38.DataFrame

38.1 创建方法

In [2]:
```
import numpy as np
import pandas as pd
df1=pd.DataFrame(np.arange(10).reshape(2,5))
df1
```

Out[2]:

	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9

In [3]:
```
df2 = pd.read_csv('bc_data.csv')
df2.shape
```

Out[3]: (569, 32)

In [4]:
```
df2=df2[["id","diagnosis","area_mean"]]
df2.head()
```

Out[4]:

	id	diagnosis	area_mean
0	842302	M	1001.0
1	842517	M	1326.0
2	84300903	M	1203.0
3	84348301	M	386.1
4	84358402	M	1297.0

38.2DataFrame中的行/列

In [5]:
```
df2.index
```

Out[5]: RangeIndex(start=0, stop=569, step=1)

In [6]:
```
df2.index.size
```

Out[6]: 569

In [7]:
```
df2.columns
```

Out[7]: Index(['id', 'diagnosis', 'area_mean'], dtype='object')

In [8]:
```
df2.columns.size
```

Out[8]: 3

In [9]:
```
df2.shape
```

Out[9]:
```
(569, 3)
```

In [10]:
```
print("行数为:", df2.shape[0])
print("列数为:", df2.shape[1])
```

```
行数为: 569
列数为: 3
```

38.3 访问元素的方法

In [13]:
```
df2["id"].head()
```

Out[13]:
```
0     842302
1     842517
2    84300903
3    84348301
4    84358402
Name: id, dtype: int64
```

In [14]:
```
df2.id.head()
```

Out[14]:
```
0     842302
1     842517
2    84300903
3    84348301
4    84358402
Name: id, dtype: int64
```

In [15]:
```
df2["id"][2]
```

Out[15]:
```
84300903
```

In [16]:
```
df2.id[2]
```

Out[16]:
```
84300903
```

In [17]:
```
df2["id"][[2,4]]
```

Out[17]:
```
2    84300903
4    84358402
Name: id, dtype: int64
```

In [18]:
```
df2.loc[1,"id"]
```

Out[18]:
```
842517
```

In [19]:
```
df2.iloc[1,0]
```

Out[19]:
```
842517
```

In [20]:
```
df2.ix[[1],["id"]]
```

```
C:\Anaconda\lib\site-packages\ipykernel_launcher.py:3: DeprecationWarnin
g:
.ix is deprecated. Please use
.loc for label based indexing or
```

```
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-deprecated
  This is separate from the ipykernel package so we can avoid doing impo
rts until
```

Out[20]:

	id
1	842517

In [21]:
```
df2.ix[[1,5],["id"]]
```

Out[21]:

	id
1	842517
5	843786

In [22]:
```
df2.ix[1:5,["id"]]
```

Out[22]:

	id
1	842517
2	84300903
3	84348301
4	84358402
5	843786

In [23]:
```
df2[["area_mean","id"]].head()
```

Out[23]:

	area_mean	id
0	1001.0	842302
1	1326.0	842517
2	1203.0	84300903
3	386.1	84348301
4	1297.0	84358402

38.4 index操作

In [24]:
```
df2.index
```

Out[24]: RangeIndex(start=0, stop=569, step=1)

In [25]:
```
df2.columns
```

Out[25]: Index(['id', 'diagnosis', 'area_mean'], dtype='object')

In [26]:
```
df2["id"].head()
```

Out[26]:
```
0    842302
1    842517
2  84300903
3  84348301
4  84358402
Name: id, dtype: int64
```

In [27]:
```
df2.reindex(index=["1","2","3"],columns=["1","2","3"])
df2.head()
```

Out[27]:

	id	diagnosis	area_mean
0	842302	M	1001.0
1	842517	M	1326.0
2	84300903	M	1203.0
3	84348301	M	386.1
4	84358402	M	1297.0

In [28]:
```
df2.reindex(index=[2,3,1], columns=["diagnosis","id","area_mean"])
```

Out[28]:

	diagnosis	id	area_mean
2	M	84300903	1203.0
3	M	84348301	386.1
1	M	842517	1326.0

In [29]:
```
df3=df2.reindex(index=[2,3,1], columns=["diagnosis","id","area_mean","MyNewColumn"],fill_value=100)
df3
```

Out[29]:

	diagnosis	id	area_mean	MyNewColumn
2	M	84300903	1203.0	100
3	M	84348301	386.1	100
1	M	842517	1326.0	100

38.4 删除或过滤行列

In [30]:
```
import pandas as pd
df2 = pd.read_csv('bc_data.csv')

df2=df2[["id","diagnosis","area_mean"]]
df2.head()
```

Out[30]:

	id	diagnosis	area_mean

0	842302	M	1001.0
1	842517	M	1326.0
2	84300903	M	1203.0
3	84348301	M	386.1
4	84358402	M	1297.0

In [31]:
```
df2.drop([2]).head()
```

Out[31]:

	id	diagnosis	area_mean
0	842302	M	1001.0
1	842517	M	1326.0
3	84348301	M	386.1
4	84358402	M	1297.0
5	843786	M	477.1

In [32]:
```
df2.head()
```

Out[32]:

	id	diagnosis	area_mean
0	842302	M	1001.0
1	842517	M	1326.0
2	84300903	M	1203.0
3	84348301	M	386.1
4	84358402	M	1297.0

In [33]:
```
import pandas as pd
df2 = pd.read_csv('bc_data.csv')
df2=df2[["id","diagnosis","area_mean"]]

df2.drop([3,4], axis=0, inplace=True)
df2.head()
```

Out[33]:

	id	diagnosis	area_mean
0	842302	M	1001.0
1	842517	M	1326.0
2	84300903	M	1203.0
5	843786	M	477.1
6	844359	M	1040.0

In [34]:
```
import pandas as pd
df2 = pd.read_csv('bc_data.csv')
df2=df2[["id","diagnosis","area_mean"]]
df2.drop([3,4], axis=0, inplace=False)
```

```
df2.head()
```

Out[34]:

	id	diagnosis	area_mean
0	842302	M	1001.0
1	842517	M	1326.0
2	84300903	M	1203.0
3	84348301	M	386.1
4	84358402	M	1297.0

In [35]:
```
import pandas as pd
df2 = pd.read_csv('bc_data.csv')
df2=df2[["id","diagnosis","area_mean"]]
del df2["area_mean"]
df2.head()
```

Out[35]:

	id	diagnosis
0	842302	M
1	842517	M
2	84300903	M
3	84348301	M
4	84358402	M

In [36]:
```
import pandas as pd
df2 = pd.read_csv('bc_data.csv')
df2=df2[["id","diagnosis","area_mean"]]
df2.drop(["id","diagnosis"], axis=1, inplace=True)
df2.head()
```

Out[36]:

	area_mean
0	1001.0
1	1326.0
2	1203.0
3	386.1
4	1297.0

In [37]:
```
import pandas as pd
df2 =pd.read_csv('bc_data.csv')

df2=df2[["id","diagnosis","area_mean"]]
df2[df2.area_mean> 1000].head()
```

Out[37]:

	id	diagnosis	area_mean
0	842302	M	1001.0

1	842517	M	1326.0
2	84300903	M	1203.0
4	84358402	M	1297.0
6	844359	M	1040.0

In [38]:
```
df2[df2.area_mean> 1000][["id","diagnosis"]].head()
```

Out[38]:

	id	diagnosis
0	842302	M
1	842517	M
2	84300903	M
4	84358402	M
6	844359	M

38.5 算术运算

In [39]:
```
df4=pd.DataFrame(np.arange(6).reshape(2,3))
df4
```

Out[39]:

	0	1	2
0	0	1	2
1	3	4	5

In [40]:
```
df5=pd.DataFrame(np.arange(10).reshape(2,5))

df5
```

Out[40]:

	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9

In [41]:
```
df4+df5
```

Out[41]:

	0	1	2	3	4
0	0	2	4	NaN	NaN
1	8	10	12	NaN	NaN

In [42]:
```
df6=df4.add(df5,fill_value=10)
df6
```

Out[42]:

	0	1	2	3	4

0	0	2	4	13.0	14.0
1	8	10	12	18.0	19.0

In [43]:
```
s1=pd.Series(np.arange(3))
s1
```

Out[43]:
```
0    0
1    1
2    2
dtype: int32
```

In [44]:
```
df6-s1
```

Out[44]:

	0	1	2	3	4
0	0.0	1.0	2.0	NaN	NaN
1	8.0	9.0	10.0	NaN	NaN

In [45]:
```
df5=pd.DataFrame(np.arange(10).reshape(2,5))
s1=pd.Series(np.arange(3))
df5-s1
```

Out[45]:

	0	1	2	3	4
0	0.0	0.0	0.0	NaN	NaN
1	5.0	5.0	5.0	NaN	NaN

In [46]:
```
df5=pd.DataFrame(np.arange(10).reshape(2,5))
s1=pd.Series(np.arange(3))
df5.sub(s1,axis=1)
```

Out[46]:

	0	1	2	3	4
0	0.0	0.0	0.0	NaN	NaN
1	5.0	5.0	5.0	NaN	NaN

In [47]:
```
df5=pd.DataFrame(np.arange(10).reshape(2,5))
s1=pd.Series(np.arange(3))
df5.sub(s1,axis=0)
```

Out[47]:

	0	1	2	3	4
0	0.0	1.0	2.0	3.0	4.0
1	4.0	5.0	6.0	7.0	8.0
2	NaN	NaN	NaN	NaN	NaN

In [48]:
```
df7=pd.DataFrame(np.arange(20).reshape(4,5))
df7
```

Out[48]:

	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9
2	10	11	12	13	14
3	15	16	17	18	19

In [49]:
```
df7+2
```

Out[49]:

	0	1	2	3	4
0	2	3	4	5	6
1	7	8	9	10	11
2	12	13	14	15	16
3	17	18	19	20	21

In [50]:
```
print(df7)
print("df7.cumsum=",df7.cumsum())
```

```
   0  1  2  3  4
0  0  1  2  3  4
1  5  6  7  8  9
2 10 11 12 13 14
3 15 16 17 18 19
df7.cumsum=    0  1  2  3  4
0  0  1  2  3  4
1  5  7  9 11 13
2 15 18 21 24 27
3 30 34 38 42 46
```

In [51]:
```
df7
```

Out[51]:

	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9
2	10	11	12	13	14
3	15	16	17	18	19

In [52]:
```
df7.rolling(2).sum()
```

Out[52]:

	0	1	2	3	4
0	NaN	NaN	NaN	NaN	NaN
1	5.0	7.0	9.0	11.0	13.0
2	15.0	17.0	19.0	21.0	23.0
3	25.0	27.0	29.0	31.0	33.0

In [53]:
```
df7.rolling(2,axis=1).sum()
```

Out[53]:

	0	1	2	3	4
0	NaN	1.0	3.0	5.0	7.0
1	NaN	11.0	13.0	15.0	17.0
2	NaN	21.0	23.0	25.0	27.0
3	NaN	31.0	33.0	35.0	37.0

In [54]:
```
df7.cov()
```

Out[54]:

	0	1	2	3	4
0	41.666667	41.666667	41.666667	41.666667	41.666667
1	41.666667	41.666667	41.666667	41.666667	41.666667
2	41.666667	41.666667	41.666667	41.666667	41.666667
3	41.666667	41.666667	41.666667	41.666667	41.666667
4	41.666667	41.666667	41.666667	41.666667	41.666667

In [55]:
```
df7.corr()
```

Out[55]:

	0	1	2	3	4
0	1.0	1.0	1.0	1.0	1.0
1	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0
4	1.0	1.0	1.0	1.0	1.0

In [56]:
```
import pandas as pd
df2 = pd.read_csv('bc_data.csv')

df2=df2[["id","diagnosis","area_mean"]][2:5]
df2.T
```

Out[56]:

	2	3	4
id	84300903	84348301	84358402
diagnosis	M	M	M
area_mean	1203	386.1	1297

38.6 大小比较运算

In [57]:
```
print(df6)
```