

DataFrame

Create

```
df=pd.DataFrame([Series1,Series2],index=['index1','index2']) * Series have same index  
df=pd.DataFrame([list1,list2],index=['index1','index2'],columns=['col1','col2'])  
df=pd.DataFrame({'col1':10,'col2':20},index=['idx2','idx2'])  
df=se.to_frame('series')
```

Querying

```
df.index/columns df.index[0]  
df.index.get_loc('idx1')  
df.loc df.loc['index1']  
df.loc[['index1','index3']]  
df.loc[:, 'col1']  
df.loc['index1':'index3', 'col1':'col3']  
df.iloc df.iloc[1]  
df.iloc[:, 1:3]  
df.iloc[ [1,3] ]  
df.dtypes df.col.dtype  
df.index.dtype  
functions df['col'].unique()  
df['col'].nunique()/df.nunique()  
df['col'].value_counts()  
emptylist = [i for i in df['col'] if i not in df2['col2']]
```

Edit

```
delete df.drop(['col1','col2'], axis=1)  
df.drop(['idx1','idx2'],axis=0)  
del df['only one col']  
df.dropna (how='any') if contains nan, drop  
(how='all') if all nan, drop  
(axis=1) column drop nan  
Data Add df.fillna(0)  
df.append(df2)  
df.insert(2,'colnew2',new2value)  
df.concat([df1,df2]) **df1/df2: df1,df2 same columns  
df.concat([df1,df2],axis=1) **df1/df2: df1,df2, different columns, same index  
pd.merge(df1,df2)  
pd.join(df1,df2)  
edit df['newcol']=df['col1']/df['col2']  
df.where(df['col1']>0,other=nan) *return same shape df with cond=T  
pd.melt(data, id_vars=[''],value_vars=['',''])  
pd.cut(data, [0,18,90])  
Index/Columns edit df.reset_index()  
df.set_index('col1')  
df.rename df.rename(lambda x:x**2) *rename index numbers to square  
df.rename(index={}, columns={'col1':'newcol1','col2':'newcol2'})  
df.index.name='newname'  
df.droplevel(level=0)  
df.reindex(columns=collist, fill_value=0)  
df.get_dummies()  
str method:df.index.str str.lower()/upper()  
str.replace('from','to')  
str.split('char',expand=True) *split one col to multiple col based on char  
str.contains('char',regex=False) *retain boolean mask  
str.extract(regex,regex=True)
```

Time Series

```
pd.to_datetime('Data')  
datetime.date.dt.year/month/day/hour/etc.  
drange=pd.date_range('20190101',periods=14, freq='D')/('start','end',freq='')  
df.resample("M").sum() *resample by month  
find missing date fakedf = pd.DataFrame(index = pd.date_range(start='',end='',freq=''))  
pd.merge(data,fakedf,right_index=True,how='outer')  
datetimeindex.strftime  
datetime to epoch seconds pd.to_datetime(epochsecs,unit='s') + pd.DateOffset(hours=24)  
datetimestring.values.astype(np.int64) // 10**9
```

Iterations

```
for colname,(data in colname) in df.iteritems():  
for indexname,(data in indexname) in df.iterrows():
```

Sorting

```
df.sort_index(axis=0) *sort based on index labels  
df.sort_index(axis=1) *sort based on column labels (reorder columns)  
df.sort_values("col1",axis=0) *sort based on the values for col1
```

Grouping

```
gp = df.groupby(['col1','col2'])  
gp.first() first element in each group  
gp.size() # elements in each group  
gp.count() # return a df, elements in each col in each group  
gp.get_group('col1val1','col2val1') the group col1=colval1  
gp[['col3','col4']].sum()/mean()  
gp.agg(['func1',self.Def_func])  
df.pivot_table pd.pivot_table(df, index='col1', columns='col2', values='col3',  
aggfunc= ['mean','count'], margins=True, margins_name='Total')
```

Categorical Data

```
df['catecol1']=df['col1'].astype('category')  
df['catecol1'].cat.categories=['rename1','rename2','rename3']  
df['catecol1'].cat.set_categories=['rename1','rename2','rename3']  
df.get_dummies()
```

Display

```
with pd.option_context('display.max_colwidth', 220):  
df
```

Plot

```
plt.figure() *start a figure()  
plt.style.available plt.style.use('stylename')  
Plot data df.plot() *default is x=index, y=all columns  
df.plot(x='col1',y='col2') *plot one column against another  
df.plot(data,kind='bar/line/scatter/box/hist/kde',c='col3',  
style='mark/line',alpha=0.8,subplot=True,figsize=(xsize,ysize) *same as df.plot.bar()  
pd.tools.plotting .scatter_matrix(df)  
.parallel_coordinates(df, 'colname')  
ax=plt.gca() for other properties editings
```