

In [1]:

```
1 !pip install pandas
```

Requirement already satisfied: pandas in /Users/anantm/opt/anaconda3/lib/python3.8/site-packages (1.2.4)  
 Requirement already satisfied: python-dateutil>=2.7.3 in /Users/anantm/opt/anaconda3/lib/python3.8/site-packages (from pandas) (2.8.1)  
 Requirement already satisfied: pytz>=2017.3 in /Users/anantm/opt/anaconda3/lib/python3.8/site-packages (from pandas) (2021.1)  
 Requirement already satisfied: numpy>=1.16.5 in /Users/anantm/opt/anaconda3/lib/python3.8/site-packages (from pandas) (1.22.2)  
 Requirement already satisfied: six>=1.5 in /Users/anantm/opt/anaconda3/lib/python3.8/site-packages (from python-dateutil>=2.7.3->pandas) (1.15.0)

In [2]:

```
1 import pandas as pd
```

In [3]:

```
1 df = pd.read_csv("data/gapminder.csv")
```

In [4]:

```
1 df
```

Out[4]:

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

1704 rows × 6 columns

In [5]:

```
1 type(df)
```

Out[5]:

pandas.core.frame.DataFrame

In [6]:

```
1 # dictionary - specialised dictionary
```

In [7]:

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

Out[7]:

```
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
```

In [8]:

```
1 df[["country", "continent"]]
```

Out[8]:

	country	continent
0	Afghanistan	Asia
1	Afghanistan	Asia
2	Afghanistan	Asia
3	Afghanistan	Asia
4	Afghanistan	Asia
...	...	...
1699	Zimbabwe	Africa
1700	Zimbabwe	Africa
1701	Zimbabwe	Africa
1702	Zimbabwe	Africa
1703	Zimbabwe	Africa

1704 rows × 2 columns

In [10]:

```
1 type(df["country"])
```

Out[10]:

pandas.core.series.Series

In [12]:

```
1 df.head(9)
```

Out[12]:

	country	continent	year	lifeExp	population	gdpPerCap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
3	Afghanistan	Asia	1967	34.020	11537966	836.197138
4	Afghanistan	Asia	1972	36.088	13079460	739.981106
5	Afghanistan	Asia	1977	38.438	14880372	786.113360
6	Afghanistan	Asia	1982	39.854	12881816	978.011439
7	Afghanistan	Asia	1987	40.822	13867957	852.395945
8	Afghanistan	Asia	1992	41.674	16317921	649.341395

In [13]:

```
1 df.tail(20)
```

Out[13]:

	country	continent	year	lifeExp	population	gdpPerCap
1684	Zambia	Africa	1972	50.107	4506497	1773.498265
1685	Zambia	Africa	1977	51.386	5216550	1588.688299
1686	Zambia	Africa	1982	51.821	6100407	1408.678565
1687	Zambia	Africa	1987	50.821	7272406	1213.315116
1688	Zambia	Africa	1992	46.100	8381163	1210.884633
1689	Zambia	Africa	1997	40.238	9417789	1071.353818
1690	Zambia	Africa	2002	39.193	10595811	1071.613938
1691	Zambia	Africa	2007	42.384	11746035	1271.211593
1692	Zimbabwe	Africa	1952	48.451	3080907	406.884115
1693	Zimbabwe	Africa	1957	50.469	3646340	518.764268
1694	Zimbabwe	Africa	1962	52.358	4277736	527.272182
1695	Zimbabwe	Africa	1967	53.995	4995432	569.795071
1696	Zimbabwe	Africa	1972	55.635	5861135	799.362176
1697	Zimbabwe	Africa	1977	57.674	6642107	685.587682
1698	Zimbabwe	Africa	1982	60.363	7636524	788.855041
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786
1701	Zimbabwe	Africa	1997	46.809	11404948	792.449960
1702	Zimbabwe	Africa	2002	39.989	11926563	672.038623
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298

In [14]:

```
1 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   continent       1704 non-null   object
2   year            1704 non-null   int64
3   lifeExp         1704 non-null   float64
4   population      1704 non-null   int64
5   gdpPerCap       1704 non-null   float64
dtypes: float64(2), int64(2), object(2)
memory usage: 80.0+ KB
```

In [15]:

```
1 df.describe()
```

Out[15]:

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

In [19]:

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

Out[19]:

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

In [20]:

```
1 df.index[4]
```

Out[20]:

4

In [21]:

```
1 df.index[3]
```

Out[21]:

3

In [22]:

```
1 df
```

Out[22]:

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

1704 rows × 6 columns

In [24]:

```
1 df["country"][2]
```

Out[24]:

'Afghanistan'

In [27]:

```
1 df_temp = df.set_index("continent")
```

In [28]:

```
1 df_temp["country"]["Asia"] # this is okay to use
```

Out[28]:

```
continent
Asia    Afghanistan
Asia    Afghanistan
Asia    Afghanistan
Asia    Afghanistan
Asia    Afghanistan
...
Asia    Yemen, Rep.
Asia    Yemen, Rep.
Asia    Yemen, Rep.
Asia    Yemen, Rep.
Asia    Yemen, Rep.
Name: country, Length: 396, dtype: object
```

In [31]:

```
1 df_temp["country"][2] # not to use it
```

Out[31]:

'Afghanistan'

In [34]:

```
1 df_temp["country"]
```

Out[34]:

```
continent
Asia      Afghanistan
Asia      Afghanistan
Asia      Afghanistan
Asia      Afghanistan
Asia      Afghanistan
...
Africa     Zimbabwe
Africa     Zimbabwe
Africa     Zimbabwe
Africa     Zimbabwe
Africa     Zimbabwe
Name: country, Length: 1704, dtype: object
```

In [35]:

```
1 # columns
```

In [36]:

```
1 df.columns
```

Out[36]:

```
Index(['country', 'continent', 'year', 'lifeExp', 'population', 'gdpPe  
rCap'], dtype='object')
```

In [37]:

```
1 df.keys()
```

Out[37]:

```
Index(['country', 'continent', 'year', 'lifeExp', 'population', 'gdpPe  
rCap'], dtype='object')
```

In [39]:

```
1 df[["country", "continent"]]
```

Out[39]:

	country	continent
0	Afghanistan	Asia
1	Afghanistan	Asia
2	Afghanistan	Asia
3	Afghanistan	Asia
4	Afghanistan	Asia
...	...	...
1699	Zimbabwe	Africa
1700	Zimbabwe	Africa
1701	Zimbabwe	Africa
1702	Zimbabwe	Africa
1703	Zimbabwe	Africa

1704 rows × 2 columns

In [40]:

```
1 df.country # attrubite style references
```

Out[40]:

```
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



In [41]:

```
1 df.continent
```

Out[41]:

```
0      Asia
1      Asia
2      Asia
3      Asia
4      Asia
...
1699   Africa
1700   Africa
1701   Africa
1702   Africa
1703   Africa
Name: continent, Length: 1704, dtype: object
```

In [42]:

```
1 df.country is df["country"]
```

Out[42]:

True

In [44]:

```
1 df.drop("continent", axis=1)
```

Out[44]:

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

1704 rows × 5 columns

In [46]:

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

In [47]:

1	df
---	----

Out[47]:

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

1704 rows × 5 columns

In [49]:

1	df = pd.read_csv("data/gapminder.csv")
2	df

Out[49]:

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

1704 rows × 6 columns

In [50]:

```
1 # iloc and loc
```

In [52]:

```
1 ser = df["country"]  
2 ser
```

Out[52]:

```
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
```

In [55]:

```
1 ser[3] # indexing
```

Out[55]:

```
'Afghanistan'
```

In [56]:

```
1 ser[3:15]
```

Out[56]:

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

In [58]:

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

Out[58]:

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

In [59]:

```
1 data[1] # using explicit index
```

Out[59]:

```
'a'
```

In [60]:

```
1 data[1:3] # using implicit indices
```

Out[60]:

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

In [61]:

```
1 # indexing - explicit
2 # slicing - implicit
3 # iloc and loc
4 # loc - labels
5 # iloc - positional index, implicit indices
```

In [62]:

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

Out[62]:

```
'b'
```

In [63]:

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

Out[63]:

```
'a'
```

In [65]:

```
1 data.iloc[1:3]
```

Out[65]:

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

In [66]:

```
1 data.loc[1:3]
```

Out[66]:

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

In [67]:

```
1 data.loc[2:3]
```

```
-----
-----
KeyError                                Traceback (most recent call
last)
~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.p
y in get_loc(self, key, method, tolerance)
    3079         try:
-> 3080             return self._engine.get_loc(casted_key)
    3081         except KeyError as err:

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.Int6
4HashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.Int6
4HashTable.get_item()
```

KeyError: 2

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

```
KeyError                                Traceback (most recent call
last)
<ipython-input-67-29f1ad944393> in <module>
----> 1 data.loc[2:3]

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in
__getitem__(self, key)
    893
    894         maybe_callable = com.apply_if_callable(key, self.o
bj)
-> 895         return self._getitem_axis(maybe_callable, axis=axis)
    896
    897     def _is_scalar_access(self, key: Tuple):

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in
_getitem_axis(self, key, axis)
    1100         if isinstance(key, slice):
    1101             self._validate_key(key, axis)
-> 1102             return self._get_slice_axis(key, axis=axis)
    1103         elif com.is_bool_indexer(key):
    1104             return self._getbool_axis(key, axis=axis)

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in
_get_slice_axis(self, slice_obj, axis)
    1134
    1135         labels = obj._get_axis(axis)
-> 1136         indexer = labels.slice_indexer(
    1137             slice_obj.start, slice_obj.stop, slice_obj.step, k
ind="loc"
    1138         )
```

```
~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.p
```

```

y in slice_indexer(self, start, end, step, kind)
    5275         slice(1, 3, None)
    5276         """
-> 5277         start_slice, end_slice = self.slice_locs(start, end, s
tep=step, kind=kind)
    5278
    5279         # return a slice

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.p
y in slice_locs(self, start, end, step, kind)
    5474         start_slice = None
    5475         if start is not None:
-> 5476             start_slice = self.get_slice_bound(start, "left",
kind)
    5477         if start_slice is None:
    5478             start_slice = 0

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.p
y in get_slice_bound(self, label, side, kind)
    5394         except ValueError:
    5395             # raise the original KeyError
-> 5396             raise err
    5397
    5398         if isinstance(slc, np.ndarray):

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.p
y in get_slice_bound(self, label, side, kind)
    5388         # we need to look up the label
    5389         try:
-> 5390             slc = self.get_loc(label)
    5391         except KeyError as err:
    5392             try:

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.p
y in get_loc(self, key, method, tolerance)
    3080         return self._engine.get_loc(casted_key)
    3081         except KeyError as err:
-> 3082             raise KeyError(key) from err
    3083
    3084         if tolerance is not None:

```

KeyError: 2

In [70]:

```
1 df.values
```

Out[70]:

```

array([[ 'Afghanistan', 'Asia', 1952, 28.801, 8425333, 779.4453145],
       [ 'Afghanistan', 'Asia', 1957, 30.332, 9240934, 820.8530296],
       [ 'Afghanistan', 'Asia', 1962, 31.997, 10267083, 853.10071],
       ...,
       [ 'Zimbabwe', 'Africa', 1997, 46.809, 11404948, 792.4499603],
       [ 'Zimbabwe', 'Africa', 2002, 39.989, 11926563, 672.0386227],
       [ 'Zimbabwe', 'Africa', 2007, 43.487, 12311143, 469.7092981]],
      dtype=object)

```

In [71]:

```
1 type(df.values[0, 1])
```

Out[71]:

str

In [72]:

```
1 type(df.values[0, 2])
```

Out[72]:

int

In [74]:

```
1 df
```

Out[74]:

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

1704 rows × 6 columns

In [73]:

```
1 df.loc[3]
```

Out[73]:

```
country      Afghanistan
continent    Asia
year         1967
lifeExp      34.02
population   11537966
gdpPerCap    836.197138
Name: 3, dtype: object
```



In [75]:

```
1 df.iloc[3]
```

Out[75]:

```
country      Afghanistan
continent      Asia
year          1967
lifeExp       34.02
population    11537966
gdpPerCap     836.197138
Name: 3, dtype: object
```

In [76]:

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

Out[76]:

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

In [77]:

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

Out[77]:

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

In [78]:

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

Out[78]:

```
country      Zimbabwe
continent     Africa
year          2007
lifeExp       43.487
population    12311143
gdpPerCap     469.709298
Name: 1703, dtype: object
```

In [ ]:

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

In [85]:

```
1 df.iloc[1:6, 1:4]
```

Out[85]:

	continent	year	lifeExp
1	Asia	1957	30.332
2	Asia	1962	31.997
3	Asia	1967	34.020
4	Asia	1972	36.088
5	Asia	1977	38.438

In [82]:

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

```
-----
-----
TypeError                                Traceback (most recent call
last)
<ipython-input-82-494208dc7680> in <module>
----> 1 df.loc[1:5, 1:4]

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in
__getitem__(self, key)
    887                                     # AttributeError for IntervalTree get_valu
e
    888                                     return self.obj._get_value(*key, takeable=
self._takeable)
--> 889                                     return self._getitem_tuple(key)
    890                                     else:
    891                                     # we by definition only have the 0th axis

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in
_getitem_tuple(self, tup)
    1067                                     return self._multi_take(tup)
    1068
-> 1069                                     return self._getitem_tuple_same_dim(tup)
    1070
    1071     def _get_label(self, label, axis: int):

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in
_getitem_tuple_same_dim(self, tup)
    773                                     continue
    774
--> 775                                     retval = getattr(retval, self.name)._getitem_axis(
key, axis=i)
    776                                     # We should never have retval.ndim < self.ndim, as
that should
    777                                     # be handled by the _getitem_lowerdim call above.

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in
_getitem_axis(self, key, axis)
    1100                                     if isinstance(key, slice):
    1101                                     self._validate_key(key, axis)
-> 1102                                     return self._get_slice_axis(key, axis=axis)
    1103                                     elif com.is_bool_indexer(key):
    1104                                     return self._getbool_axis(key, axis=axis)

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in
_get_slice_axis(self, slice_obj, axis)
    1134
    1135                                     labels = obj._get_axis(axis)
-> 1136                                     indexer = labels.slice_indexer(
    1137                                     slice_obj.start, slice_obj.stop, slice_obj.step, k
ind="loc"
    1138                                     )

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.p
y in slice_indexer(self, start, end, step, kind)
    5275                                     slice(1, 3, None)
    5276                                     """
-> 5277                                     start_slice, end_slice = self.slice_locs(start, end, s
```

```

tep=step, kind=kind)
    5278
    5279         # return a slice

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.p
y in slice_locs(self, start, end, step, kind)
    5474         start_slice = None
    5475         if start is not None:
-> 5476             start_slice = self.get_slice_bound(start, "left",
kind)
    5477         if start_slice is None:
    5478             start_slice = 0

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.p
y in get_slice_bound(self, label, side, kind)
    5384         # For datetime indices label may be a string that has
to be converted
    5385         # to datetime boundary according to its resolution.
-> 5386         label = self._maybe_cast_slice_bound(label, side, kind
)
    5387
    5388         # we need to look up the label

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.p
y in _maybe_cast_slice_bound(self, label, side, kind)
    5336         # reject them, if index does not contain label
    5337         if (is_float(label) or is_integer(label)) and label no
t in self.values:
-> 5338             raise self._invalid_indexer("slice", label)
    5339
    5340         return label

```

**TypeError:** cannot do slice indexing on Index with these indexers [1] o  
f type int

In [84]:

```
1 df.loc[1:5, ["country", "lifeExp"]]
```

Out[84]:

	country	lifeExp
1	Afghanistan	30.332
2	Afghanistan	31.997
3	Afghanistan	34.020
4	Afghanistan	36.088
5	Afghanistan	38.438

In [88]:

```
1 df.loc[1:5, "country":"population"]
```

Out[88]:

	country	continent	year	lifeExp	population
1	Afghanistan	Asia	1957	30.332	9240934
2	Afghanistan	Asia	1962	31.997	10267083
3	Afghanistan	Asia	1967	34.020	11537966
4	Afghanistan	Asia	1972	36.088	13079460
5	Afghanistan	Asia	1977	38.438	14880372

In [89]:

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

Out[89]:

	country	year	lifeExp
0	Afghanistan	1952	28.801
10	Afghanistan	2002	42.129
100	Bangladesh	1972	45.252

In [90]:

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

Out[90]:

	country	continent	year	lifeExp	population	gdpPerCap
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
3	Afghanistan	Asia	1967	34.020	11537966	836.197138
5	Afghanistan	Asia	1977	38.438	14880372	786.113360
7	Afghanistan	Asia	1987	40.822	13867957	852.395945
9	Afghanistan	Asia	1997	41.763	22227415	635.341351

In [92]:

```
1 df.drop(3, axis=0, inplace=True)
```

In [93]:

1 df

Out[93]:

	country	continent	year	lifeExp	population	gdpPerCap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
4	Afghanistan	Asia	1972	36.088	13079460	739.981106
5	Afghanistan	Asia	1977	38.438	14880372	786.113360
...	...	...	...	...	...	...
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786
1701	Zimbabwe	Africa	1997	46.809	11404948	792.449960
1702	Zimbabwe	Africa	2002	39.989	11926563	672.038623
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298

1703 rows × 6 columns

In [94]:

1 df.iloc[3]

Out[94]:

```
country      Afghanistan
continent      Asia
year          1972
lifeExp       36.088
population    13079460
gdpPerCap     739.981106
Name: 4, dtype: object
```

In [95]:

```
1 df.loc[3]
```

```
-----
-----
KeyError                                Traceback (most recent call
last)
~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.p
y in get_loc(self, key, method, tolerance)
    3079         try:
-> 3080             return self._engine.get_loc(casted_key)
    3081         except KeyError as err:

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.Int6
4HashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.Int6
4HashTable.get_item()
```

KeyError: 3

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

```
KeyError                                Traceback (most recent call
last)
<ipython-input-95-2d3184acc116> in <module>
----> 1 df.loc[3]

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in
__getitem__(self, key)
    893
    894         maybe_callable = com.apply_if_callable(key, self.o
bj)
-> 895         return self._getitem_axis(maybe_callable, axis=axis)
    896
    897     def _is_scalar_access(self, key: Tuple):

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in
_getitem_axis(self, key, axis)
    1122         # fall thru to straight lookup
    1123         self._validate_key(key, axis)
-> 1124         return self._get_label(key, axis=axis)
    1125
    1126     def _get_slice_axis(self, slice_obj: slice, axis: int):

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in
_get_label(self, label, axis)
    1071     def _get_label(self, label, axis: int):
    1072         # GH#5667 this will fail if the label is not present i
n the axis.
-> 1073         return self.obj.xs(label, axis=axis)
    1074
    1075     def _handle_lowerdim_multi_index_axis0(self, tup: Tuple):

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/generic.py in
```

```

xs(self, key, axis, level, drop_level)
3737         raise TypeError(f"Expected label or tuple of 1
abels, got {key}") from e
3738     else:
-> 3739         loc = index.get_loc(key)
3740
3741         if isinstance(loc, np.ndarray):

~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.p
y in get_loc(self, key, method, tolerance)
3080         return self._engine.get_loc(casted_key)
3081     except KeyError as err:
-> 3082         raise KeyError(key) from err
3083
3084         if tolerance is not None:

```

KeyError: 3

In [96]:

```
1 df
```

Out[96]:

	country	continent	year	lifeExp	population	gdpPerCap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
4	Afghanistan	Asia	1972	36.088	13079460	739.981106
5	Afghanistan	Asia	1977	38.438	14880372	786.113360
...	...	...	...	...	...	...
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786
1701	Zimbabwe	Africa	1997	46.809	11404948	792.449960
1702	Zimbabwe	Africa	2002	39.989	11926563	672.038623
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298

1703 rows × 6 columns

In [97]:

```
1 df["lifeExp"].mean()
```

Out[97]:

59.489386189078054



In [98]:

```
1 df["lifeExp"].sum()
```

Out[98]:

101310.42468000001

In [99]:

```
1 df["lifeExp"].count()
```

Out[99]:

1703

In [100]:

```
1 df["lifeExp"].sum()/df["lifeExp"].count()
```

Out[100]:

59.489386189078104

In [102]:

```
1 import numpy as np
2 np.sin(df["lifeExp"] * np.pi)
```

Out[102]:

```
0      0.585241
1      0.863923
2     -0.009425
4      0.272952
5      0.981091
...
1699   0.892428
1700   0.926266
1701   0.564679
1702  -0.034551
1703  -0.999166
Name: lifeExp, Length: 1703, dtype: float64
```

In [103]:

```
1 # filter thr rows where lifexp > 60
2 # SELECT * FROM df WHERE lifeExp > 60
```

In [104]:

```
1 df["lifeExp"] > 60 # masking
```

Out[104]:

```
0      False
1      False
2      False
4      False
5      False
...
1699    True
1700    True
1701    False
1702    False
1703    False
Name: lifeExp, Length: 1703, dtype: bool
```

In [107]:

```
1 # filtering
2 df.loc[df["lifeExp"] > 60]
```

Out[107]:

	country	continent	year	lifeExp	population	gdpPerCap
14	Albania	Europe	1962	64.820	1728137	2312.888958
15	Albania	Europe	1967	66.220	1984060	2760.196931
16	Albania	Europe	1972	67.690	2263554	3313.422188
17	Albania	Europe	1977	68.930	2509048	3533.003910
18	Albania	Europe	1982	70.420	2780097	3630.880722
...	...	...	...	...	...	...
1678	Yemen, Rep.	Asia	2002	60.308	18701257	2234.820827
1679	Yemen, Rep.	Asia	2007	62.698	22211743	2280.769906
1698	Zimbabwe	Africa	1982	60.363	7636524	788.855041
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786

877 rows × 6 columns

In [108]:

```
1 df.loc[df.lifeExp > 60]
```

Out[108]:

	country	continent	year	lifeExp	population	gdpPerCap
14	Albania	Europe	1962	64.820	1728137	2312.888958
15	Albania	Europe	1967	66.220	1984060	2760.196931
16	Albania	Europe	1972	67.690	2263554	3313.422188
17	Albania	Europe	1977	68.930	2509048	3533.003910
18	Albania	Europe	1982	70.420	2780097	3630.880722
...	...	...	...	...	...	...
1678	Yemen, Rep.	Asia	2002	60.308	18701257	2234.820827
1679	Yemen, Rep.	Asia	2007	62.698	22211743	2280.769906
1698	Zimbabwe	Africa	1982	60.363	7636524	788.855041
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786

877 rows × 6 columns

In [112]:

```
1 df.loc[df.lifeExp>60, ['country', 'lifeExp']]
```

Out[112]:

	country	lifeExp
14	Albania	64.820
15	Albania	66.220
16	Albania	67.690
17	Albania	68.930
18	Albania	70.420
...	...	...
1678	Yemen, Rep.	60.308
1679	Yemen, Rep.	62.698
1698	Zimbabwe	60.363
1699	Zimbabwe	62.351
1700	Zimbabwe	60.377

877 rows × 2 columns

In [123]:

```
1 df[df["lifeExp"] > 60]
```

Out[123]:

	country	continent	year	lifeExp	population	gdpPerCap
14	Albania	Europe	1962	64.820	1728137	2312.888958
15	Albania	Europe	1967	66.220	1984060	2760.196931
16	Albania	Europe	1972	67.690	2263554	3313.422188
17	Albania	Europe	1977	68.930	2509048	3533.003910
18	Albania	Europe	1982	70.420	2780097	3630.880722
...	...	...	...	...	...	...
1678	Yemen, Rep.	Asia	2002	60.308	18701257	2234.820827
1679	Yemen, Rep.	Asia	2007	62.698	22211743	2280.769906
1698	Zimbabwe	Africa	1982	60.363	7636524	788.855041
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786

877 rows × 6 columns

In [129]:

```
1 df.loc[(df["lifeExp"] < 60) & (df["lifeExp"] > 30)]
```

Out[129]:

	country	continent	year	lifeExp	population	gdpPerCap
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
4	Afghanistan	Asia	1972	36.088	13079460	739.981106
5	Afghanistan	Asia	1977	38.438	14880372	786.113360
6	Afghanistan	Asia	1982	39.854	12881816	978.011439
...	...	...	...	...	...	...
1696	Zimbabwe	Africa	1972	55.635	5861135	799.362176
1697	Zimbabwe	Africa	1977	57.674	6642107	685.587682
1701	Zimbabwe	Africa	1997	46.809	11404948	792.449960
1702	Zimbabwe	Africa	2002	39.989	11926563	672.038623
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298

823 rows × 6 columns

In [130]:

```

1 # "Kenya" or "Egypt
2 df.loc[(df["country"] == 'Kenya') | (df["country"] == 'Egypt')]

```

Out[130]:

	country	continent	year	lifeExp	population	gdpPerCap
456	Egypt	Africa	1952	41.893	22223309	1418.822445
457	Egypt	Africa	1957	44.444	25009741	1458.915272
458	Egypt	Africa	1962	46.992	28173309	1693.335853
459	Egypt	Africa	1967	49.293	31681188	1814.880728
460	Egypt	Africa	1972	51.137	34807417	2024.008147
461	Egypt	Africa	1977	53.319	38783863	2785.493582
462	Egypt	Africa	1982	56.006	45681811	3503.729636
463	Egypt	Africa	1987	59.797	52799062	3885.460710
464	Egypt	Africa	1992	63.674	59402198	3794.755195
465	Egypt	Africa	1997	67.217	66134291	4173.181797
466	Egypt	Africa	2002	69.806	73312559	4754.604414
467	Egypt	Africa	2007	71.338	80264543	5581.180998
816	Kenya	Africa	1952	42.270	6464046	853.540919
817	Kenya	Africa	1957	44.686	7454779	944.438315
818	Kenya	Africa	1962	47.949	8678557	896.966373
819	Kenya	Africa	1967	50.654	10191512	1056.736457
820	Kenya	Africa	1972	53.559	12044785	1222.359968
821	Kenya	Africa	1977	56.155	14500404	1267.613204
822	Kenya	Africa	1982	58.766	17661452	1348.225791
823	Kenya	Africa	1987	59.339	21198082	1361.936856
824	Kenya	Africa	1992	59.285	25020539	1341.921721
825	Kenya	Africa	1997	54.407	28263827	1360.485021
826	Kenya	Africa	2002	50.992	31386842	1287.514732
827	Kenya	Africa	2007	54.110	35610177	1463.249282

In [131]:

```
1 # "Nigeria"
2 df.loc[df["country"] < 'Nigeria']
```

Out[131]:

	country	continent	year	lifeExp	population	gdpPerCap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
4	Afghanistan	Asia	1972	36.088	13079460	739.981106
5	Afghanistan	Asia	1977	38.438	14880372	786.113360
...	...	...	...	...	...	...
1123	Niger	Africa	1987	44.555	7332638	668.300023
1124	Niger	Africa	1992	47.391	8392818	581.182725
1125	Niger	Africa	1997	51.313	9666252	580.305209
1126	Niger	Africa	2002	54.496	11140655	601.074501
1127	Niger	Africa	2007	56.867	12894865	619.676892

1127 rows × 6 columns

In [132]:

```
1 # Calculate the average life expectancy of Asia
2 df.loc[df["continent"] == "Asia", "lifeExp"].mean()
```

Out[132]:

60.13083969620247

In [133]:

```
1 df["continent"] == "Asia"
```

Out[133]:

```
0      True
1      True
2      True
4      True
5      True
...
1699   False
1700   False
1701   False
1702   False
1703   False
Name: continent, Length: 1703, dtype: bool
```

