```
import pandas as pd

df=pd.read_csv('survey_data.csv')
```

New Section

```
df.head(5)
```

	Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Sub
0	BDCQ.SF1AA2CA	2016.06	1116.386	NaN	F	Dollars	6	Busi Colle -
1	BDCQ.SF1AA2CA	2016.09	1070.874	NaN	F	Dollars	6	Busi Colle -
2	BDCQ.SF1AA2CA	2016.12	1054.408	NaN	F	Dollars	6	Busi Colle -
3	BDCQ.SF1AA2CA	2017.03	1010.665	NaN	F	Dollars	6	Busi Colle -
4	BDCQ.SF1AA2CA	2017.06	1233.700	NaN	F	Dollars	6	Busi Colle -

Saved successfully!

```
df.describe()
```

	Period	Data_value	Magnitude	Series_title_5
count	5955.000000	5600.000000	5955.0	0.0
mean	2018.973159	4725.778359	6.0	NaN
std	1.827219	6886.030728	0.0	NaN
min	2016.060000	-398.194000	6.0	NaN

```
df.isnull().sum()
```

```
Series_reference       0
Period                 0
Data_value           355
Suppressed          5940
STATUS                 0
UNITS                  0
Magnitude              0
Subject                0
Group                  0
Series_title_1         0
Series_title_2         0
Series_title_3         0
Series_title_4         0
Series_title_5      5955
dtype: int64
```

```
df.shape
```

```
(5955, 14)
```

```
df_10=df[['UNITS','Data_value','Period','Magnitude']].head(10)
```

Saved successfully!

	UNITS	Data_value	Period	Magnitude
0	Dollars	1116.386	2016.06	6
1	Dollars	1070.874	2016.09	6
2	Dollars	1054.408	2016.12	6
3	Dollars	1010.665	2017.03	6
4	Dollars	1233.700	2017.06	6
5	Dollars	1282.436	2017.09	6
6	Dollars	1290.820	2017.12	6
7	Dollars	1412.007	2018.03	6
8	Dollars	1488.055	2018.06	6
9	Dollars	1497.678	2018.09	6

```
df_10['rank']=df_10.groupby('UNITS')['Period'].rank(method='dense',ascending=False)
```

```
df_10
```

	UNITS	Data_value	Period	Magnitude	rank
0	Dollars	1116.386	2016.06	6	10.0
1	Dollars	1070.874	2016.09	6	9.0
2	Dollars	1054.408	2016.12	6	8.0
3	Dollars	1010.665	2017.03	6	7.0
4	Dollars	1233.700	2017.06	6	6.0
5	Dollars	1282.436	2017.09	6	5.0
6	Dollars	1290.820	2017.12	6	4.0
7	Dollars	1412.007	2018.03	6	3.0
8	Dollars	1488.055	2018.06	6	2.0
9	Dollars	1497.678	2018.09	6	1.0

```
df_10[['UNITS','Magnitude']].value_counts()
```

```
UNITS    Magnitude
Dollars  6            10
dtype: int64
```

```
df_10['UNITS'].unique()
```

```
)
```

Saved successfully!

```
df_10['UNITS'].nunique()
```

```
1
```

```
df.columns
```

```
Index(['Series_reference', 'Period', 'Data_value', 'Suppressed', 'STATUS',
       'UNITS', 'Magnitude', 'Subject', 'Group', 'Series_title_1',
       'Series_title_2', 'Series_title_3', 'Series_title_4', 'Series_title_5'],
      dtype='object')
```

```
df.drop(columns=['Suppressed','Series_title_5']).columns #dropping the columns which are m
```

```
Index(['Series_reference', 'Period', 'Data_value', 'STATUS', 'UNITS',
       'Magnitude', 'Subject', 'Group', 'Series_title_1', 'Series_title_2',
       'Series_title_3', 'Series_title_4'],
      dtype='object')
```

```
len(df_10)
```

```
10
```

```
df_10.query("Data_value<Period")# we are comparing te condition
```

	UNITS	Data_value	Period	Magnitude	rank
0	Dollars	1116.386	2016.06	6	10.0
1	Dollars	1070.874	2016.09	6	9.0
2	Dollars	1054.408	2016.12	6	8.0
3	Dollars	1010.665	2017.03	6	7.0
4	Dollars	1233.700	2017.06	6	6.0
5	Dollars	1282.436	2017.09	6	5.0
6	Dollars	1290.820	2017.12	6	4.0
7	Dollars	1412.007	2018.03	6	3.0
8	Dollars	1488.055	2018.06	6	2.0
9	Dollars	1497.678	2018.09	6	1.0

```
df_10.replace(2016.06,1100,inplace=True)#replacing the value
```

```
df_10[df_10['Data_value']>df_10['Period']] # we are comparing te condition
```

	UNITS	Data_value	Period	Magnitude	rank
				6	10.0

Saved successfully!

```
df.iloc[2:5,:5]#it contains row number and column index 5 in this ex. is excluded
```

	Series_reference	Period	Data_value	Suppressed	STATUS
2	BDCQ.SF1AA2CA	2016.12	1054.408	NaN	F
3	BDCQ.SF1AA2CA	2017.03	1010.665	NaN	F
4	BDCQ.SF1AA2CA	2017.06	1233.700	NaN	F

```
df.loc[2:5,['Series_reference','Period','Data_value','Suppressed','STATUS']]#it contain ro
```

	Series_reference	Period	Data_value	Suppressed	STATUS

```
df_10.select_dtypes('float')#select only those column who is having float data type
```

	Data_value	Period	rank
0	1116.386	1100.00	10.0
1	1070.874	2016.09	9.0
2	1054.408	2016.12	8.0
3	1010.665	2017.03	7.0
4	1233.700	2017.06	6.0
5	1282.436	2017.09	5.0
6	1290.820	2017.12	4.0
7	1412.007	2018.03	3.0
8	1488.055	2018.06	2.0
9	1497.678	2018.09	1.0

```
for i in df_10.columns:
  if i.startswith('D'):
    print(i)#return the value which is starting with D
```

```
Data_value
```

```
df_10[df_10['rank'].apply(lambda x:x>5)]#only on series # we take func as parameter
```

Saved successfully!

				Magnitude	rank
0	Dollars	1116.386	1100.00	6	10.0
1	Dollars	1070.874	2016.09	6	9.0
2	Dollars	1054.408	2016.12	6	8.0
3	Dollars	1010.665	2017.03	6	7.0
4	Dollars	1233.700	2017.06	6	6.0

```
df_10.groupby('UNITS').agg({'rank':'median','Data_value':'mean'})
```

	rank	Data_value
UNITS		
Dollars	5.5	1245.7029

```
df_10.sort_values(by='rank').head(5)
```

	UNITS	Data_value	Period	Magnitude	rank
9	Dollars	1497.678	2018.09	6	1.0
8	Dollars	1488.055	2018.06	6	2.0
7	Dollars	1412.007	2018.03	6	3.0
6	Dollars	1290.820	2017.12	6	4.0
5	Dollars	1282.436	2017.09	6	5.0

```
df_10['lag']=df_10['rank'].shift(1)#ek ghar pudhe jate
```

```
df_10.head(5)
```

	UNITS	Data_value	Period	Magnitude	rank	lag
0	Dollars	1116.386	1100.00	6	10.0	NaN
1	Dollars	1070.874	2016.09	6	9.0	10.0
2	Dollars	1054.408	2016.12	6	8.0	9.0
3	Dollars	1010.665	2017.03	6	7.0	8.0
4	Dollars	1233.700	2017.06	6	6.0	7.0

```
df_10['lag']=df_10['rank'].shift(-1)#ek ghar maghe sarkte
```

```
df_10.tail(5)
```

Saved successfully!

	UNITS	Data_value	Period	Magnitude	rank	lag
5	Dollars	1282.436	2017.09	6	5.0	4.0
6	Dollars	1290.820	2017.12	6	4.0	3.0
7	Dollars	1412.007	2018.03	6	3.0	2.0
8	Dollars	1488.055	2018.06	6	2.0	1.0
9	Dollars	1497.678	2018.09	6	1.0	NaN

```
person={
    'empid':[1,2,3,4,5],
    'email':['a@g.com','b@g.com','c@g.com','b@g.com','a@g.com']
}
```

```
df_person=pd.DataFrame(person)
```

```
df_person.drop_duplicates(subset=['email'],keep='first')
```

	empid	email
0	1	a@g.com
1	2	b@g.com
2	3	c@g.com

```
df_person
```

```
df_10['rank'].cumsum()
```

```
0    10.0
1    19.0
2    27.0
3    34.0
4    40.0
5    45.0
6    49.0
7    52.0
8    54.0
9    55.0
Name: rank, dtype: float64
```

```
df_10
```

	UNITS	Data_value	Period	Magnitude	rank	lag
0	Dollars	1116.386	1100.00	6	10.0	9.0
1	Dollars	1070.874	2016.09	6	9.0	8.0
				6	8.0	7.0
				6	7.0	6.0
4	Dollars	1233.700	2017.06	6	6.0	5.0
5	Dollars	1282.436	2017.09	6	5.0	4.0
6	Dollars	1290.820	2017.12	6	4.0	3.0
7	Dollars	1412.007	2018.03	6	3.0	2.0
8	Dollars	1488.055	2018.06	6	2.0	1.0
9	Dollars	1497.678	2018.09	6	1.0	NaN

Saved successfully!

```
df_10['UNITS'].str.split('a')
```

```
0    [Doll, rs]
1    [Doll, rs]
2    [Doll, rs]
3    [Doll, rs]
4    [Doll, rs]
5    [Doll, rs]
```

```
6    [Doll, rs]
7    [Doll, rs]
8    [Doll, rs]
9    [Doll, rs]
Name: UNITS, dtype: object
```

```
df_10=df_10.append({'UNITS':'Rupees'},ignore_index=True).tail(5)
```

```
df_10.sort_values(by='rank')
```

	UNITS	Data_value	Period	Magnitude	rank	lag
9	Dollars	1497.678	2018.09	6.0	1.0	NaN
8	Dollars	1488.055	2018.06	6.0	2.0	1.0
7	Dollars	1412.007	2018.03	6.0	3.0	2.0
6	Dollars	1290.820	2017.12	6.0	4.0	3.0
10	Rupees	NaN	NaN	NaN	NaN	NaN

```
df_10.nlargest(2,'rank')
```

	UNITS	Data_value	Period	Magnitude	rank	lag
6	Dollars	1290.820	2017.12	6.0	4.0	3.0
7	Dollars	1412.007	2018.03	6.0	3.0	2.0

```
df_10.nsmallest(2,'rank')
```

Saved successfully!

				Magnitude	rank	lag
				6.0	1.0	NaN
8	Dollars	1488.055	2018.06	6.0	2.0	1.0

```
df_10
```

	UNITS	Data_value	Period	Magnitude	rank	lag
6	Dollars	1290.820	2017.12	6.0	4.0	3.0
7	Dollars	1412.007	2018.03	6.0	3.0	2.0
8	Dollars	1488.055	2018.06	6.0	2.0	1.0
9	Dollars	1497.678	2018.09	6.0	1.0	NaN
10	Rupees	NaN	NaN	NaN	NaN	NaN

```
df_10.dropna(axis='index',how='any')
```

	UNITS	Data_value	Period	Magnitude	rank	lag
6	Dollars	1290.820	2017.12	6.0	4.0	3.0
7	Dollars	1412.007	2018.03	6.0	3.0	2.0
8	Dollars	1488.055	2018.06	6.0	2.0	1.0

```
df_10.loc[6:6,'UNITS']='RUPEES'
```

```
df_10
```

	UNITS	Data_value	Period	Magnitude	rank	lag
6	RUPEES	1290.820	2017.12	6.0	4.0	3.0
7	Dollars	1412.007	2018.03	6.0	3.0	2.0
8	Dollars	1488.055	2018.06	6.0	2.0	1.0
9	Dollars	1497.678	2018.09	6.0	1.0	NaN
10	Rupees	NaN	NaN	NaN	NaN	NaN

```
df.isna()
```

	Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Sul
0	False	False	False	True	False	False	False	
1	False	False	False	True	False	False	False	
2	False	False	False	True	False	False	False	
			False	True	False	False	False	
4	False	False	False	True	False	False	False	
...	...	...	...	...	...	...	...	
5950	False	False	False	True	False	False	False	
5951	False	False	False	True	False	False	False	
5952	False	False	False	True	False	False	False	
5953	False	False	False	True	False	False	False	
5954	False	False	False	True	False	False	False	

5955 rows × 14 columns

Saved successfully!

```
df_10.dtypes
```

```
UNITS          object
Data_value    float64
Period        float64
```

```
Magnitude     float64
rank          float64
lag           float64
dtype: object
```

```
df_10['rank'].replace('mean()',000)
```

```
6     4.0
7     3.0
8     2.0
9     1.0
10    NaN
Name: rank, dtype: float64
```

```
df_10['rank'].fillna(0)
```

```
6     4.0
7     3.0
8     2.0
9     1.0
10    0.0
Name: rank, dtype: float64
```

```
df_10
#df.to_datetime(date_column,format=)
#df[date_column].day_name()
#
```

	UNITS	Data_value	Period	Magnitude	rank	lag
6	RUPEES	1290.820	2017.12	6.0	4.0	3.0
7	Dollars	1412.007	2018.03	6.0	3.0	2.0
8			06	6.0	2.0	1.0
9	Dollars	1497.678	2018.09	6.0	1.0	NaN
10	Rupees	NaN	NaN	NaN	NaN	NaN

Saved successfully!

```
data = {
    "A": ["TeamA", "TeamB", "TeamB", "TeamC", "TeamA"],
    "B": [50, 40, 40, 30, 50],
    "C": [True, False, False, False, True]
}
```

```
df_data=pd.DataFrame(data)
```

```
set_firstrow_header=df_data.T#Transpose
set_firstrow_header
```

	0	1	2	3	4
A	TeamA	TeamB	TeamB	TeamC	TeamA
B	50	40	40	30	50

```
set_firstrow_header.columns=set_firstrow_header.iloc[0]
set_firstrow_header
```

	A	TeamA	TeamB	TeamB	TeamC	TeamA
A	TeamA	TeamB	TeamB	TeamC	TeamA	
B	50	40	40	30	50	
C	True	False	False	False	True	

```
set_firstrow_header=set_firstrow_header[1:]
set_firstrow_header
```

	A	TeamA	TeamB	TeamB	TeamC	TeamA
B	50	40	40	30	50	
C	True	False	False	False	True	

```
set_firstrow_header['TeamA']
```

	A	TeamA	TeamA
B	50	50	
C	True	True	

Saved successfully!

```
df_data.drop_duplicates()
```

	A	B	C
0	TeamA	50	True
1	TeamB	40	False
3	TeamC	30	False

```
df_data[df_data.duplicated()]
```

	A	B	C
2	TeamB	40	False
4	TeamA	50	True

```
df_data.where(df_data['A']=='TeamB')
```

	A	B	C
0	NaN	NaN	NaN
1	TeamB	40.0	False
2	TeamB	40.0	False
3	NaN	NaN	NaN
4	NaN	NaN	NaN

```
set_firstrow_header.reset_index().set_index('index')
```

index	A	TeamA	TeamB	TeamB	TeamC	TeamA
B		50	40	40	30	50
C		True	False	False	False	True

```
df_data
```

	A	B	C
0	TeamA	50	True
1	TeamB	40	False
2	TeamB	40	False

Saved successfully!

```
df_data[~df_data.duplicated()].set_index('A')
```

A	B	C
TeamA	50	True
TeamB	40	False
TeamC	30	False

```
#innner=pd.merge(left_column,right_column,left_on,right_on,how='inner','left','right','out
#IN CROSS JOINWE DON't use left_on and right_on
#-- index-index,index-column_value,column_value-index,column_value-column_value
```

```
#CONCAT([first_dataframe,second_dataframe],sort=False,keys=['num0','num1'],axis=0 or 1)---

#JOIN(left,right,left_on,right_on,how)--index of left and index of left --column values of
```