pandas-lecture-2-dec-batch

June 6, 2023

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
df = pd.read_csv('mckinsey.csv')
```

[3]:
```
df
```

[3]:

	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]

[5]:
```
temp = pd.DataFrame([['a', 1, 'xyz', 123]], columns=['a', 'b', 'c', 'd'])
temp
```

[5]:

	a	b	c	d
0	a	1	xyz	123

[6]:
```
df.columns
```

[6]:
```
Index(['country', 'year', 'population', 'continent', 'life_exp', 'gdp_cap'],
dtype='object')
```

[7]:
```
df.index.values
```

[7]:
```
array([   0,    1,    2, …, 1701, 1702, 1703])
```

[8]:
```
df.index = list(range(1, df.shape[0]+1))
df
```

[8]:

	country	year	population	continent	life_exp	gdp_cap
1	Afghanistan	1952	8425333	Asia	28.801	779.445314
2	Afghanistan	1957	9240934	Asia	30.332	820.853030
3	Afghanistan	1962	10267083	Asia	31.997	853.100710
4	Afghanistan	1967	11537966	Asia	34.020	836.197138
5	Afghanistan	1972	13079460	Asia	36.088	739.981106
…	…	…	…	…	…	…
1700	Zimbabwe	1987	9216418	Africa	62.351	706.157306
1701	Zimbabwe	1992	10704340	Africa	60.377	693.420786
1702	Zimbabwe	1997	11404948	Africa	46.809	792.449960
1703	Zimbabwe	2002	11926563	Africa	39.989	672.038623
1704	Zimbabwe	2007	12311143	Africa	43.487	469.709298

[1704 rows x 6 columns]

[10]:
```
df.index[1]
```

[10]: 2

[11]:
```
df.index = np.arange(1, 1705, dtype='float')
df
```

[11]:

	country	year	population	continent	life_exp	gdp_cap
1.0	Afghanistan	1952	8425333	Asia	28.801	779.445314
2.0	Afghanistan	1957	9240934	Asia	30.332	820.853030
3.0	Afghanistan	1962	10267083	Asia	31.997	853.100710
4.0	Afghanistan	1967	11537966	Asia	34.020	836.197138
5.0	Afghanistan	1972	13079460	Asia	36.088	739.981106
…	…	…	…	…	…	…
1700.0	Zimbabwe	1987	9216418	Africa	62.351	706.157306
1701.0	Zimbabwe	1992	10704340	Africa	60.377	693.420786
1702.0	Zimbabwe	1997	11404948	Africa	46.809	792.449960
1703.0	Zimbabwe	2002	11926563	Africa	39.989	672.038623
1704.0	Zimbabwe	2007	12311143	Africa	43.487	469.709298

[1704 rows x 6 columns]

[12]:
```
df.index[1]
```

[12]: 2.0

[13]:
```
sample = df.head()
sample
```

[13]:
```
         country  year  population continent  life_exp     gdp_cap
1.0  Afghanistan  1952     8425333      Asia    28.801  779.445314
2.0  Afghanistan  1957     9240934      Asia    30.332  820.853030
3.0  Afghanistan  1962    10267083      Asia    31.997  853.100710
4.0  Afghanistan  1967    11537966      Asia    34.020  836.197138
5.0  Afghanistan  1972    13079460      Asia    36.088  739.981106
```

[14]:
```
sample.index = ['a', 'b', 'c', 'd', 'e']
sample
```

[14]:
```
       country  year  population continent  life_exp     gdp_cap
a  Afghanistan  1952     8425333      Asia    28.801  779.445314
b  Afghanistan  1957     9240934      Asia    30.332  820.853030
c  Afghanistan  1962    10267083      Asia    31.997  853.100710
d  Afghanistan  1967    11537966      Asia    34.020  836.197138
e  Afghanistan  1972    13079460      Asia    36.088  739.981106
```

[16]:
```
df.index = list(range(1, 1705))
df
```

[16]:
```
          country  year  population continent  life_exp     gdp_cap
1     Afghanistan  1952     8425333      Asia    28.801  779.445314
2     Afghanistan  1957     9240934      Asia    30.332  820.853030
3     Afghanistan  1962    10267083      Asia    31.997  853.100710
4     Afghanistan  1967    11537966      Asia    34.020  836.197138
5     Afghanistan  1972    13079460      Asia    36.088  739.981106
…             …     …           …         …         …         …
1700     Zimbabwe  1987     9216418    Africa    62.351  706.157306
1701     Zimbabwe  1992    10704340    Africa    60.377  693.420786
1702     Zimbabwe  1997    11404948    Africa    46.809  792.449960
1703     Zimbabwe  2002    11926563    Africa    39.989  672.038623
1704     Zimbabwe  2007    12311143    Africa    43.487  469.709298

[1704 rows x 6 columns]
```

[17]:
```
df.iloc[1]
```

[17]:
```
country       Afghanistan
year                 1957
population        9240934
continent            Asia
life_exp           30.332
gdp_cap          820.85303
Name: 2, dtype: object
```

[18]:
```
df.loc[1]
```

[18]:
```
country       Afghanistan
year                 1952
population        8425333
continent            Asia
life_exp            28.801
gdp_cap         779.445314
Name: 1, dtype: object
```

[19]:
```
df.iloc[[1, 10, 15]]
```

[19]:

	country	year	population	continent	life_exp	gdp_cap
2	Afghanistan	1957	9240934	Asia	30.332	820.853030
11	Afghanistan	2002	25268405	Asia	42.129	726.734055
16	Albania	1967	1984060	Europe	66.220	2760.196931

[20]:
```
df.loc[[1, 10, 15]]
```

[20]:

	country	year	population	continent	life_exp	gdp_cap
1	Afghanistan	1952	8425333	Asia	28.801	779.445314
10	Afghanistan	1997	22227415	Asia	41.763	635.341351
15	Albania	1962	1728137	Europe	64.820	2312.888958

[21]:
```
df.iloc[-1]
```

[21]:
```
country         Zimbabwe
year                2007
population      12311143
continent         Africa
life_exp           43.487
gdp_cap        469.709298
Name: 1704, dtype: object
```

[22]:
```
df.loc[-1]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File /usr/local/lib/python3.9/site-packages/pandas/core/indexes/base.py:3621, i
 ↪Index.get_loc(self, key, method, tolerance)
   3620 try:
-> 3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:

File /usr/local/lib/python3.9/site-packages/pandas/_libs/index.pyx:136, in
 ↪pandas._libs.index.IndexEngine.get_loc()

File /usr/local/lib/python3.9/site-packages/pandas/_libs/index.pyx:163, in
 ↪pandas._libs.index.IndexEngine.get_loc()
```

```
File pandas/_libs/hashtable_class_helper.pxi:2131, in pandas._libs.hashtable.
 ↪Int64HashTable.get_item()

File pandas/_libs/hashtable_class_helper.pxi:2140, in pandas._libs.hashtable.
 ↪Int64HashTable.get_item()

KeyError: -1

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

KeyError                                  Traceback (most recent call last)
Input In [22], in <cell line: 1>()
----> 1 df.loc[-1]

File /usr/local/lib/python3.9/site-packages/pandas/core/indexing.py:967, in␣
 ↪_LocationIndexer.__getitem__(self, key)
    964 axis = self.axis or 0
    966 maybe_callable = com.apply_if_callable(key, self.obj)
--> 967 return self._getitem_axis(maybe_callable, axis=axis)

File /usr/local/lib/python3.9/site-packages/pandas/core/indexing.py:1202, in␣
 ↪_LocIndexer._getitem_axis(self, key, axis)
   1200 # fall thru to straight lookup
   1201 self._validate_key(key, axis)
-> 1202 return self._get_label(key, axis=axis)

File /usr/local/lib/python3.9/site-packages/pandas/core/indexing.py:1153, in␣
 ↪_LocIndexer._get_label(self, label, axis)
   1151 def _get_label(self, label, axis: int):
   1152     # GH#5667 this will fail if the label is not present in the axis.
-> 1153     return self.obj.xs(label, axis=axis)

File /usr/local/lib/python3.9/site-packages/pandas/core/generic.py:3876, in␣
 ↪NDFrame.xs(self, key, axis, level, drop_level)
   3874             new_index = index[loc]
   3875 else:
-> 3876     loc = index.get_loc(key)
   3878     if isinstance(loc, np.ndarray):
   3879         if loc.dtype == np.bool_:

File /usr/local/lib/python3.9/site-packages/pandas/core/indexes/base.py:3623, i␣
 ↪Index.get_loc(self, key, method, tolerance)
   3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
   3624 except TypeError:
   3625     # If we have a listlike key, _check_indexing_error will raise
```

```
   3626     #  InvalidIndexError. Otherwise we fall through and re-raise
   3627     #  the TypeError.
   3628     self._check_indexing_error(key)

KeyError: -1
```

[23]:
```
df.iloc[1:5]
```

[23]:

	country	year	population	continent	life_exp	gdp_cap
2	Afghanistan	1957	9240934	Asia	30.332	820.853030
3	Afghanistan	1962	10267083	Asia	31.997	853.100710
4	Afghanistan	1967	11537966	Asia	34.020	836.197138
5	Afghanistan	1972	13079460	Asia	36.088	739.981106

[24]:
```
df.loc[1:5]
```

[24]:

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

[25]:
```
temp = df.set_index('country')
temp
```

[25]:

country	year	population	continent	life_exp	gdp_cap
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
…	…	…	…	…	…
Zimbabwe	1987	9216418	Africa	62.351	706.157306
Zimbabwe	1992	10704340	Africa	60.377	693.420786
Zimbabwe	1997	11404948	Africa	46.809	792.449960
Zimbabwe	2002	11926563	Africa	39.989	672.038623
Zimbabwe	2007	12311143	Africa	43.487	469.709298

[1704 rows x 5 columns]

[26]:
```
temp.loc['Afghanistan']
```

[26]:

country	year	population	continent	life_exp	gdp_cap
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
Afghanistan  1977    14880372      Asia    38.438  786.113360
Afghanistan  1982    12881816      Asia    39.854  978.011439
Afghanistan  1987    13867957      Asia    40.822  852.395945
Afghanistan  1992    16317921      Asia    41.674  649.341395
Afghanistan  1997    22227415      Asia    41.763  635.341351
Afghanistan  2002    25268405      Asia    42.129  726.734055
Afghanistan  2007    31889923      Asia    43.828  974.580338
```

[27]:
```
temp.iloc[1]
```

[27]:
```
year              1957
population      9240934
continent          Asia
life_exp         30.332
gdp_cap       820.85303
Name: Afghanistan, dtype: object
```

[30]:
```
temp.reset_index(inplace=True)
```

[31]:
```
temp
```

[31]:
```
          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]
```

[36]:
```
df.reset_index(drop=True)
```

[36]:
```
          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]
```

0.0.1 Add a row

[38]:
```
new_row = {'country': 'India', 'year': 2020, 'population': 876567898,␣
 ↪'life_exp': 56.65,
           'gdp_cap': 678.89}
df.append(new_row, ignore_index=True)
```

```
/var/folders/dd/pdgq_y0s7jx6r2n91_kx1jyw0000gn/T/ipykernel_3757/3639086027.py:3:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  df.append(new_row, ignore_index=True)
```

[38]:
```
          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
…           …   …         …        …       …       …
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        India  2020   876567898       NaN    56.650  678.890000

[1705 rows x 6 columns]
```

[39]:
```
df
```

[39]:
```
          country  year  population continent  life_exp     gdp_cap
1     Afghanistan  1952     8425333      Asia    28.801  779.445314
2     Afghanistan  1957     9240934      Asia    30.332  820.853030
3     Afghanistan  1962    10267083      Asia    31.997  853.100710
4     Afghanistan  1967    11537966      Asia    34.020  836.197138
5     Afghanistan  1972    13079460      Asia    36.088  739.981106
…           …   …         …        …       …       …
1700     Zimbabwe  1987     9216418    Africa    62.351  706.157306
```

```
1701     Zimbabwe  1992    10704340    Africa    60.377  693.420786
1702     Zimbabwe  1997    11404948    Africa    46.809  792.449960
1703     Zimbabwe  2002    11926563    Africa    39.989  672.038623
1704     Zimbabwe  2007    12311143    Africa    43.487  469.709298

[1704 rows x 6 columns]
```

[40]:
```
help(df.append)
```

```
Help on method append in module pandas.core.frame:

append(other, ignore_index: 'bool' = False, verify_integrity: 'bool' = False,
sort: 'bool' = False) -> 'DataFrame' method of pandas.core.frame.DataFrame
instance
    Append rows of `other` to the end of caller, returning a new object.

    .. deprecated:: 1.4.0
        Use :func:`concat` instead. For further details see
        :ref:`whatsnew_140.deprecations.frame_series_append`

    Columns in `other` that are not in the caller are added as new columns.

    Parameters
    ----------
    other : DataFrame or Series/dict-like object, or list of these
        The data to append.
    ignore_index : bool, default False
        If True, the resulting axis will be labeled 0, 1, …, n - 1.
    verify_integrity : bool, default False
        If True, raise ValueError on creating index with duplicates.
    sort : bool, default False
        Sort columns if the columns of `self` and `other` are not aligned.

        .. versionchanged:: 1.0.0

            Changed to not sort by default.

    Returns
    -------
    DataFrame
        A new DataFrame consisting of the rows of caller and the rows of
`other`.

    See Also
    --------
    concat : General function to concatenate DataFrame or Series objects.
```

```
    Notes
    -----
    If a list of dict/series is passed and the keys are all contained in
    the DataFrame's index, the order of the columns in the resulting
    DataFrame will be unchanged.

    Iteratively appending rows to a DataFrame can be more computationally
    intensive than a single concatenate. A better solution is to append
    those rows to a list and then concatenate the list with the original
    DataFrame all at once.

    Examples
    --------
    >>> df = pd.DataFrame([[1, 2], [3, 4]], columns=list('AB'), index=['x',
'y'])
    >>> df
       A  B
    x  1  2
    y  3  4
    >>> df2 = pd.DataFrame([[5, 6], [7, 8]], columns=list('AB'), index=['x',
'y'])
    >>> df.append(df2)
       A  B
    x  1  2
    y  3  4
    x  5  6
    y  7  8

    With `ignore_index` set to True:

    >>> df.append(df2, ignore_index=True)
       A  B
    0  1  2
    1  3  4
    2  5  6
    3  7  8

    The following, while not recommended methods for generating DataFrames,
    show two ways to generate a DataFrame from multiple data sources.

    Less efficient:

    >>> df = pd.DataFrame(columns=['A'])
    >>> for i in range(5):
    …     df = df.append({'A': i}, ignore_index=True)
    >>> df
       A
    0  0
```

```
1  1
2  2
3  3
4  4

More efficient:

>>> pd.concat([pd.DataFrame([i], columns=['A']) for i in range(5)],
…           ignore_index=True)
   A
0  0
1  1
2  2
3  3
4  4
```

[42]:
```
df.iloc[1703] = ['India', 2020, 98765678, 'Asia', 45.56, 678.87]
df
```

[42]:

	country	year	population	continent	life_exp	gdp_cap
1	Afghanistan	1952	8425333	Asia	28.801	779.445314
2	Afghanistan	1957	9240934	Asia	30.332	820.853030
3	Afghanistan	1962	10267083	Asia	31.997	853.100710
4	Afghanistan	1967	11537966	Asia	34.020	836.197138
5	Afghanistan	1972	13079460	Asia	36.088	739.981106
…	…	…	…	…	…	…
1700	Zimbabwe	1987	9216418	Africa	62.351	706.157306
1701	Zimbabwe	1992	10704340	Africa	60.377	693.420786
1702	Zimbabwe	1997	11404948	Africa	46.809	792.449960
1703	Zimbabwe	2002	11926563	Africa	39.989	672.038623
1704	India	2020	98765678	Asia	45.560	678.870000

[1704 rows x 6 columns]

[43]:
```
df.loc[len(df.index)] = ['India', 2021, 98765678, 'Asia', 45.56, 678.87]
df.loc[len(df.index)] = ['Srilanka', 2021, 98765678, 'Asia', 45.56, 678.87]
df.loc[len(df.index)] = ['Srilank', 2021, 98765678, 'Asia', 45.56, 678.87]
df.loc[len(df.index)] = ['Pakistan', 2021, 98765678, 'Asia', 45.56, 678.87]
df
```

[43]:

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

```
…           …   …          …        …        …         …
1700     Zimbabwe  1987     9216418    Africa    62.351  706.157306
1701     Zimbabwe  1992    10704340    Africa    60.377  693.420786
1702     Zimbabwe  1997    11404948    Africa    46.809  792.449960
1703     Zimbabwe  2002    11926563    Africa    39.989  672.038623
1704     Pakistan  2021    98765678      Asia    45.560  678.870000

[1704 rows x 6 columns]
```

[44]:
```
len(df.index)
```

[44]:
```
1704
```

[46]:
```
df.loc[len(df.index)+1] = ['Srilanka', 2021, 98765678, 'Asia', 45.56, 678.87]
df.loc[len(df.index)+1] = ['Srilanka', 2021, 98765678, 'Asia', 45.56, 678.87]
df
```

[46]:
```
          country  year  population continent  life_exp     gdp_cap
1     Afghanistan  1952     8425333      Asia    28.801  779.445314
2     Afghanistan  1957     9240934      Asia    30.332  820.853030
3     Afghanistan  1962    10267083      Asia    31.997  853.100710
4     Afghanistan  1967    11537966      Asia    34.020  836.197138
5     Afghanistan  1972    13079460      Asia    36.088  739.981106
…           …   …          …        …        …         …
1703     Zimbabwe  2002    11926563    Africa    39.989  672.038623
1704     Pakistan  2021    98765678      Asia    45.560  678.870000
1705        India  2021    98765678      Asia    45.560  678.870000
1706     Srilanka  2021    98765678      Asia    45.560  678.870000
1707     Srilanka  2021    98765678      Asia    45.560  678.870000

[1707 rows x 6 columns]
```

[49]:
```
df.drop(1705, axis=0, inplace=True)
```

[50]:
```
df
```

[50]:
```
          country  year  population continent  life_exp     gdp_cap
1     Afghanistan  1952     8425333      Asia    28.801  779.445314
2     Afghanistan  1957     9240934      Asia    30.332  820.853030
3     Afghanistan  1962    10267083      Asia    31.997  853.100710
4     Afghanistan  1967    11537966      Asia    34.020  836.197138
5     Afghanistan  1972    13079460      Asia    36.088  739.981106
…           …   …          …        …        …         …
1702     Zimbabwe  1997    11404948    Africa    46.809  792.449960
1703     Zimbabwe  2002    11926563    Africa    39.989  672.038623
1704     Pakistan  2021    98765678      Asia    45.560  678.870000
1706     Srilanka  2021    98765678      Asia    45.560  678.870000
```

```
1707     Srilanka  2021    98765678      Asia    45.560  678.870000

[1706 rows x 6 columns]
```

[51]:
```
df.drop([1701, 1702, 1703], axis=0)
```

[51]:
```
          country  year  population continent  life_exp     gdp_cap
1     Afghanistan  1952     8425333      Asia    28.801  779.445314
2     Afghanistan  1957     9240934      Asia    30.332  820.853030
3     Afghanistan  1962    10267083      Asia    31.997  853.100710
4     Afghanistan  1967    11537966      Asia    34.020  836.197138
5     Afghanistan  1972    13079460      Asia    36.088  739.981106
…             …    …          …         …       …          …
1699     Zimbabwe  1982     7636524    Africa    60.363  788.855041
1700     Zimbabwe  1987     9216418    Africa    62.351  706.157306
1704     Pakistan  2021    98765678      Asia    45.560  678.870000
1706     Srilanka  2021    98765678      Asia    45.560  678.870000
1707     Srilanka  2021    98765678      Asia    45.560  678.870000

[1703 rows x 6 columns]
```

[52]:
```
df.duplicated()
```

[52]:
```
1       False
2       False
3       False
4       False
5       False
        …
1702    False
1703    False
1704    False
1706    False
1707     True
Length: 1706, dtype: bool
```

[53]:
```
df.loc[df.duplicated()]
```

[53]:
```
       country  year  population continent  life_exp  gdp_cap
1707  Srilanka  2021    98765678      Asia     45.56   678.87
```

[57]:
```
df.drop_duplicates(keep=False)
```

[57]:
```
          country  year  population continent  life_exp     gdp_cap
1     Afghanistan  1952     8425333      Asia    28.801  779.445314
2     Afghanistan  1957     9240934      Asia    30.332  820.853030
3     Afghanistan  1962    10267083      Asia    31.997  853.100710
```

```
4     Afghanistan  1967    11537966      Asia    34.020  836.197138
5     Afghanistan  1972    13079460      Asia    36.088  739.981106
…             …    …           …         …       …          …
1700     Zimbabwe  1987     9216418    Africa    62.351  706.157306
1701     Zimbabwe  1992    10704340    Africa    60.377  693.420786
1702     Zimbabwe  1997    11404948    Africa    46.809  792.449960
1703     Zimbabwe  2002    11926563    Africa    39.989  672.038623
1704     Pakistan  2021    98765678      Asia    45.560  678.870000

[1704 rows x 6 columns]
```

0.0.2 Working with rows and Columns Togethers

[58]:
```
df.iloc[:4, :3]
```

[58]:
```
       country  year  population
1  Afghanistan  1952     8425333
2  Afghanistan  1957     9240934
3  Afghanistan  1962    10267083
4  Afghanistan  1967    11537966
```

[59]:
```
df.loc[:4, :3]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [59], in <cell line: 1>()
----> 1 df.loc[:4, :3]

File /usr/local/lib/python3.9/site-packages/pandas/core/indexing.py:961, in␣
 ↪_LocationIndexer.__getitem__(self, key)
    959     if self._is_scalar_access(key):
    960         return self.obj._get_value(*key, takeable=self._takeable)
--> 961     return self._getitem_tuple(key)
    962 else:
    963     # we by definition only have the 0th axis
    964     axis = self.axis or 0

File /usr/local/lib/python3.9/site-packages/pandas/core/indexing.py:1149, in␣
 ↪_LocIndexer._getitem_tuple(self, tup)
   1146 if self._multi_take_opportunity(tup):
   1147     return self._multi_take(tup)
-> 1149 return self._getitem_tuple_same_dim(tup)

File /usr/local/lib/python3.9/site-packages/pandas/core/indexing.py:827, in␣
 ↪_LocationIndexer._getitem_tuple_same_dim(self, tup)
    824 if com.is_null_slice(key):
    825     continue
```

```
--> 827 retval = getattr(retval, self.name)._getitem_axis(key, axis=i)
    828 # We should never have retval.ndim < self.ndim, as that should
    829 #  be handled by the _getitem_lowerdim call above.
    830 assert retval.ndim == self.ndim

File /usr/local/lib/python3.9/site-packages/pandas/core/indexing.py:1180, in␣
 ↪_LocIndexer._getitem_axis(self, key, axis)
   1178 if isinstance(key, slice):
   1179     self._validate_key(key, axis)
-> 1180     return self._get_slice_axis(key, axis=axis)
   1181 elif com.is_bool_indexer(key):
   1182     return self._getbool_axis(key, axis=axis)

File /usr/local/lib/python3.9/site-packages/pandas/core/indexing.py:1214, in␣
 ↪_LocIndexer._get_slice_axis(self, slice_obj, axis)
   1211     return obj.copy(deep=False)
   1213 labels = obj._get_axis(axis)
-> 1214 indexer =␣
 ↪labels.slice_indexer(slice_obj.start, slice_obj.stop, slice_obj.step)
   1216 if isinstance(indexer, slice):
   1217     return self.obj._slice(indexer, axis=axis)

File /usr/local/lib/python3.9/site-packages/pandas/core/indexes/base.py:6274, i␣
 ↪Index.slice_indexer(self, start, end, step, kind)
   6231 """
   6232 Compute the slice indexer for input labels and step.
   6233
   (…)
   6270 slice(1, 3, None)
   6271 """
   6272 self._deprecated_arg(kind, "kind", "slice_indexer")
-> 6274 start_slice, end_slice = self.slice_locs(start, end, step=step)
   6276 # return a slice
   6277 if not is_scalar(start_slice):

File /usr/local/lib/python3.9/site-packages/pandas/core/indexes/base.py:6490, i␣
 ↪Index.slice_locs(self, start, end, step, kind)
   6488 end_slice = None
   6489 if end is not None:
-> 6490     end_slice = self.get_slice_bound(end, "right")
   6491 if end_slice is None:
   6492     end_slice = len(self)

File /usr/local/lib/python3.9/site-packages/pandas/core/indexes/base.py:6393, i␣
 ↪Index.get_slice_bound(self, label, side, kind)
   6389 original_label = label
   6391 # For datetime indices label may be a string that has to be converted
   6392 # to datetime boundary according to its resolution.
```

```
-> 6393 label = self._maybe_cast_slice_bound(label, side)
   6395 # we need to look up the label
   6396 try:

File /usr/local/lib/python3.9/site-packages/pandas/core/indexes/base.py:6340, i
 ↪Index._maybe_cast_slice_bound(self, label, side, kind)
   6335 # We are a plain index here (sub-class override this method if they
   6336 # wish to have special treatment for floats/ints, e.g. Float64Index and
   6337 # datetimelike Indexes
   6338 # reject them, if index does not contain label
   6339 if (is_float(label) or is_integer(label)) and label not in self:
-> 6340     raise self._invalid_indexer("slice", label)
   6342 return label

TypeError: cannot do slice indexing on Index with these indexers [3] of type in
```

[60]:
```
df.loc[1:5, ['country', 'life_exp']]
```

[60]:

	country	life_exp
1	Afghanistan	28.801
2	Afghanistan	30.332
3	Afghanistan	31.997
4	Afghanistan	34.020
5	Afghanistan	36.088

[61]:
```
df.loc[1:5, 'country': 'life_exp']
```

[61]:

	country	year	population	continent	life_exp
1	Afghanistan	1952	8425333	Asia	28.801
2	Afghanistan	1957	9240934	Asia	30.332
3	Afghanistan	1962	10267083	Asia	31.997
4	Afghanistan	1967	11537966	Asia	34.020
5	Afghanistan	1972	13079460	Asia	36.088

[62]:
```
df.iloc[[1, 10, 15], [0, 1, 4]]
```

[62]:

	country	year	life_exp
2	Afghanistan	1957	30.332
11	Afghanistan	2002	42.129
16	Albania	1967	66.220

[63]:
```
df.loc[[1, 10, 15], ['country', 'year', 'life_exp']]
```

[63]:

	country	year	life_exp
1	Afghanistan	1952	28.801
10	Afghanistan	1997	41.763
15	Albania	1962	64.820

[64]:
```
df.iloc[1:10:2]
```

[64]:

	country	year	population	continent	life_exp	gdp_cap
2	Afghanistan	1957	9240934	Asia	30.332	820.853030
4	Afghanistan	1967	11537966	Asia	34.020	836.197138
6	Afghanistan	1977	14880372	Asia	38.438	786.113360
8	Afghanistan	1987	13867957	Asia	40.822	852.395945
10	Afghanistan	1997	22227415	Asia	41.763	635.341351

[66]:
```
df.loc[1:10:2, 'country': 'life_exp':2]
```

[66]:

	country	population	life_exp
1	Afghanistan	8425333	28.801
3	Afghanistan	10267083	31.997
5	Afghanistan	13079460	36.088
7	Afghanistan	12881816	39.854
9	Afghanistan	16317921	41.674

0.0.3 Sorting

[69]:
```
df.sort_values(['life_exp'], ascending=False)
```

[69]:

	country	year	population	continent	life_exp	gdp_cap
804	Japan	2007	127467972	Asia	82.603	31656.068060
672	Hong Kong, China	2007	6980412	Asia	82.208	39724.978670
803	Japan	2002	127065841	Asia	82.000	28604.591900
696	Iceland	2007	301931	Europe	81.757	36180.789190
1488	Switzerland	2007	7554661	Europe	81.701	37506.419070
…	…	…	…	…	…	…
1345	Sierra Leone	1952	2143249	Africa	30.331	879.787736
37	Angola	1952	4232095	Africa	30.015	3520.610273
553	Gambia	1952	284320	Africa	30.000	485.230659
1	Afghanistan	1952	8425333	Asia	28.801	779.445314
1293	Rwanda	1992	7290203	Africa	23.599	737.068595

[1706 rows x 6 columns]

[70]:
```
df.sort_values(['year', 'life_exp'])
```

[70]:

	country	year	population	continent	life_exp	gdp_cap
1	Afghanistan	1952	8425333	Asia	28.801	779.445314
553	Gambia	1952	284320	Africa	30.000	485.230659
37	Angola	1952	4232095	Africa	30.015	3520.610273
1345	Sierra Leone	1952	2143249	Africa	30.331	879.787736
1033	Mozambique	1952	6446316	Africa	31.286	468.526038
…	…	…	…	…	…	…
672	Hong Kong, China	2007	6980412	Asia	82.208	39724.978670

```
804              Japan  2007   127467972      Asia    82.603  31656.068060
1704          Pakistan  2021    98765678      Asia    45.560    678.870000
1706          Srilanka  2021    98765678      Asia    45.560    678.870000
1707          Srilanka  2021    98765678      Asia    45.560    678.870000

[1706 rows x 6 columns]
```

[71]:
```
df.sort_values(['year', 'life_exp'], ascending=[True, False])
```

[71]:
```
          country  year  population continent  life_exp       gdp_cap
1141       Norway  1952     3327728    Europe    72.670  10095.421720
685       Iceland  1952      147962    Europe    72.490   7267.688428
1081  Netherlands  1952    10381988    Europe    72.130   8941.571858
1465       Sweden  1952     7124673    Europe    71.860   8527.844662
409       Denmark  1952     4334000    Europe    70.780   9692.385245
…           …   …         …         …         …           …
1044   Mozambique  2007    19951656    Africa    42.082    823.685621
1464    Swaziland  2007     1133066    Africa    39.613   4513.480643
1704     Pakistan  2021    98765678      Asia    45.560    678.870000
1706     Srilanka  2021    98765678      Asia    45.560    678.870000
1707     Srilanka  2021    98765678      Asia    45.560    678.870000

[1706 rows x 6 columns]
```

[72]:
```
le = df['life_exp']
le
```

[72]:
```
1       28.801
2       30.332
3       31.997
4       34.020
5       36.088
         …
1702    46.809
1703    39.989
1704    45.560
1706    45.560
1707    45.560
Name: life_exp, Length: 1706, dtype: float64
```

[73]:
```
le.mean()
```

[73]:
```
59.45934213364595
```

[74]:
```
le.max()
```

[74]:
```
82.603
```

[75]:
```
le.min()
```

[75]:
```
23.599
```

[76]:
```
le.sum()
```

[76]:
```
101437.63767999999
```

[77]:
```
le.count()
```

[77]:
```
1706
```

[ ]:
```

```