

```
import pandas as pd

df = pd.read_csv("/content/healthcare_LOS_dataset.csv")

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Age                   1000 non-null   int64
1   Sex                   1000 non-null   int64
2   BMI                   1000 non-null   float64
3   Severity_Score        1000 non-null   int64
4   Comorbidities_Count   1000 non-null   int64
5   Previous_Admissions   1000 non-null   int64
6   Admission_Type        1000 non-null   int64
7   Blood_Pressure        1000 non-null   float64
8   Heart_Rate            1000 non-null   float64
9   WBC_Count             1000 non-null   int64
10  LOS                   1000 non-null   float64
dtypes: float64(4), int64(7)
memory usage: 86.1 KB
```


df

	Age	Sex	BMI	Severity_Score	Comorbidities_Count	Previous_Admissions	Admission_Type	Blood_Pressure	Heart_Rate	WBC_Count	LOS
0	69	0	34.7	5	2	1	1	131.801737	79.305769	5821	12.2
1	32	0	36.0	1	4	0	1	131.923441	92.132398	8273	8.5
2	89	1	23.6	8	4	0	1	130.485348	79.080733	9753	14.6
3	78	0	28.4	1	2	3	1	162.641398	66.828044	6241	6.6
4	38	0	20.4	4	2	2	1	146.651894	45.742046	7254	10.6
...	...	...	...	...	...	...	...	...	...	...	...
995	27	1	27.6	7	3	0	0	123.856145	74.128021	5424	8.7
996	51	1	29.3	8	2	1	0	137.083958	97.970103	6534	6.8
997	72	1	31.5	2	2	5	0	126.381124	95.303418	7730	10.4
998	49	0	24.9	7	2	0	1	125.022324	69.819966	8558	8.2
999	67	0	24.0	5	4	1	1	144.098348	70.882284	6432	12.2

1000 rows × 11 columns

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

```
df.isna().sum()
```

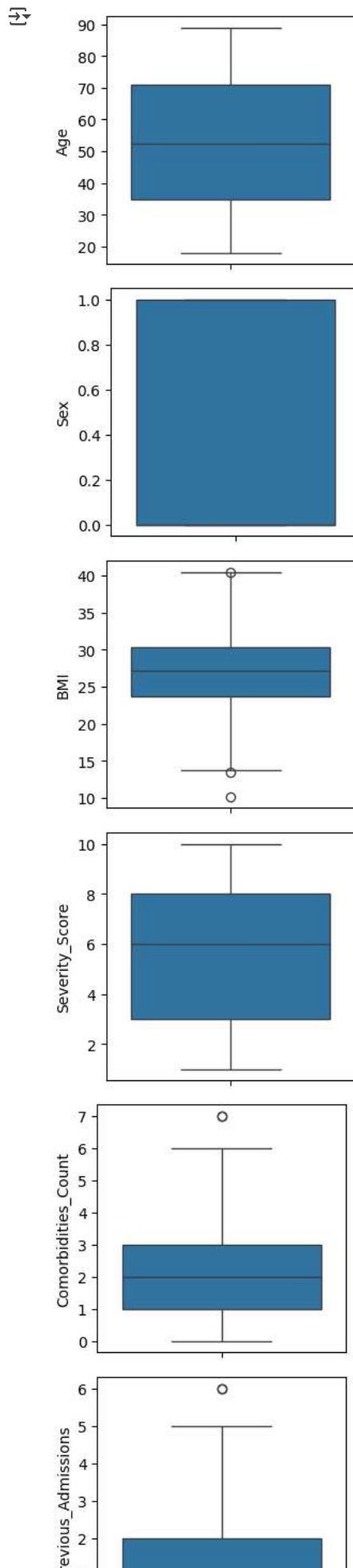


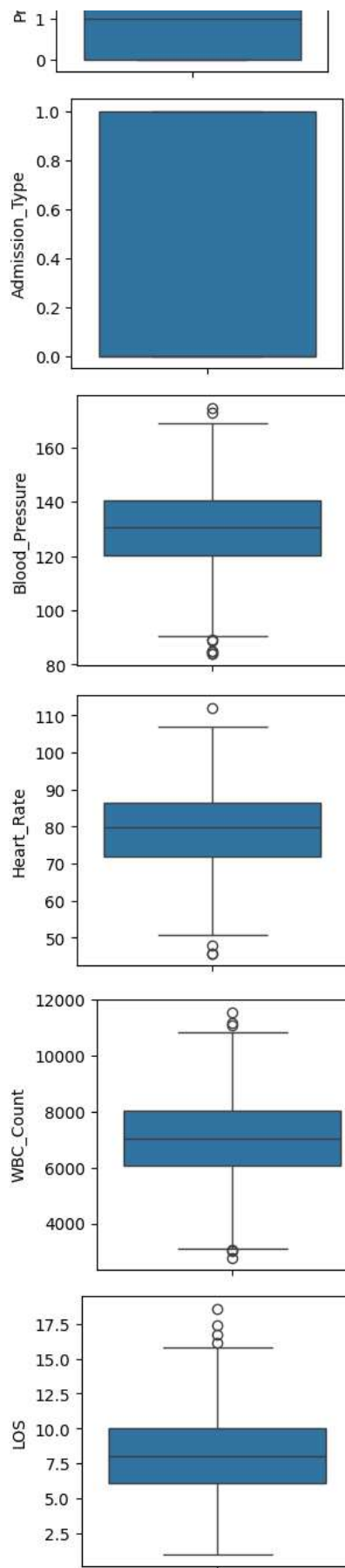
	0
Age	0
Sex	0
BMI	0
Severity_Score	0
Comorbidities_Count	0
Previous_Admissions	0
Admission_Type	0
Blood_Pressure	0
Heart_Rate	0
WBC_Count	0
LOS	0

**dtype:** int64

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
for column in df.columns:
    plt.figure(figsize=(3, 3))
    sns.boxplot(y=df[column])
    plt.show()
```





```
#REMOVE OUTLIER
```

```
import pandas as pd
```

```
def filter_outliers(df):
    df_cleaned = df.copy()
    for column in df.columns:
        if df[column].dtype in ['int64', 'float64']: # Apply only to numerical columns
            q1 = df[column].quantile(0.25)
            q3 = df[column].quantile(0.75)
            IQR = q3 - q1 # Corrected calculation
            lower_bound = q1 - 1.5 * IQR
            upper_bound = q3 + 1.5 * IQR
            df_cleaned = df_cleaned[(df_cleaned[column] >= lower_bound) & (df_cleaned[column] <= upper_bound)]

    return df_cleaned
```

```
# Example Usage:
```

```
df_cleaned = filter_outliers(df)
print(df_cleaned)
```

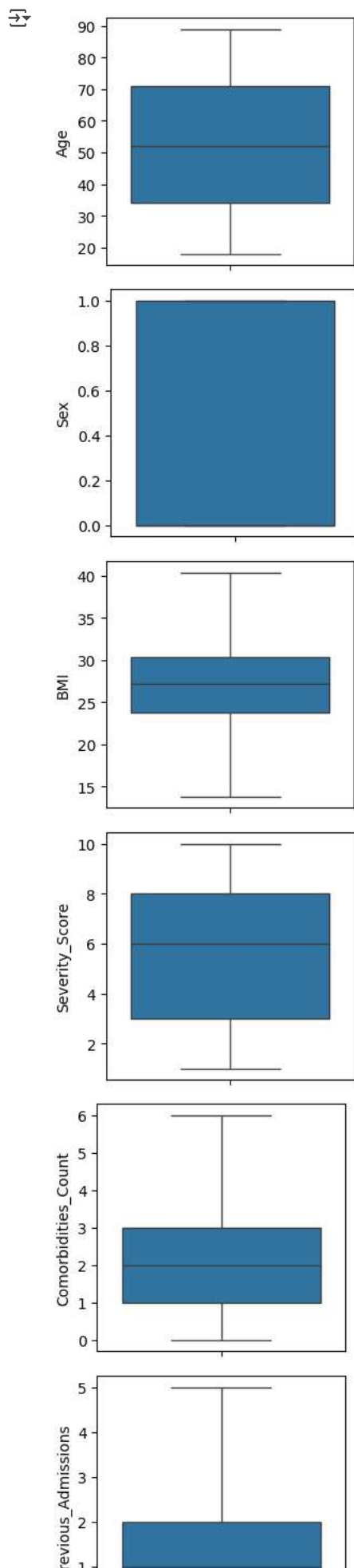
```
↩
```

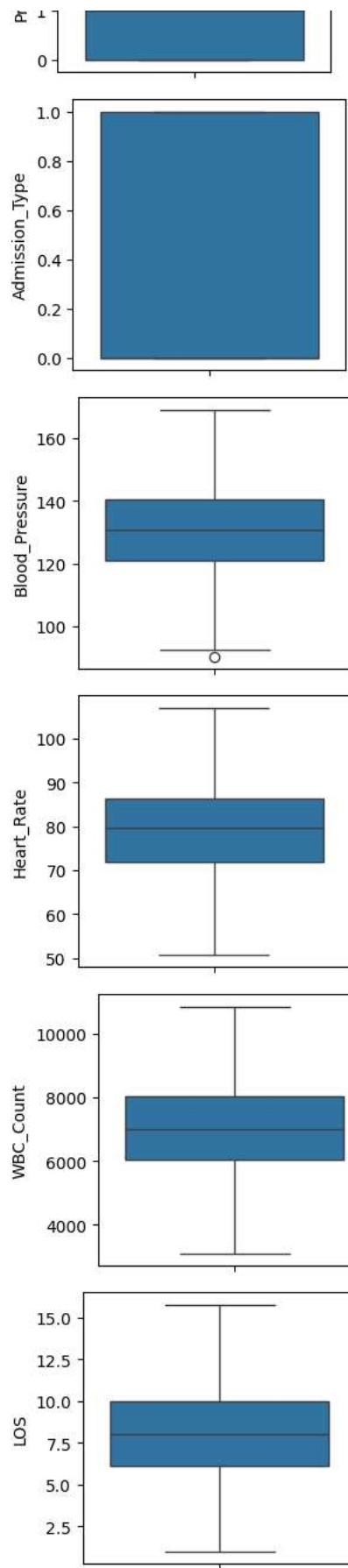
	Age	Sex	BMI	Severity_Score	Comorbidities_Count	Previous_Admissions	\
0	69	0	34.7	5	2		1
1	32	0	36.0	1	4		0
2	89	1	23.6	8	4		0
3	78	0	28.4	1	2		3
5	41	0	32.4	9	4		1
..	...	...	...	...	...		...
995	27	1	27.6	7	3		0
996	51	1	29.3	8	2		1
997	72	1	31.5	2	2		5
998	49	0	24.9	7	2		0
999	67	0	24.0	5	4		1

	Admission_Type	Blood_Pressure	Heart_Rate	WBC_Count	LOS
0	1	131.801737	79.305769	5821	12.2
1	1	131.923441	92.132398	8273	8.5
2	1	130.485348	79.080733	9753	14.6
3	1	162.641398	66.828044	6241	6.6
5	0	134.430876	61.618516	5871	10.7
..	...	...	...	...	...
995	0	123.856145	74.128021	5424	8.1
996	0	137.083958	97.970103	6534	6.8
997	0	126.381124	95.303418	7730	10.4
998	1	125.022324	69.819966	8558	8.2
999	1	144.098348	70.882284	6432	12.2


```
[972 rows x 11 columns]
```

```
for column in df_cleaned.columns:
    plt.figure(figsize=(3, 3))
    sns.boxplot(y=df_cleaned[column])
    plt.show()
```





df\_cleaned




	Age	Sex	BMI	Severity_Score	Comorbidities_Count	Previous_Admissions	Admission_Type	Blood_Pressure	Heart_Rate	WBC_Count	LOS
0	69	0	34.7	5	2	1	1	131.801737	79.305769	5821	12.2
1	32	0	36.0	1	4	0	1	131.923441	92.132398	8273	8.5
2	89	1	23.6	8	4	0	1	130.485348	79.080733	9753	14.6
3	78	0	28.4	1	2	3	1	162.641398	66.828044	6241	6.6
5	41	0	32.4	9	4	1	0	134.430876	61.618516	5871	10.7
...	...	...	...	...	...	...	...	...	...	...	...
995	27	1	27.6	7	3	0	0	123.856145	74.128021	5424	8.7
996	51	1	29.3	8	2	1	0	137.083958	97.970103	6534	6.8
997	72	1	31.5	2	2	5	0	126.381124	95.303418	7730	10.4
998	49	0	24.9	7	2	0	1	125.022324	69.819966	8558	8.2
999	67	0	24.0	5	4	1	1	144.098348	70.882284	6432	12.2

972 rows × 11 columns

Next steps: [Generate code with df\\_cleaned](#) [View recommended plots](#) [New interactive sheet](#)

df\_cleaned.describe().T




	count	mean	std	min	25%	50%	75%	max
Age	972.0	52.765432	21.004647	18.000000	34.000000	52.000000	71.000000	89.000000
Sex	972.0	0.496914	0.500248	0.000000	0.000000	0.000000	1.000000	1.000000
BMI	972.0	27.068210	4.846654	13.800000	23.700000	27.200000	30.400000	40.400000
Severity_Score	972.0	5.484568	2.885164	1.000000	3.000000	6.000000	8.000000	10.000000
Comorbidities_Count	972.0	1.929012	1.401816	0.000000	1.000000	2.000000	3.000000	6.000000
Previous_Admissions	972.0	1.034979	1.005039	0.000000	0.000000	1.000000	2.000000	5.000000
Admission_Type	972.0	0.493827	0.500219	0.000000	0.000000	0.000000	1.000000	1.000000
Blood_Pressure	972.0	130.650570	14.504539	90.407271	120.831193	130.846763	140.433651	169.070684
Heart_Rate	972.0	79.464631	9.906675	50.792090	71.900940	79.742867	86.441707	107.066183
WBC_Count	972.0	7013.416667	1483.096693	3108.000000	6059.000000	7008.500000	8038.250000	10855.000000
LOS	972.0	7.992490	2.943700	1.000000	6.100000	8.000000	10.000000	15.800000

#Normalized

from sklearn.preprocessing import MinMaxScaler

```
scaler=MinMaxScaler()
df_cleaned_norm = scaler.fit_transform(df_cleaned)
df_cleaned_norm = pd.DataFrame(df_cleaned_norm, columns=df_cleaned.columns)
df_cleaned_norm.head()
```



	Age	Sex	BMI	Severity_Score	Comorbidities_Count	Previous_Admissions	Admission_Type	Blood_Pressure	Heart_Rate	WBC_Count
0	0.718310	0.0	0.785714	0.444444	0.333333	0.2	1.0	0.526223	0.506693	0.350201
1	0.197183	0.0	0.834586	0.000000	0.666667	0.0	1.0	0.527770	0.734624	0.6667
2	1.000000	1.0	0.368421	0.777778	0.666667	0.0	1.0	0.509488	0.502694	0.85771
3	0.845070	0.0	0.548872	0.000000	0.333333	0.6	1.0	0.918268	0.284962	0.4044
4	0.323944	0.0	0.699248	0.888889	0.666667	0.2	0.0	0.559645	0.192387	0.35661

Next steps: [Generate code with df\\_cleaned\\_norm](#) [View recommended plots](#) [New interactive sheet](#)