DATA WRANGLING

```
import pandas as pd
import numpy as np
data = pd.DataFrame(np.arange(6).reshape((2, 3)),
index=pd.Index(['Shoes', 'Dress'], name='Items'),
columns=pd.Index(['one', 'two', 'three'],
name='number'))
data
```

number	one	two	three
Items			
Shoes	0	1	2
Dress	3	4	5

```
result = data.stack()
print(result)
result.unstack()
```

```
Items  number
Shoes  one       0
       two       1
       three     2
Dress  one       3
       two       4
       three     5
dtype: int64
```

number	one	two	three
Items			
Shoes	0	1	2
Dress	3	4	5

```
df = pd.DataFrame({'Items': ['Dress', 'Shoes', 'Electronics'],
'A': [1, 2, 3],
'B': [8, 5, 7],
'C': [6, 8, 10]})
df
```

	Items	A	B	C
0	Dress	1	8	6

```
import pandas as pd

# Assign data
data = {'Items': ['Electronics', 'Shoes', 'Dress',],
                'Stock': [17, 19, 18,],
                        'Cost': [10000, 8000, 'NaN']}
df = pd.DataFrame(data)
df
```

	Items	Stock	Cost
0	Electronics	17	10000
1	Shoes	19	8000
2	Dress	18	NaN

```
#fill the missing values
c = avg = 0
for ele in df['Cost']:
    if str(ele).isnumeric():
        c += 1
        avg += ele
avg /= c

# Replace missing values
df = df.replace(to_replace="NaN",
                value=avg)
df
```

	Items	Stock	Cost
0	Electronics	17	10000.0
1	Shoes	19	8000.0
2	Dress	18	9000.0

```
import pandas as pd
import numpy as np
String_data = pd.Series(['Dress', 'Shoes', np.nan, 'Electronics'])
String_data
```

```
    0          Dress
    1          Shoes
```

```
     2            NaN
     3    Electronics
     dtype: object
```

```
String_data.isnull()
```

```
     0    False
     1    False
     2     True
     3    False
     dtype: bool
```

```
String_data[0] = None
String_data.isnull()
```

```
     0     True
     1    False
     2     True
     3    False
     dtype: bool
```

```
from numpy import nan as NA
data = pd.Series([1, NA, 3.5, NA, 7])
data.dropna()
```

```
     0    1.0
     2    3.5
     4    7.0
     dtype: float64
```

```
data = pd.DataFrame([[1.8, 6.5, 2.4], [1., NA, NA],
 ....: [NA, NA, NA], [NA, 6.5, 4.3]])
cleaned = data.dropna()
data
```

	0	1	2
0	1.8	6.5	2.4
1	1.0	NaN	NaN
2	NaN	NaN	NaN
3	NaN	6.5	4.3

```
data[4] = NA
data
```

	0	1	2	4
0	1.8	6.5	2.4	NaN
1	1.0	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN

```
df = pd.DataFrame(np.random.randn(7, 3))
df.iloc[:4, 1] = NA
df.iloc[:2, 2] = NA
df
```

	0	1	2
0	-1.121206	NaN	NaN
1	0.513693	NaN	NaN
2	0.680431	NaN	-0.809259
3	0.341197	NaN	0.005515
4	0.956743	-1.633433	0.580981
5	1.401984	-0.465003	0.664114
6	1.014327	1.620899	-1.330772

```
df.fillna(0)
```

	0	1	2
0	-0.260732	0.000000	0.000000
1	0.526847	0.000000	0.000000
2	0.017321	0.000000	0.531534
3	-0.752444	0.000000	-0.853585
4	-0.281223	-0.836164	0.706368
5	1.349068	1.732472	-0.075129
6	0.717544	-1.127148	0.072644

**DATA TRANSFORMATION **

```
data = pd.DataFrame({'k1': ['one', 'two'] * 3 + ['two'],
                     'k2': [1, 1, 2, 3, 3, 4, 4]})
```

```
print(data)
data.duplicated()
```

```
     k1  k2
0   one   1
1   two   1
2   one   2
3   two   3
4   one   3
5   two   4
6   two   4
0    False
1    False
2    False
3    False
4    False
5    False
6     True
dtype: bool
```

```
 data.drop_duplicates()
```

	k1	k2
0	one	1
1	two	1
2	one	2
3	two	3
4	one	3
5	two	4

```
data = pd.DataFrame({'Dress': ['Shirts', 'Pants', 'Jeans'],
'Qty': [4, 3, 12]})
data
```

	Dress	Qty
0	Shirts	4
1	Pants	3
2	Jeans	12

```
lowercased = data['Dress'].str.lower()
lowercased
```

```
0    shirts
1     pants
```

```
    2     jeans
    Name: Dress, dtype: object
```

```
data = pd.Series([1., -999., 2., -999., -1000., 3.])
print(data)
data.replace(-999, np.nan)
```

```
    0       1.0
    1    -999.0
    2       2.0
    3    -999.0
    4   -1000.0
    5       3.0
    dtype: float64
    0       1.0
    1       NaN
    2       2.0
    3       NaN
    4   -1000.0
    5       3.0
    dtype: float64
```

```
data = pd.DataFrame(np.random.randn(1000, 4))
data.describe()
col = data[2]
col[np.abs(col) > 3]
```

```
    56    -3.264502
    163    3.292536
    763    3.035598
    803   -3.438694
    Name: 2, dtype: float64
```

```
df = pd.DataFrame(np.arange(5 * 4).reshape((5, 4)))
sampler = np.random.permutation(5)
sampler
```

```
    array([2, 4, 1, 3, 0])
```

```
print(df)
df.take(sampler)
```

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

0 1 2 3

```
df = pd.DataFrame({'key': ['b', 'b', 'a', 'c', 'a', 'b'],
'data1': range(6)})
pd.get_dummies(df['key'])
```

	a	b	c
0	0	1	0
1	0	1	0
2	1	0	0
3	0	0	1
4	1	0	0
5	0	1	0