant for McKinsey Digital

```
!pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/pythor
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packa
```

```
import pandas as pd
```

```
# GDP per capita and Life Expectancy
```

```
!gdown 1E3bwvYGf1ig32RmcYiWc0IXPN-mD_bI_
```

```
Downloading...
```

```
From: https://drive.google.com/uc?id=1E3bwvYGf1ig32RmcYiWc0IXPN-mD\_bI\_
```

```
To: /content/gapminder.csv
```

```
100% 83.8k/83.8k [00:00<00:00, 73.1MB/s]
```

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

Saved successfully!



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



```
type(df)
```

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

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

```
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
```

```
type(df["country"]) # vector
```

```
pandas.core.series.Series
```

```
# dataframe as matrix
```

```
# series as vector, stack multiple series together --> pandas dataframe
```

```
df.info()
```

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

```
RangeIndex: 1704 entries, 0 to 1703
```

```
Data columns (total 6 columns):
```

		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
```

Saved successfully!



```
df.head(11)
```

	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
5	Afghanistan	1977	14880372	Asia	38.438	786.113360
6	Afghanistan	1982	12881816	Asia	39.854	978.011439
7	Afghanistan	1987	13867957	Asia	40.822	852.395945

```
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

```
df.shape
```

```
(1704, 6)
```

```
df.describe()
```

Saved successfully!

			life_exp	gdp_cap
count	1704.00000	1.704000e+03	1704.000000	1704.000000
mean	1979.50000	2.960121e+07	59.474439	7215.327081
std	17.26533	1.061579e+08	12.917107	9857.454543
min	1952.00000	6.001100e+04	23.599000	241.165876
25%	1965.75000	2.793664e+06	48.198000	1202.060309
50%	1979.50000	7.023596e+06	60.712500	3531.846988
75%	1993.25000	1.958522e+07	70.845500	9325.462346
max	2007.00000	1.318683e+09	82.603000	113523.132900

```
df.describe(include="object")
```

	country	continent
count	1704	1704
unique	142	5
top	Afghanistan	Africa
freq	12	624



# Basic Operations on Columns

```
df.columns

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

```
df.keys()

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

# specialised dictionary

```
df["country"]

0      Afghanistan
1      Afghanistan
2      Afghanistan
3      Afghanistan
4      Afghanistan
...
1699    Zimbabwe
1700    Zimbabwe
1701    Zimbabwe
1702    Zimbabwe
1703    Zimbabwe
dtype: object
```

Saved successfully!

df[["country", "life\_exp"]] # not possible in a Python Dictinary

	country	life_exp
0	Afghanistan	28.801
1	Afghanistan	30.332
2	Afghanistan	31.997
3	Afghanistan	34.020

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

```
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',
      'Vietnam', 'West Bank and Gaza', 'Yemen, Rep.',
      ], dtype=object)
```

Saved successfully!

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

```
Afghanistan      12
Pakistan         12
New Zealand      12
Nicaragua        12
Niger            12
..
Eritrea          12
Equatorial Guinea 12
El Salvador      12
Egypt            12
Zimbabwe         12
Name: country, Length: 142, dtype: int64
```

```
type(df["country"].value_counts()) # will come to this later
```

```
pandas.core.series.Series
```

```
df["country"].value_counts().index # will come to this later
```

```
Index(['Afghanistan', 'Pakistan', 'New Zealand', 'Nicaragua', 'Niger',
      'Nigeria', 'Norway', 'Oman', 'Panama', 'Nepal',
      ...,
      'Gambia', 'Gabon', 'France', 'Finland', 'Ethiopia', 'Eritrea',
      'Equatorial Guinea', 'El Salvador', 'Egypt', 'Zimbabwe'],
      dtype='object', length=142)
```

```
# rename a column name
```

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

	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 x 6 columns

Saved successfully!



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



```
df.rename({"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
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 x 6 columns

```
axis=1, inplace=True)
```

Saved successfully!



```
df["country"] # dictionary way
```

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, not recommended
```

```
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
```

```
# column name has some space
# whgat if name of the column is same as one of the attribute
# column names are contain numbers

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

	country	year	population	life_exp	gdp_cap
0	Afghanistan	1952	8425333	28.801	779.445314
1	Afghanistan	1957	9240934	30.332	820.853030
2	Afghanistan	1962	10267083	31.997	853.100710
3	Afghanistan	1967	11537966	34.020	836.197138
4	Afghanistan	1972	13079460	36.088	739.981106
...	...	...	...	...	...
1699	Zimbabwe	1987	9216418	62.351	706.157306
1700	Zimbabwe	1992	10704340	60.377	693.420786
1701	Zimbabwe	1997	11404948	46.809	792.449960
1702	Zimbabwe	2002	1226563	39.989	672.038623
1703	Zimbabwe	2007	1211143	43.487	469.709298

Saved successfully!

1704 rows x 5 columns

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



```
df["total_gdp"] = df["population"] * df["gdp_cap"]
```

df

	country	year	population	continent	life_exp	gdp_cap	total_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	12211143	Africa	43.487	469.709298	5.782658e+09

Saved successfully!

✕

```
# list, array, series
df["custom"] = [i for i in range(1704)]
```

df

	country	year	population	continent	life_exp	gdp_cap	total_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

```
df.drop(["custom", "total_gdp"], axis=1, inplace= True)
1699 Zimbabwe 1987 9216418 Africa 62.351 706.157306 6.508241e+09
```

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
			11143	Africa	43.487	469.709298



Saved successfully!



1704 rows x 6 columns

# 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[3] # 4th row, indexing
```

```
'Afghanistan'
```

```
# 4th-15th row, slicing
ser[4:16]
```

```

4      Afghanistan
5      Afghanistan
6      Afghanistan
7      Afghanistan
8      Afghanistan
9      Afghanistan
10     Afghanistan
11     Afghanistan
12      Albania
13      Albania
14      Albania
15      Albania
Name: country, dtype: object

```

```
ser.index
```

```
RangeIndex(start=0, stop=1704, step=1)
```

```

import numpy as np
ser.index = np.arange(1, df.shape[0]+1, 1)

```

Saved successfully!



```

2      Afghanistan
3      Afghanistan
4      Afghanistan
5      Afghanistan
...
1700     Zimbabwe
1701     Zimbabwe
1702     Zimbabwe
1703     Zimbabwe
1704     Zimbabwe
Name: country, Length: 1704, dtype: object

```

```
# Implicit and Explicit Indices
```

```
ser.index
```

```
Int64Index([    1,     2,     3,     4,     5,     6,     7,     8,     9,    10,
            ...
           1695, 1696, 1697, 1698, 1699, 1700, 1701, 1702, 1703, 1704],
           dtype='int64', length=1704)
```

```
ser.index[1]
```

```
2
```

```
data = pd.Series(["a", "b", "c"], index=[1, 5, 3])
```

```
data
```

```
1    a
5    b
3    c
dtype: object
```

```
type(data)
```

```
pandas.core.series.Series
```

```
# indexing and slicing
```

```
data[1] # indexing uses explicit indices
```

```
'a'
```

```
data[1:3] # slicing uses implicit or positional indices
```

```
5    b
3    c
```

Saved successfully!

```
data = pd.Series(['a', 'b', 'c'], index=['x', 'y', 'z'])
data
```

```
x    a
y    b
z    c
dtype: object
```

```
data["x"] # indexes uses explicit indexing
```

```
'a'
```

```
data[:1] # positional indexing
```

```
x    a
```

```
dtype: object
```

```
data = pd.Series(['a', 'b', 'c'], index=[1, 2, 2])
data
```

```
1    a
2    b
2    c
dtype: object
```

```
# To avoid confusion Pandas has special indexers
# 1. loc - works with explicit indexes/labels
# 2. iloc - works with implicit indexes/positional indexes
```

```
data
```

```
1    a
2    b
2    c
dtype: object
```

```
data.loc[1]
```

```
'a'
```

```
data.loc[2]
```

```
2    b
2    c
dtype: object
```

```
data.iloc[1]
```

```
'b'
```

Saved successfully!



```
'c'
```

```
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
...	...	...	...	...	...	...

```
df.iloc[[1, 10, 100]]
```

	country	year	population	continent	life_exp	gdp_cap
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
10	Afghanistan	2002	25268405	Asia	42.129	726.734055
100	Bangladesh	1972	70759295	Asia	45.252	630.233627

```
df.loc[[1, 10, 100]]
```

	country	year	population	continent	life_exp	gdp_cap
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
10	Afghanistan	2002	25268405	Asia	42.129	726.734055
100	Bangladesh	1972	70759295	Asia	45.252	630.233627

```
df.iloc[-1] # row is a series
```

```
country      Zimbabwe
year         2007
```

Saved successfully!

```
life_exp      46.709298
gdp_cap       469.709298
Name: 1703, dtype: object
```

```
df.loc[-1]
```

-----

ValueError

Traceback (most recent call last)

[/usr/local/lib/python3.7/dist-packages/pandas/core/indexes/range.py](#) in `get_loc(self, key, method, tolerance)`

384                   try:

--> 385                    return self.\_range.index(new\_key)

386                   except ValueError as err:

ValueError: -1 is not in range


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

KeyError


Traceback (most recent call last)

----- ^ 5 frames -----

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	

1704 rows x 6 columns

Saved successfully! 

```
temp = df.set_index("country")
```

temp

	year	population	continent	life_exp	gdp_cap	
country						
Afghanistan	1952	8425333	Asia	28.801	779.445314	
Afghanistan	1957	9240934	Asia	30.332	820.853030	
Afghanistan	1962	10267083	Asia	31.997	853.100710	
Afghanistan	1967	11537966	Asia	34.020	836.197138	
Afghanistan	1972	13079460	Asia	36.088	739.981106	

```
temp["life_exp"].loc["Zimbabwe"] # usyaklky recommended to keep row labels unique
```

```
country
Zimbabwe    48.451
Zimbabwe    50.469
Zimbabwe    52.358
Zimbabwe    53.995
Zimbabwe    55.635
Zimbabwe    57.674
Zimbabwe    60.363
Zimbabwe    62.351
Zimbabwe    60.377
Zimbabwe    46.809
Zimbabwe    39.989
Zimbabwe    43.487
Name: life_exp, dtype: float64
```

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	
			10267083	Asia	31.997	853.100710	
			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 x 6 columns



```
# Add a row

d = {'country': 'India', 'year': 2000, 'life_exp':37.08,
      'population':135000, "continent": "Asia", "gdp_cap": 100}

d

{'continent': 'Asia',
 'country': 'India',
 'gdp_cap': 100,
 'life_exp': 37.08,
 'population': 135000,
 'year': 2000}

df.append(d, ignore_index=True)
# df.append(['India',2023,13500000,"Asia", 37.08,900.23], ignore_index=True)
```

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1962	10267083	Asia	31.997	853.100710
2	Afghanistan	1967	11537966	Asia	34.020	836.197138
3	Afghanistan	1972	13079460	Asia	36.088	739.981106
4	Afghanistan	1977	14880372	Asia	38.438	786.113360
...	...	...	...	...	...	...
1701	Zimbabwe	2002	11926563	Africa	39.989	672.038623
1702	Zimbabwe	2007	12311143	Africa	43.487	469.709298
1703	India	2000	13500000	Asia	37.080	900.230000
1704	India	2000	13500000	Asia	37.080	900.230000
1705	India	2000	135000	Asia	37.080	100.000000



Saved successfully!

✕

```
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

df.loc[len(df.index)] = ['India',2000,13500000,"Asia", 37.08,900.23]

df.loc[len(df.index)] = ['India',2000,13500000,"Asia", 37.08,900.23]

1700 Zimbabwe 1997 11404948 Africa 46.809 792.449960

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
...	...	...	...	...	...	...
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	India	2000	13500000	Asia	37.080	900.230000
1705	India	2000	13500000	Asia	37.080	900.230000

Saved successfully!

df.iloc[len(df.index)] = ['India',2000,13500000,"Asia", 37.08,900.23]

```

-----
IndexError                                Traceback (most recent call last)
<ipython-input-127-c87277c41b8f> in <module>()
----> 1 df.iloc[len(df.index)] = ['India',2000,13500000,"Asia", 37.08,900.23]

```

```
# drop rows, drop duplicates
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/indexing.py in
```

```
df.duplicated() # mask
```

```

0      False
1      False
2      False
3      False
4      False
...
1701   False
1702   False
1703   False
1704   False
1705    True
Length: 1706, dtype: bool

```

```
df.loc[df.duplicated()]
```

	country	year	population	continent	life_exp	gdp_cap
<b>1705</b>	India	2000	13500000	Asia	37.08	900.23

```
df.drop_duplicates(keep="last")
```

```
# first, last, False
```

	country	year	population	continent	life_exp	gdp_cap
<b>0</b>	Afghanistan	1952	8425333	Asia	28.801	779.445314
<b>1</b>	Afghanistan	1957	9240934	Asia	30.332	820.853030
			37083	Asia	31.997	853.100710
<b>3</b>	Afghanistan	1967	11537966	Asia	34.020	836.197138
<b>4</b>	Afghanistan	1972	13079460	Asia	36.088	739.981106
...	...	...	...	...	...	...
<b>1700</b>	Zimbabwe	1992	10704340	Africa	60.377	693.420786
<b>1701</b>	Zimbabwe	1997	11404948	Africa	46.809	792.449960
<b>1702</b>	Zimbabwe	2002	11926563	Africa	39.989	672.038623
<b>1703</b>	Zimbabwe	2007	12311143	Africa	43.487	469.709298
<b>1705</b>	India	2000	13500000	Asia	37.080	900.230000

```
1705 rows x 6 columns
```

```
# delete the row
```

```
df.drop(1, axis=0, inplace=True)
```

```
df
```

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
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
5	Afghanistan	1977	14880372	Asia	38.438	786.113360
...	...	...	...	...	...	...
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	India	2000	13500000	Asia	37.080	900.230000
1705	India	2000	13500000	Asia	37.080	900.230000

1705 rows x 6 columns

```
df.iloc[1]
```

country	Afghanistan
year	1962
population	10267083
continent	Asia
life_exp	31.997

Saved successfully!

```
df.loc[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:

7 frames
pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.Int64HashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.Int64HashTable.get_item()
```

# Work with rows and columns

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

```
df.iloc[1:5, 0:4]
```

	country	year	population	continent
2	Afghanistan	1962	10267083	Asia
3	Afghanistan	1967	11537966	Asia
4	Afghanistan	1972	13079460	Asia
5	Afghanistan	1977	14880372	Asia

df

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
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
5	Afghanistan	1977	14880372	Asia	38.438	786.113360
...	...	...	...	...	...	...
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	India	2000	13500000	Asia	37.080	900.230000
1705	India	2000	13500000	Asia	37.080	900.230000

1705 rows x 6 columns

```
df.loc[1:5, 0:4]
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-146-05daf87a861d> in <module>()
----> 1 df.loc[1:5, 0:4]
```

8 frames

```
/usr/local/lib/python3.7/dist-packages/pandas/core/indexes/base.py in
_maybe_cast_slice_bound(self, label, side, kind)
    5747         # reject them, if index does not contain label
    5748         if (is_float(label) or is_integer(label)) and label not in
self._values:
-> 5749             raise self._invalid_indexer("slice", label)
    5750
    5751         return label
```

**TypeError:** cannot do slice indexing on Index with these indexers [0] of type int

```
df.loc[1:5, ["country", "life_exp"]] # this will work even if 1 isnt there
# slicing with loc considers the end point as well
```

	country	life_exp
2	Afghanistan	31.997
3	Afghanistan	34.020
4	Afghanistan	36.088
5	Afghanistan	38.438

```
df.loc[1:5, "year":"life_exp"]
```

	year	population	continent	life_exp
2	1962	10267082	Asia	31.997
3	1967	10267082	Asia	34.020
4	1972	13079460	Asia	36.088
5	1977	14880372	Asia	38.438

Saved successfully!

```
df.iloc[[0,10,100], [0,2,3]]
```

	country	population	continent
0	Afghanistan	8425333	Asia
11	Afghanistan	31889923	Asia
101	Bangladesh	80428306	Asia

```
df.iloc[1:10:2]
```

	country	year	population	continent	life_exp	gdp_cap
2	Afghanistan	1962	10267083	Asia	31.997	853.100710
4	Afghanistan	1972	13079460	Asia	36.088	739.981106
6	Afghanistan	1982	12881816	Asia	39.854	978.011439
8	Afghanistan	1992	16317921	Asia	41.674	649.341395
10	Afghanistan	2002	25268405	Asia	42.129	726.734055



```
# How to select records from 30th to 40th row for the last 3 columns using iloc?
```

```
# a. df.iloc[29:40,-3:]
```

```
# b. df.iloc[30:39,-3:]
```

```
# c. df.iloc[31:41,-3:]
```

```
# d. df.iloc[29:39,-3:]
```

```
df["life_exp"].mean()
```

```
59.46526256891491
```

```
df["life_exp"].sum()
```

```
101388.27268
```

```
df["life_exp"].count()
```

```
1705
```


```
df["life_exp"].sum() / df["life_exp"].count()
```

```
59.46526256891495
```

Saved successfully!



```
df
```

	country	year	population	continent	life_exp	gdp_cap	
0	Afghanistan	1952	8425333	Asia	28.801	779.445314	
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	
5	Afghanistan	1977	14880372	Asia	38.438	786.113360	

```
df.sort_values(["year"])
```

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
924	Malawi	1952	2917802	Africa	36.256	369.165080
144	Bosnia and Herzegovina	1952	2791000	Europe	53.820	973.533195
912	Madagascar	1952	4762912	Africa	36.681	1443.011715
900	Libya	1952	1019729	Africa	42.723	2387.548060
...	...	...	...	...	...	...
1211	Peru	2007	28674757	Americas	71.421	7408.905561
1223	Philippines	2007	91077287	Asia	71.688	3190.481016
1235	Poland	2007	38518241	Europe	75.563	15389.924680
143	Bolivia	2007	9119152	Americas	65.554	3822.137084
131	Benin	2007	8078314	Africa	56.728	1441.284873

1705 rows x 6 columns

```
df.sort_values(["year"], ascending=False)
```

Saved successfully!





```

country year population continent life_exp gdp_cap
143 Bolivia 2007 9119152 Americas 65.554 3822.137084
1451 Sudan 2007 12202020 Africa 58.556 2602.301005
df.sort_values(["life_exp"])

```

	country	year	population	continent	life_exp	gdp_cap	
1292	Rwanda	1992	7290203	Africa	23.599	737.068595	
0	Afghanistan	1952	8425333	Asia	28.801	779.445314	
552	Gambia	1952	284320	Africa	30.000	485.230659	
36	Angola	1952	4232095	Africa	30.015	3520.610273	
1344	Sierra Leone	1952	2143249	Africa	30.331	879.787736	
...	...	...	...	...	...	...	
1487	Switzerland	2007	7554661	Europe	81.701	37506.419070	
695	Iceland	2007	301931	Europe	81.757	36180.789190	
802	Japan	2002	127065841	Asia	82.000	28604.591900	
671	Hong Kong, China	2007	6980412	Asia	82.208	39724.978670	
803	Japan	2007	127467972	Asia	82.603	31656.068060	

1705 rows x 6 columns

```
df.sort_values(["year", "life_exp"])
```

	country	year	population	continent	life_exp	gdp_cap	
0	Afghanistan	1952	8425333	Asia	28.801	779.445314	
552	Gambia	1952	284320	Africa	30.000	485.230659	
			4232095	Africa	30.015	3520.610273	
1344	Sierra Leone	1952	2143249	Africa	30.331	879.787736	
1032	Mozambique	1952	6446316	Africa	31.286	468.526038	
...	...	...	...	...	...	...	
71	Australia	2007	20434176	Oceania	81.235	34435.367440	
1487	Switzerland	2007	7554661	Europe	81.701	37506.419070	
695	Iceland	2007	301931	Europe	81.757	36180.789190	
671	Hong Kong, China	2007	6980412	Asia	82.208	39724.978670	
803	Japan	2007	127467972	Asia	82.603	31656.068060	

1705 rows x 6 columns

```
df.sort_values(["year", "life_exp"], ascending=[False, True])
```

	country	year	population	continent	life_exp	gdp_cap
<b>1463</b>	Swaziland	2007	1133066	Africa	39.613	4513.480643
<b>1043</b>	Mozambique	2007	19951656	Africa	42.082	823.685621
<b>1691</b>	Zambia	2007	11746035	Africa	42.384	1271.211593
<b>1355</b>	Sierra Leone	2007	6144562	Africa	42.568	862.540756
<b>887</b>	Lesotho	2007	2012649	Africa	42.592	1569.331442
...	...	...	...	...	...	...
<b>408</b>	Denmark	1952	4334000	Europe	70.780	9692.385245
<b>1464</b>	Sweden	1952	7124673	Europe	71.860	8527.844662
<b>1080</b>	Netherlands	1952	10381988	Europe	72.130	8941.571858
<b>684</b>	Iceland	1952	147962	Europe	72.490	7267.688428
<b>1140</b>	Norway	1952	3327728	Europe	72.670	10095.421720

1705 rows x 6 columns

```
# Create a dataframe from scratch
```

```
df.loc[0]
```

```
country      Afghanistan
year          1952
population    8425333
continent     Asia
life_exp      28.801
gdp_cap       779.445314
Name: 0, dtype: object
```

Saved successfully!

```
0      Afghanistan
2      Afghanistan
3      Afghanistan
4      Afghanistan
5      Afghanistan
...
1701    Zimbabwe
1702    Zimbabwe
1703    Zimbabwe
1704      India
1705      India
Name: country, Length: 1705, dtype: object
```

```
pd.Series([10, 20, 30], index=["a", "b", "c"])
```

```
a    10
b    20
c    30
dtype: int64
```

```
# dataframe
# Approach-1: Row-oriented
# Approach-2: Column Oriented
```

```
# row oriented
pd.DataFrame([[10, 20], [30, 40]], columns=["A", "B"])
```

	A	B
0	10	20
1	30	40

```
# column oriented approach
pd.DataFrame({"A": [10, 30], "B": [20, 40]})
```

	A	B
0	10	20
1	30	40

```
# Pending: Concatenation of dataframes, merging/joining of dataframe - please ignore
# concat(), merge()
```

Saved successfully!