**Lesson 4**

In this lesson were going to go back to the basics. We will be working with a small data set so that you can easily understand what I am trying to explain. We will be adding columns, deleting columns, and slicing the data many different ways. Enjoy!

In [1]:

*# Import libraries*

**import** **pandas** **as** **pd**

**import** **sys**

In [2]:

print 'Python version ' + sys.version

print 'Pandas version: ' + pd.\_\_version\_\_

Python version 2.7.5 |Anaconda 2.1.0 (64-bit)| (default, Jul 1 2013, 12:37:52) [MSC v.1500 64 bit (AMD64)]

Pandas version: 0.15.2

In [3]:

*# Our small data set*

d = [0,1,2,3,4,5,6,7,8,9]

*# Create dataframe*

df = pd.DataFrame(d)

df

Out[3]:

|  |  |
| --- | --- |
|  | **0** |
| **0** | 0 |
| **1** | 1 |
| **2** | 2 |
| **3** | 3 |
| **4** | 4 |
| **5** | 5 |
| **6** | 6 |
| **7** | 7 |
| **8** | 8 |
| **9** | 9 |

In [4]:

*# Lets change the name of the column*

df.columns = ['Rev']

df

Out[4]:

|  |  |
| --- | --- |
|  | **Rev** |
| **0** | 0 |
| **1** | 1 |
| **2** | 2 |
| **3** | 3 |
| **4** | 4 |
| **5** | 5 |
| **6** | 6 |
| **7** | 7 |
| **8** | 8 |
| **9** | 9 |

In [5]:

*# Lets add a column*

df['NewCol'] = 5

df

Out[5]:

|  |  |  |
| --- | --- | --- |
|  | **Rev** | **NewCol** |
| **0** | 0 | 5 |
| **1** | 1 | 5 |
| **2** | 2 | 5 |
| **3** | 3 | 5 |
| **4** | 4 | 5 |
| **5** | 5 | 5 |
| **6** | 6 | 5 |
| **7** | 7 | 5 |
| **8** | 8 | 5 |
| **9** | 9 | 5 |

In [6]:

*# Lets modify our new column*

df['NewCol'] = df['NewCol'] + 1

df

Out[6]:

|  |  |  |
| --- | --- | --- |
|  | **Rev** | **NewCol** |
| **0** | 0 | 6 |
| **1** | 1 | 6 |
| **2** | 2 | 6 |
| **3** | 3 | 6 |
| **4** | 4 | 6 |
| **5** | 5 | 6 |
| **6** | 6 | 6 |
| **7** | 7 | 6 |
| **8** | 8 | 6 |
| **9** | 9 | 6 |

In [7]:

*# We can delete columns*

**del** df['NewCol']

df

Out[7]:

|  |  |
| --- | --- |
|  | **Rev** |
| **0** | 0 |
| **1** | 1 |
| **2** | 2 |
| **3** | 3 |
| **4** | 4 |
| **5** | 5 |
| **6** | 6 |
| **7** | 7 |
| **8** | 8 |
| **9** | 9 |

In [8]:

*# Lets add a couple of columns*

df['test'] = 3

df['col'] = df['Rev']

df

Out[8]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Rev** | **test** | **col** |
| **0** | 0 | 3 | 0 |
| **1** | 1 | 3 | 1 |
| **2** | 2 | 3 | 2 |
| **3** | 3 | 3 | 3 |
| **4** | 4 | 3 | 4 |
| **5** | 5 | 3 | 5 |
| **6** | 6 | 3 | 6 |
| **7** | 7 | 3 | 7 |
| **8** | 8 | 3 | 8 |
| **9** | 9 | 3 | 9 |

In [9]:

*# If we wanted, we could change the name of the index*

i = ['a','b','c','d','e','f','g','h','i','j']

df.index = i

df

Out[9]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Rev** | **test** | **col** |
| **a** | 0 | 3 | 0 |
| **b** | 1 | 3 | 1 |
| **c** | 2 | 3 | 2 |
| **d** | 3 | 3 | 3 |
| **e** | 4 | 3 | 4 |
| **f** | 5 | 3 | 5 |
| **g** | 6 | 3 | 6 |
| **h** | 7 | 3 | 7 |
| **i** | 8 | 3 | 8 |
| **j** | 9 | 3 | 9 |

We can now start to select pieces of the dataframe using ***loc***.

In [10]:

df.loc['a']

Out[10]:

Rev 0

test 3

col 0

Name: a, dtype: int64

In [11]:

*# df.loc[inclusive:inclusive]*

df.loc['a':'d']

Out[11]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Rev** | **test** | **col** |
| **a** | 0 | 3 | 0 |
| **b** | 1 | 3 | 1 |
| **c** | 2 | 3 | 2 |
| **d** | 3 | 3 | 3 |

In [12]:

*# df.iloc[inclusive:exclusive]*

*# Note: .iloc is strictly integer position based. It is available from [version 0.11.0] (http://pandas.pydata.org/pandas-docs/stable/whatsnew.html#v0-11-0-april-22-2013)*

df.iloc[0:3]

Out[12]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Rev** | **test** | **col** |
| **a** | 0 | 3 | 0 |
| **b** | 1 | 3 | 1 |
| **c** | 2 | 3 | 2 |

We can also select using the column name.

In [13]:

df['Rev']

Out[13]:

a 0

b 1

c 2

d 3

e 4

f 5

g 6

h 7

i 8

j 9

Name: Rev, dtype: int64

In [14]:

df[['Rev', 'test']]

Out[14]:

|  |  |  |
| --- | --- | --- |
|  | **Rev** | **test** |
| **a** | 0 | 3 |
| **b** | 1 | 3 |
| **c** | 2 | 3 |
| **d** | 3 | 3 |
| **e** | 4 | 3 |
| **f** | 5 | 3 |
| **g** | 6 | 3 |
| **h** | 7 | 3 |
| **i** | 8 | 3 |
| **j** | 9 | 3 |

In [15]:

*# df['ColumnName'][inclusive:exclusive]*

df['Rev'][0:3]

Out[15]:

a 0

b 1

c 2

Name: Rev, dtype: int64

In [16]:

df['col'][5:]

Out[16]:

f 5

g 6

h 7

i 8

j 9

Name: col, dtype: int64

In [17]:

df[['col', 'test']][:3]

Out[17]:

|  |  |  |
| --- | --- | --- |
|  | **col** | **test** |
| **a** | 0 | 3 |
| **b** | 1 | 3 |
| **c** | 2 | 3 |

There is also some handy function to select the top and bottom records of a dataframe.

In [18]:

*# Select top N number of records (default = 5)*

df.head()

Out[18]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Rev** | **test** | **col** |
| **a** | 0 | 3 | 0 |
| **b** | 1 | 3 | 1 |
| **c** | 2 | 3 | 2 |
| **d** | 3 | 3 | 3 |
| **e** | 4 | 3 | 4 |

In [19]:

*# Select bottom N number of records (default = 5)*

df.tail()

Out[19]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Rev** | **test** | **col** |
| **f** | 5 | 3 | 5 |
| **g** | 6 | 3 | 6 |
| **h** | 7 | 3 | 7 |
| **i** | 8 | 3 | 8 |
| **j** | 9 | 3 | 9 |