

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

# Step 1: Load the dataset
df = pd.read_csv('kidney_disease.csv')

# shows the first 5 lines of the table.
df.head()
```

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	...	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	...	38	6000	NaN	no	no	no	good	no	no	ckd
2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	...	31	7500	NaN	no	yes	no	poor	no	yes	ckd
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	...	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	...	35	7300	4.6	no	no	no	good	no	no	ckd

5 rows × 25 columns

```
# Step 2: Handle Missing Values
# Filling numeric columns with Mean
numeric_cols = df.select_dtypes(include=['float64', 'int64']).columns
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())

# Filling categorical columns with Mode
object_cols = df.select_dtypes(include=['object']).columns
for col in object_cols:
    df[col] = df[col].fillna(df[col].mode()[0])

# all the spaces are filled in so it should print 0
print("Total number of empty cells now:")
print(df.isnull().sum().sum())
```

Total number of empty cells now:

0

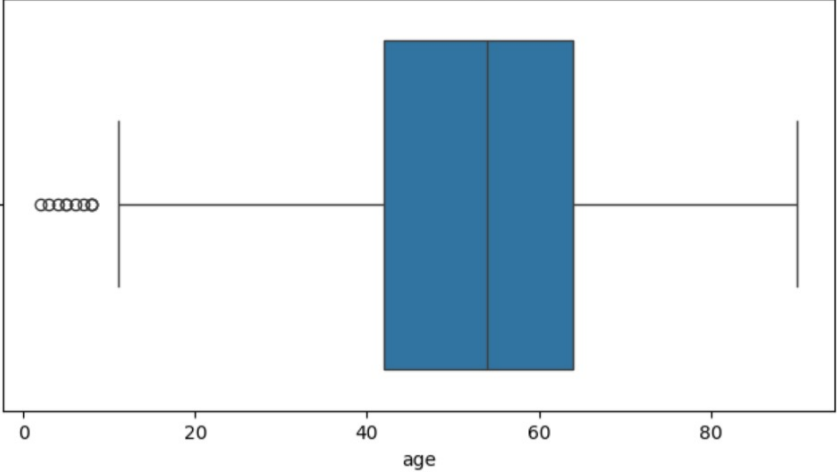
```
# Step 3: Handle Outliers using IQR method
# Visualizing outliers in 'age' as an example
plt.figure(figsize=(8, 4))
sns.boxplot(x=df['age'])
plt.title('Outliers in Age')
plt.show()

# Removing outliers
numeric_cols = df.select_dtypes(include=[np.number]).columns
Q1 = df[numeric_cols].quantile(0.25)
Q3 = df[numeric_cols].quantile(0.75)
IQR = Q3 - Q1

df = df[~((df[numeric_cols] < (Q1 - 1.5 * IQR)) | (df[numeric_cols] > (Q3 + 1.5 * IQR))).any(axis=1)]

print("Outliers cleaned successfully!")
print("Number of remaining rows in the dataset:", df.shape[0])
```

Outliers in Age



Outliers cleaned successfully!
Number of remaining rows in the dataset: 253

```
# Step 4: Data Visualization
# 1. Correlation Heatmap
plt.figure(figsize=(12, 8))
numeric_df = df.select_dtypes(include=[np.number])
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()

# 2. Disease Classification Distribution
plt.figure(figsize=(6, 4))
sns.countplot(x='classification', data=df, palette='Set2')
plt.title('Distribution of Kidney Disease Cases')
plt.show()

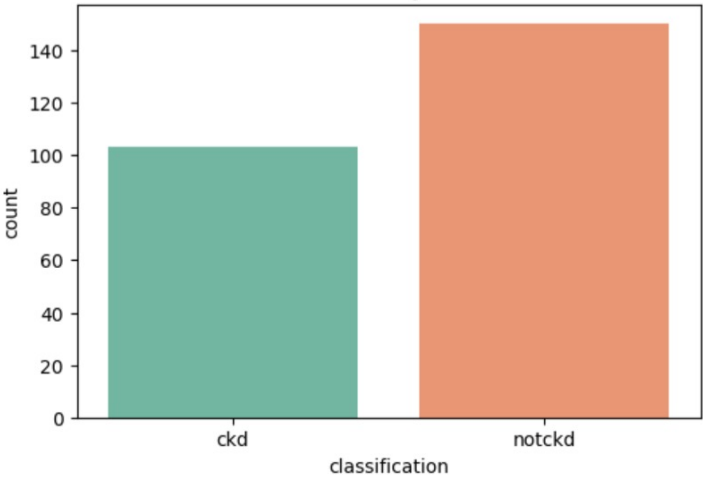
# 3. Blood Pressure Distribution
plt.figure(figsize=(8, 5))
sns.histplot(df['bp'], kde=True, color='purple')
plt.title('Distribution of Blood Pressure')
plt.show()
```

/tmp/ipykernel_920/3688114482.py:11: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False`

```
sns.countplot(x='classification', data=df, palette='Set2')
```

Distribution of Kidney Disease Cases



Distribution of Blood Pressure

