```
# Handling numerical Data

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler, StandardScaler,Normalizer
```

```
df = pd.read_csv("housing.csv")
df.head(5)
```

	RM	LSTAT	PTRATIO	MEDV
0	6.575	4.98	15.3	504000.0
1	NaN	9.14	17.8	453600.0
2	7.185	4.00	17.8	728700.0
3	6.998	2.94	18.7	701400.0
4	7.147	5.33	18.7	760200.0

Next steps: Generate code with df New interactive sheet

```
df.describe()
```

	RM	LSTAT	PTRATIO	MEDV
count	486.000000	485.000000	486.000000	4.880000e+02
mean	6.220737	12.933340	18.525720	4.543961e+05
std	0.733110	7.094904	2.101756	1.655058e+05
min	0.727000	1.980000	12.600000	1.050000e+05
25%	5.881000	7.370000	17.400000	3.501750e+05
50%	6.185000	11.660000	19.100000	4.389000e+05
75%	6.578000	17.120000	20.200000	5.187000e+05
max	8.398000	37.970000	22.000000	1.024800e+06

```
df.duplicated().sum()
```

```
np.int64(0)
```

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

	0
RM	3
LSTAT	4
PTRATIO	3
MEDV	1

dtype: int64

```
df["RM"] = df["RM"].fillna(df["RM"].mode())
df["LSTAT"] = df["LSTAT"].fillna(df["LSTAT"].mean())
df["PTRATIO"] = df["PTRATIO"].fillna(df["PTRATIO"].median())
```

```
df["MEDV"] = df["MEDV"].fillna(df["MEDV"].median())
```

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

	0
RM	0
LSTAT	0
PTRATIO	0
MEDV	0

dtype: int64

```
# min max scaling
scaler = MinMaxScaler()

df_minmax = df.copy()
df_minmax[["RM", "LSTAT", "PTRATIO", "MEDV"]] = scaler.fit_transform(
    df_minmax[["RM", "LSTAT", "PTRATIO", "MEDV"]]
)

print("After MinMax Scaling:")
display(df_minmax.head())
```

```
After MinMax Scaling:
```

	RM	LSTAT	PTRATIO	MEDV
0	0.762352	0.083356	0.287234	0.433790
1	0.716170	0.198944	0.553191	0.378995
2	0.841872	0.056127	0.553191	0.678082
3	0.817494	0.026674	0.648936	0.648402
4	0.836918	0.093081	0.648936	0.712329

```
# Standardization or Z score formula.. (x - mean) / std

std = StandardScaler()

df_std = df.copy()
df_std[["RM", "LSTAT", "PTRATIO", "MEDV"]] = std.fit_transform(
    df_std[["RM", "LSTAT", "PTRATIO", "MEDV"]]
)

print("After Standardization:")
display(df_std.head())
```

```
After Standardization:
```

	RM	LSTAT	PTRATIO	MEDV
0	0.485222	-1.126769	-1.542419	0.300515
1	0.000000	-0.537412	-0.348317	-0.004628
2	1.320718	-1.265608	-0.348317	1.660944
3	1.064590	-1.415780	0.081560	1.495659
4	1.268670	-1.077183	0.081560	1.851659

```
# Normalization or L2
norm = Normalizer()
```

```
df_norm = df.copy()
# Impute missing values before normalization
df_norm["RM"] = df_norm["RM"].fillna(df_norm["RM"].mean())
df_norm["LSTAT"] = df_norm["LSTAT"].fillna(df_norm["LSTAT"].mean())
df_norm["PTRATIO"] = df_norm["PTRATIO"].fillna(df_norm["PTRATIO"].mean())
df_norm["MEDV"] = df_norm["MEDV"].fillna(df_norm["MEDV"].mean())

df_norm[["RM", "LSTAT", "PTRATIO", "MEDV"]] = norm.fit_transform(
    df_norm[["RM", "LSTAT", "PTRATIO", "MEDV"]]
)

print("After Normalization:")
display(df_norm.head())
```

```
After Normalization:
```

	RM	LSTAT	PTRATIO	MEDV
0	0.000013	0.000010	0.000030	1.0
1	0.000014	0.000020	0.000039	1.0
2	0.000010	0.000005	0.000024	1.0
3	0.000010	0.000004	0.000027	1.0
4	0.000009	0.000007	0.000025	1.0

```
df.describe()
```

	RM	LSTAT	PTRATIO	MEDV
count	489.000000	489.000000	489.000000	4.890000e+02
mean	6.220737	12.933340	18.529243	4.543644e+05
std	0.730853	7.065766	2.095767	1.653376e+05
min	0.727000	1.980000	12.600000	1.050000e+05
25%	5.884000	7.390000	17.400000	3.507000e+05
50%	6.193000	11.720000	19.100000	4.389000e+05
75%	6.575000	17.110000	20.200000	5.187000e+05
max	8.398000	37.970000	22.000000	1.024800e+06