Problems in dataframe:

- missing values
- wrong information
- wrong format
- normalization

In [1]:

```
#import the package
import pandas as pd
```

In [2]:

```
#reading the data
df_sample = pd.read_clipboard()
```

In [3]:

```
#viewing the data
df_sample
```

Out[3]:

	id	Name	mark	gender	dept
0	1	a	10	m	cyber
1	2	b	20	male	cys
2	3	c	30	m	iot
3	4	d	40	female	ai ml
4	5	e	50	girl	ai da
5	6	f	60	girl	da
6	7	g	70	female	ml
7	8	h	80	f	ai da
8	9	i	90	male	cyber

In [4]:

```
#since the gender column is in wrong fomat
# mapping m,male,Male,M,boy to M
# mapping f,female,F,Female,girl to F
df_sample['gender'].unique()
```

Out[4]:

```
array(['m', 'male', 'female', 'girl', 'f'], dtype=object)
```

In [5]:

```
#creating a dictionary to map the respective values in the same format
dic = {'m':"M",'male':"M","female":"F","girl":"F","f":"F"}
```

In [6]:

```
df_sample['gender'] = df_sample['gender'].map(dic)
```

In [7]:

```
df_sample
```

Out[7]:

	id	Name	mark	gender	dept
0	1	a	10	M	cyber
1	2	b	20	M	cys
2	3	c	30	M	iot
3	4	d	40	F	ai ml
4	5	e	50	F	ai da
5	6	f	60	F	da
6	7	g	70	F	ml
7	8	h	80	F	ai da
8	9	i	90	M	cyber

In [8]:

```
unique = set(df_sample["dept"])
```

In [9]:

```
unique
```

Out[9]:

```
{' ai ml', 'ai da', 'cyber', 'cys', 'da', 'iot', 'ml'}
```

In [10]:

```
dic = {" ai ml":"AI & ML","ai da":"AI & DA","cyber":"Cybersecurity & IOT","cys":"Cyb
```

In [11]:

```
df_sample['dept'] = df_sample['dept'].map(dic)
```

In [12]:

```
df_sample
```

Out[12]:

	id	Name	mark	gender	dept
0	1	a	10	M	Cybersecurity & IOT
1	2	b	20	M	Cybersecurity & IOT
2	3	c	30	M	Cybersecurity & IOT
3	4	d	40	F	AI & ML
4	5	e	50	F	AI & DA
5	6	f	60	F	AI & DA
6	7	g	70	F	AI & ML
7	8	h	80	F	AI & DA
8	9	i	90	M	Cybersecurity & IOT

In [13]:

```
df_sample
```

Out[13]:

	id	Name	mark	gender	dept
0	1	a	10	M	Cybersecurity & IOT
1	2	b	20	M	Cybersecurity & IOT
2	3	c	30	M	Cybersecurity & IOT
3	4	d	40	F	AI & ML
4	5	e	50	F	AI & DA
5	6	f	60	F	AI & DA
6	7	g	70	F	AI & ML
7	8	h	80	F	AI & DA
8	9	i	90	M	Cybersecurity & IOT

In [18]:

```
# handling the missing data
new_df = pd.read_clipboard()
```

In [19]:

```
1 new_df
```

Out[19]:

	id	Mark
0	1	10.0
1	2	20.0
2	3	NaN
3	4	40.0
4	5	50.0
5	6	NaN
6	7	70.0
7	8	80.0
8	9	NaN

In [20]:

```
1 new_df['Mark']
```

Out[20]:

```
0    10.0
1    20.0
2     NaN
3    40.0
4    50.0
5     NaN
6    70.0
7    80.0
8     NaN
Name: Mark, dtype: float64
```

In [21]:

```
1 new_df.isnull()
```

Out[21]:

	id	Mark
0	False	False
1	False	False
2	False	True
3	False	False
4	False	False
5	False	True
6	False	False
7	False	False
8	False	True

In [22]:

```
1 new_df.isnull().sum()
```

Out[22]:

```
id      0
Mark    3
dtype: int64
```

In [23]:

```
1 new_df.describe()
```

Out[23]:

	id	Mark
count	9.000000	6.000000
mean	5.000000	45.000000
std	2.738613	27.386128
min	1.000000	10.000000
25%	3.000000	25.000000
50%	5.000000	45.000000
75%	7.000000	65.000000
max	9.000000	80.000000

In [27]:

```
1 new_df['Mark']
```

Out[27]:

```
0    10.0
1    20.0
2     NaN
3    40.0
4    50.0
5     NaN
6    70.0
7    80.0
8     NaN
Name: Mark, dtype: float64
```

In [26]:

```
1 new_df['Mark'].mean()
```

Out[26]:

```
45.0
```

In [29]:

```
1 new_df['Mark'].fillna(new_df['Mark'].mean())
```

Out[29]:

```
0    10.0
1    20.0
2    45.0
3    40.0
4    50.0
5    45.0
6    70.0
7    80.0
8    45.0
Name: Mark, dtype: float64
```

In [30]:

```
1 new_df['Mark'].fillna(new_df['Mark'].min())
```

Out[30]:

```
0    10.0
1    20.0
2    10.0
3    40.0
4    50.0
5    10.0
6    70.0
7    80.0
8    10.0
Name: Mark, dtype: float64
```

In [31]:

```
1 new_df['Mark'].fillna(new_df['Mark'].max())
```

Out[31]:

```
0    10.0
1    20.0
2    80.0
3    40.0
4    50.0
5    80.0
6    70.0
7    80.0
8    80.0
Name: Mark, dtype: float64
```

In [32]:

```
1 new_df['Mark'].fillna(new_df['Mark'].std())
```

Out[32]:

```
0    10.000000
1    20.000000
2    27.386128
3    40.000000
4    50.000000
5    27.386128
6    70.000000
7    80.000000
8    27.386128
Name: Mark, dtype: float64
```

In [33]:

```
1 new_df['Mark'].fillna(new_df['id'].max())
```

Out[33]:

```
0    10.0
1    20.0
2     9.0
3    40.0
4    50.0
5     9.0
6    70.0
7    80.0
8     9.0
Name: Mark, dtype: float64
```