

Python Cheat Sheet

Pandas



Reshaping Data

Pivot

```
>>> df3= df2.pivot(index='Date',
columns='Type',
values='Value')
```

Spread rows into columns

	Date	Type	Value
0	2016-03-01	a	11.43
1	2016-03-02	b	2
2	2016-03-02	c	13.03
3	2016-03-01	a	1
4	2016-03-01	a	20.78
5	2016-03-03	c	4

Type	a	b	c
Date			
2016-03-01	11.43	NaN	20.78
2016-03-02	2	13.03	4 NaN
2016-03-03	1.303	1 NaN	20.78

Pivot Table

6

```
values= 'Value',
index='Date',
columns='Type')
```

Spread rows into columns

Stack / Unstack

```
>>> stacked = df5.stack()
```

```
>>> stacked.unstack()
```

Pivot a level of column labels
Pivot a level of index

	0	1
0	1.5	0.233482
1	0.184713	0.390959
2	0.438522	0.421401

Unstacked

	0	1
0	1.5	0.233482
1	0.184713	0.390959
2	0.438522	0.421401

Stacked

Melt

```
>>> pd.melt(df2,
            id_vars=["Date"],
            value_name="Observations")
```

Gather columns into rows

	Date	Type	Value
0	2016-03-01	a	11.43
1	2016-03-02	b	2
2	2016-03-02	c	13.03
3	2016-03-01	a	1
4	2016-03-01	a	20.78
5	2016-03-03	c	4

	Date	Type	Observations
0	2016-03-01	a	11.43
1	2016-03-02	b	2
2	2016-03-02	c	13.03
3	2016-03-01	a	1
4	2016-03-01	a	20.78
5	2016-03-03	c	4

Iteration

(Column-index, Series)

Also See NumPy Arrays

Selecting

```
>>> df3.loc[:, (df3>1).any()]
>>> df3.loc[:, (df3>1).all()]
>>> df3.loc[:, df3.isnull().any()]
>>> df3.loc[:, df3.notnull().all()]
```

Select cols with any vals >1
Select cols with vals > 1
Select cols with NaN
Select cols without NaN

Indexing With isin

```
>>> df[(df.Country.isin(df2.Type))]
>>> df3.filter(items=["a", "b"])
>>> df.select(lambda x: not x%5)
```

Find same elements
Filter on values
Select specific elements

Where

```
>>> s.where(s > 0)
```

Subset the data

Query

```
>>> df6.query('second > first')
```

Query DataFrame

Setting/Resetting Index

```
>>> df.set_index('Country')
```

Set the index

```
>>> df4 = df.reset_index()
>>> df = df.rename(index=str,
                  columns="Country":"cntry",
                  "Capital":"cptl",
                  "Population":"ppltn")
```

Reset the index
Rename DataFrame

Reindexing

```
>>> s2 = s.reindex(['a','c','d','e','b'])
```

Forward Filling

```
>>> df.reindex(range(4),
               method='ffill')
Country Capital Population
0 Belgium Brussels 11190846
```

Backward Filling

```
>>> s3 = s.reindex(range(5),
                   method='bfill')
```

MultiIndexing

```
>>> arrays = [np.array([1,2,3]),
              np.array([5,4,3])]
>>> df5 = pd.DataFrame(np.random.rand(3, 2), index=arrays)
>>> tuples = list(zip(*arrays))
>>> index = pd.MultiIndex.from_tuples(tuples,
                                   names=['first', 'second'])
>>> df6 = pd.DataFrame(np.random.rand(3, 2), index=index)
>>> df2.set_index(['Date', 'Type'])
```

Duplicate Data

```
>>> s3.unique()
```

Return unique values

```
>>> df2.duplicated('Type')
```

Check duplicates
Drop duplicates

```
>>> df.index.duplicated()
```

Check index duplicates

Grouping Data

Aggregation

```
>>> df2.groupby(by=['Date', 'Type']).mean()
>>> df4.groupby(level=0).sum()
>>> df4.groupby(level=0).agg({'a':lambda x:sum(x)/len(x),
                             'b': np.sum})
```

Transformation

```
>>> df4.groupby(level=0).transform(customSum)
```

```
>>> df.dropna()
```

Drop NaN values

Combining Data

data1

X1	X2
a	11.43
b	2
c	1.303

data2

X1	X3
a	20.78
b	NaN
d	20.78

Merge

```
>>> pd.merge(data1,
             data2,
             how='left',
             on='X1')
```

X1	X2	X3
a	11.43	20.78
b	2	4 NaN
c	1.303	NaN

```
>>> pd.merge(data1,
             data2,
             how='right',
             on='X1')
```

X1	X2	X3
a	11.43	20.78
b	2	4 NaN
d	1.303	20.78

```
>>> pd.merge(data1,
             data2,
             how='inner', on='X1')
```

X1	X2	X3
a	11.43	20.78
b	2	4 NaN

```
>>> df.iterrows()
```

(Row-index, pairs)

```
>>> pd.merge(data1,
data2,
             how='outer',
             on='X1')
```

a	11.43	20.78
b	2	4 NaN
c	1.303	NaN
d	99.90	20.78
	6 NaN	4

Join

```
>>> data1.join(data2, how='right')
```

Concatenate

Vertical

```
>>> s.append(s2)
```

Horizontal/Vertical

```
>>> pd.concat([s,s2],axis=1, keys=['One','Two'])
```

```
>>> pd.concat([data1, data2], axis=1, join='inner')
```

Dates

```
>>> df2['Date']= pd.to_datetime(df2['Date'])
>>> df2['Date']= pd.date_range('2000-1-1', periods=6,
freq='M')
>>> dates = [datetime(2012,5,1), datetime(2012,5,2)]
>>> index = pd.DatetimeIndex(dates)
>>> index = pd.date_range(datetime(2012,2,1), end, freq='BM')
```

Visualization

Also see Matplotlib

```
>>> import matplotlib.pyplot as plt
>>> s.plot()>>> df2.plot()
>>> plt.show()>>> plt.show()
```

