Intro to Pandas

September 28, 2023

[91]:
```
# dir(pandas)
import pandas
dir(pandas)[:10]
```

[91]:
```
['ArrowDtype',
 'BooleanDtype',
 'Categorical',
 'CategoricalDtype',
 'CategoricalIndex',
 'DataFrame',
 'DateOffset',
 'DatetimeIndex',
 'DatetimeTZDtype',
 'ExcelFile']
```

[92]:
```
# dir(pd)
import pandas as my_pd
dir(my_pd)
import pandas as pd
dir(pd)[:10]
```

[92]:
```
['ArrowDtype',
 'BooleanDtype',
 'Categorical',
 'CategoricalDtype',
 'CategoricalIndex',
 'DataFrame',
 'DateOffset',
 'DatetimeIndex',
 'DatetimeTZDtype',
 'ExcelFile']
```

0.0.1 Part 1: Introduction to Pandas and Basic Data Structures

Pandas is a powerful Python library for data manipulation and analysis. It provides two primary data structures: `Series` and `DataFrame`.

- **Series**: A one-dimensional labeled array.

```
import pandas as pd

# Creating a Series
s = pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])
print(s)
```

- **DataFrame**: A two-dimensional labeled data structure, similar to a table in a database, an Excel spreadsheet, or a data frame in R.

```
# Creating a DataFrame
data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 35],
    'City': ['New York', 'San Francisco', 'Los Angeles']
}
df = pd.DataFrame(data)
print(df)
```

[ ]:
```
# Object Oriented Programming
# my_list = [1,2,3,4,5]
# my_list.append(6)
```

[18]:
```
my_series = pd.Series([1,2,3,4,5])
# print(type(my_series))
# print(type('a'))
my_series.sum()
my_series.mean()
my_series.median()
my_series.mode()
my_series.min()
my_series.max()
my_series.std()
```

[18]:
```
15
```

[23]:
```
my_series.index
list(my_series.index)
```

[23]:
```
[0, 1, 2, 3, 4]
```

[27]:
```
s = pd.Series([1, 2, 3, 4])
print(s)
s = pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])
print(s)
```

```
0    1
1    2
2    3
3    4
```

```
dtype: int64
a    1
b    2
c    3
d    4
dtype: int64
```

[28]:
```
s = pd.Series([1, 2, 3, 4], index=['a', 'b', 'c'])
print(s)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/Users/yigalweinberger/Documents/INT/DS bootcamp/Intro to Pandas.ipynb Cell 8␣
 ↪line 1
----> <a href='vscode-notebook-cell:/Users/yigalweinberger/Documents/INT/
 ↪DS%20bootcamp/Intro%20to%20Pandas.ipynb#X20sZmlsZQ%3D%3D?line=0'>1</a> s = pd
 ↪Series([1, 2, 3, 4], index=['a', 'b', 'c'])
      <a href='vscode-notebook-cell:/Users/yigalweinberger/Documents/INT/
 ↪DS%20bootcamp/Intro%20to%20Pandas.ipynb#X20sZmlsZQ%3D%3D?line=1'>2</a> print(s)

File ~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/series.py:461, in␣
 ↪Series.__init__(self, data, index, dtype, name, copy, fastpath)
    459     index = default_index(len(data))
    460 elif is_list_like(data):
--> 461     com.require_length_match(data, index)
    463 # create/copy the manager
    464 if isinstance(data, (SingleBlockManager, SingleArrayManager)):

File ~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/common.py:571, in␣
 ↪require_length_match(data, index)
    567 """
    568 Check the length of data matches the length of the index.
    569 """
    570 if len(data) != len(index):
--> 571     raise ValueError(
    572         "Length of values "
    573         f"({len(data)}) "
    574         "does not match length of index "
    575         f"({len(index)})"
    576     )

ValueError: Length of values (4) does not match length of index (3)
```

[31]:
```
data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 35],
    'City': ['New York', 'San Francisco', 'Los Angeles']
```

```
}
df = pd.DataFrame(data)
df
```

[31]:

	Name	Age	City
0	Alice	25	New York
1	Bob	30	San Francisco
2	Charlie	35	Los Angeles

[84]:
```
my_list = [
    {'Name': 'Yigal',
    'Age': 25,
    'City': 'New York'}
]
my_list.append({'Name': 'Bob', 'Age': 30, 'City': 'San Francisco'})
my_list.append({'Name': 'Charlie', 'Age': 35, 'City': 'Los Angeles'})
df = pd.DataFrame(my_list)
df
```

[84]:

	Name	Age	City
0	Yigal	25	New York
1	Bob	30	San Francisco
2	Charlie	35	Los Angeles

0.0.2 Part 2: Data Manipulation with Pandas

Pandas provides a plethora of functions to manipulate, filter, and aggregate data.

- **Selection**: You can select specific columns or rows based on conditions.

```
# Selecting the 'Name' column
print(df['Name'])

# Selecting rows where age is greater than 28
print(df[df['Age'] > 28])
```

- **Aggregation**: Functions like `mean`, `sum`, and `count` can be applied.

```
# Average age
print(df['Age'].mean())
```

- **Modification**: You can add new columns or modify existing ones.

```
df['Salary'] = [50000, 60000, 70000]
print(df)
```

[41]:
```
type(df['Name'])
```

[41]:
```
pandas.core.series.Series
```

[42]:
```
df['Age']
```

[42]:
```
0    25
1    30
2    35
Name: Age, dtype: int64
```

[44]:
```
df[df['Age'] > 26]
```

[44]:
```
      Name  Age           City
1      Bob   30  San Francisco
2  Charlie   35    Los Angeles
```

[52]:
```
# df > 26
i_am_a_series = df['Age'] > 26
print(type(i_am_a_series))
df[i_am_a_series]
```

```
<class 'pandas.core.series.Series'>
```

[52]:
```
0    25
1    30
2    35
Name: Age, dtype: int64
```

[59]:
```
my_string = 'Age'
return_value_from_string = df[my_string]
print(type(return_value_from_string))
return_value_from_series = df[df['Age'] > 26]
# return_value_from_series = df[i_am_a_series]
return_value_from_series
# print(type(return_value_from_series))
```

```
<class 'pandas.core.series.Series'>
```

[59]:
```
      Name  Age           City
1      Bob   30  San Francisco
2  Charlie   35    Los Angeles
```

- **Aggregation**: Functions like `mean`, `sum`, and `count` can be applied.

```
# Average age
print(df['Age'].mean())
```

[62]:
```
print(my_series.mean())
print(df['Age'])
print(df['Age'].mean())
```

```
3.0
0    25
1    30
2    35
```

```
Name: Age, dtype: int64
30.0
```

[82]:
```
df['Salary'] = [500000, 60000, 70000]
print(df)
df['Another Salary'] = [500000, 60000, 70000]
print(df)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/Users/yigalweinberger/Documents/INT/DS bootcamp/Intro to Pandas.ipynb Cell 19␣
 ↪line 1
----> <a href='vscode-notebook-cell:/Users/yigalweinberger/Documents/INT/
 ↪DS%20bootcamp/Intro%20to%20Pandas.ipynb#X33sZmlsZQ%3D%3D?line=0'>1</a>␣
 ↪df['Salary'] = [500000, 60000, 70000]
      <a href='vscode-notebook-cell:/Users/yigalweinberger/Documents/INT/
 ↪DS%20bootcamp/Intro%20to%20Pandas.ipynb#X33sZmlsZQ%3D%3D?line=1'>2</a>␣
 ↪print(df)
      <a href='vscode-notebook-cell:/Users/yigalweinberger/Documents/INT/
 ↪DS%20bootcamp/Intro%20to%20Pandas.ipynb#X33sZmlsZQ%3D%3D?line=2'>3</a>␣
 ↪df['Another Salary'] = [500000, 60000, 70000]

File ~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/frame.py:3980, in␣
 ↪DataFrame.__setitem__(self, key, value)
   3977     self._setitem_array([key], value)
   3978 else:
   3979     # set column
-> 3980     self._set_item(key, value)

File ~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/frame.py:4174, in␣
 ↪DataFrame._set_item(self, key, value)
   4164 def _set_item(self, key, value) -> None:
   4165     """
   4166     Add series to DataFrame in specified column.
   4167
   (…)
   4172     ensure homogeneity.
   4173     """
-> 4174     value = self._sanitize_column(value)
   4176     if (
   4177         key in self.columns
   4178         and value.ndim == 1
   4179         and not is_extension_array_dtype(value)
   4180     ):
   4181         # broadcast across multiple columns if necessary
   4182         if not self.columns.is_unique or isinstance(self.columns,␣
 ↪MultiIndex):
```

```
File ~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/frame.py:4915, in␣
 ↪DataFrame._sanitize_column(self, value)
   4912     return _reindex_for_setitem(Series(value), self.index)
   4914 if is_list_like(value):
-> 4915     com.require_length_match(value, self.index)
   4916 return sanitize_array(value, self.index, copy=True, allow_2d=True)

File ~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/common.py:571, in␣
 ↪require_length_match(data, index)
    567 """
    568 Check the length of data matches the length of the index.
    569 """
    570 if len(data) != len(index):
--> 571     raise ValueError(
    572         "Length of values "
    573         f"({len(data)}) "
    574         "does not match length of index "
    575         f"({len(index)})"
    576     )

ValueError: Length of values (3) does not match length of index (623989)
```

=### Part 3: Reading and Writing Data with Pandas

Pandas supports various file formats, making data I/O seamless and straightforward.

- **Reading Data**: You can read data from CSV, Excel, SQL databases, and more.

```
# Reading from a CSV file
data_from_csv = pd.read_csv('path_to_file.csv')
```

- **Writing Data**: Similarly, you can write your data back to various formats.

```
# Writing to an Excel file
df.to_excel('output_file.xlsx', index=False)
```

[81]:
```
df
```

[81]:

	Date	Open	High	Low	Close	Volume	\
0	2018-09-04	173.705350	173.870349	171.230328	172.096588	2108500.0	
1	2018-09-05	171.840894	174.827431	171.065397	174.761429	1951500.0	
2	2018-09-06	174.901636	176.031892	174.134385	175.512146	2363500.0	
3	2018-09-07	175.099618	175.701870	173.226847	175.371872	2739200.0	
4	2018-09-10	176.287655	178.506928	175.611156	175.668900	2399700.0	
…	…	…	…	…	…	…	
623984	2023-08-25	183.080002	185.009995	181.440002	182.820007	879700.0	
623985	2023-08-28	184.509995	187.929993	184.300003	186.979996	1546900.0	
623986	2023-08-29	187.300003	192.869995	186.860001	192.770004	2423800.0	
623987	2023-08-30	192.699997	194.369995	191.820007	192.699997	1673500.0	
623988	2023-08-31	192.259995	193.279999	190.470001	190.509995	1893000.0	

```
       stock  Adj Close
0        MMM        NaN
1        MMM        NaN
2        MMM        NaN
3        MMM        NaN
4        MMM        NaN
…        …          …
623984   ZTS        NaN
623985   ZTS        NaN
623986   ZTS        NaN
623987   ZTS        NaN
623988   ZTS        NaN

[623989 rows x 8 columns]
```

[88]:
```
df = pd.read_csv('my_data.csv')
df = pd.read_csv('/Users/yigalweinberger/Documents/INT/DS bootcamp/my_data.csv')
```

[80]:
```
df['stock'].value_counts()
```

[80]:
```
MMM     1257
MSI     1257
NTRS    1257
NSC     1257
NDSN    1257
        …
ABNB     685
OGN      579
CEG      407
GEHC     178
KVUE      83
Name: stock, Length: 501, dtype: int64
```

[86]:
```
df['Salary'] = [500000, 60000, 70000, 80000]
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/Users/yigalweinberger/Documents/INT/DS bootcamp/Intro to Pandas.ipynb Cell 24␣
 ↪line 1
----> <a href='vscode-notebook-cell:/Users/yigalweinberger/Documents/INT/
 ↪DS%20bootcamp/Intro%20to%20Pandas.ipynb#X40sZmlsZQ%3D%3D?line=0'>1</a>␣
 ↪df['Salary'] = [500000, 60000, 70000, 80000]

File ~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/frame.py:3980, in␣
 ↪DataFrame.__setitem__(self, key, value)
   3977     self._setitem_array([key], value)
   3978 else:
```

```
   3979     # set column
-> 3980     self._set_item(key, value)

File ~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/frame.py:4174, in␣
 ↪DataFrame._set_item(self, key, value)
   4164 def _set_item(self, key, value) -> None:
   4165     """
   4166     Add series to DataFrame in specified column.
   4167
   (…)
   4172     ensure homogeneity.
   4173     """
-> 4174     value = self._sanitize_column(value)
   4176     if (
   4177         key in self.columns
   4178         and value.ndim == 1
   4179         and not is_extension_array_dtype(value)
   4180     ):
   4181         # broadcast across multiple columns if necessary
   4182         if not self.columns.is_unique or isinstance(self.columns,␣
 ↪MultiIndex):

File ~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/frame.py:4915, in␣
 ↪DataFrame._sanitize_column(self, value)
   4912     return _reindex_for_setitem(Series(value), self.index)
   4914 if is_list_like(value):
-> 4915     com.require_length_match(value, self.index)
   4916 return sanitize_array(value, self.index, copy=True, allow_2d=True)

File ~/opt/anaconda3/lib/python3.8/site-packages/pandas/core/common.py:571, in␣
 ↪require_length_match(data, index)
    567 """
    568 Check the length of data matches the length of the index.
    569 """
    570 if len(data) != len(index):
--> 571     raise ValueError(
    572         "Length of values "
    573         f"({len(data)}) "
    574         "does not match length of index "
    575         f"({len(index)})"
    576     )

ValueError: Length of values (4) does not match length of index (3)
```

[90]:
```
# df.to_csv('my_new_data.csv')
df[:100].to_csv('my_new_data.csv')
```