Exploratory Data Analysis

```
import pandas as pd
df=pd.read_csv('penguins_size.csv')
df.info()
df=df.dropna()
df.isnull().values.any()
df['species'].replace({'Adelie': 1, 'Chinstrap': 2, 'Gentoo': 3}, inplace=True)
df['island'].replace({'Torgersen': 1, 'Biscoe': 2, 'Dream': 3}, inplace=True)
df['sex'].replace({'MALE': 1, 'FEMALE': 2}, inplace=True)
df.info()
print(df)
df['sex'] = pd.to_numeric(df['sex'], errors='coerce')
import matplotlib.pyplot as plt
import seaborn as sb
sb.heatmap(df.corr() > 0.7, annot=True, cbar=False)
plt.show()
d1 = df.drop('flipper_length_mm', axis=1)
d2 = d1.drop('species', axis=1)
x=d2.drop('body_mass_g', axis=1)
y=d2.body_mass_g
sb.heatmap(d2.corr() > 0.7, annot=True, cbar=False)
plt.show()
from sklearn.model_selection import train_test_split
x_train,x_test,y_train, y_test= train_test_split(x,y, test_size=0.2, random_state=42)
from sklearn.linear_model import LinearRegression
reg = LinearRegression().fit(x,y)
reg.score(x,y)
#Compare to before preprocessing
x1_train,x1_test,y1_train, y1_test= train_test_split(x1,y1, test_size=0.2,
random_state=42)
x1=df.drop('body_mass_g', axis=1)
y1=df.body_mass_g
reg = LinearRegression().fit(x1,y1)
reg.score(x1,y1)
```

```
wn = pd.read_csv('winequality2.csv')
print(wn.head())
wn.info()
wn.isnull().values.any()
#Before Preprocessing score
from sklearn.model_selection import train_test_split
x=wn.drop('quality', axis=1)
y=wn.quality
x_train,x_test,y_train, y_test= train_test_split(x,y, test_size=0.2, random_state=42)
from sklearn.linear_model import LinearRegression
reg = LinearRegression().fit(x,y)
reg.score(x,y)
sb.heatmap(wn.corr() > 0.5, annot=True, cbar=False)
plt.show()
w1 = wn.drop('free sulfur dioxide', axis=1)
w2 = w1.drop('density', axis=1)
w3 = w2.drop('volatile acidity', axis=1)
w4 = w3.drop('citric acid', axis=1)
x1=w4.drop('quality', axis=1)
y1=w4.quality
x1_train,x1_test,y1_train, y1_test= train_test_split(x1,y1, test_size=0.2,
random_state=42)
sb.heatmap(w4.corr() > 0.5, annot=True, cbar=False)
plt.show()
reg = LinearRegression().fit(x1,y1)
reg.score(x1,y1)

print("Highest value",w4['fixed acidity'].mean() + 3*w4['fixed acidity'].std())
print("Lowest value",w4['fixed acidity'].mean() - 3*w4['fixed acidity'].std())
w4[(w4['fixed acidity'] > 13.54) | (w4['fixed acidity'] < 3.10)]
new_w4 = w4[(w4['fixed acidity'] < 13.54) & (w4['fixed acidity'] > 3.10)]
new_w4
sb.boxplot(w4['fixed acidity'])
plt.show()
sb.boxplot(new_w4['fixed acidity'])
plt.show()
upper_limit = w4['fixed acidity'].mean() + 3*w4['fixed acidity'].std()
lower_limit = w4['fixed acidity'].mean() – 3*w4['fixed acidity'].std()
w4['fixed acidity'] = np.where(
w4['fixed acidity']>upper_limit,
```

```
upper_limit,
np.where(
w4['fixed acidity']<lower_limit,
lower_limit,
w4['fixed acidity']))
w4['fixed acidity'].describe()
sb.boxplot(w4['fixed acidity'])
plt.show()
```

```
print("Highest value",w4['residual sugar'].mean() + 3*w4['residual sugar'].std())
print("Lowest value",w4['residual sugar'].mean() - 3*w4['residual sugar'].std())
w4[(w4['residual sugar'] > 6.77) | (w4['residual sugar'] < -1.69)]
new1_w4 = w4[(w4['residual sugar'] < 6.77) & (w4['residual sugar'] > -1.69)]
new1_w4
sb.boxplot(w4['residual sugar'])
plt.show()
sb.boxplot(new1_w4['residual sugar'])
plt.show()
upper_limit = w4['residual sugar'].mean() + 3*w4['residual sugar'].std()
lower_limit = w4['residual sugar'].mean() – 3*w4['residual sugar'].std()
w4['residual sugar'] = np.where(
w4['residual sugar']>upper_limit,
upper_limit,
np.where(
w4['residual sugar']<lower_limit,
lower_limit,
w4['residual sugar']))
w4['residual sugar'].describe()
sb.boxplot(w4['residual sugar'])
plt.show()
```

```
print("Highest value",w4['chlorides'].mean() + 3*w4['chlorides'].std())
print("Lowest allowed",w4['chlorides'].mean() - 3*w4['chlorides'].std())
w4[(w4['chlorides'] > 0.23) | (w4['chlorides'] <-0.05)]
new2_w4 = w4[(w4['chlorides'] < 0.23) & (w4['chlorides'] > -0.05)]
```

```
new2_w4
sb.boxplot(w4['chlorides'])
plt.show()
sb.boxplot(new1_w4['chlorides'])
plt.show()
upper_limit = w4['chlorides'].mean() + 3*w4['chlorides'].std()
lower_limit = w4['chlorides'].mean() – 3*w4['chlorides'].std()
w4['chlorides'] = np.where(
w4['chlorides']>upper_limit,
upper_limit,
np.where(
w4['chlorides']<lower_limit,
lower_limit,
w4['chlorides']))
w4['chlorides'].describe()
sb.boxplot(w4['chlorides'])
plt.show()

print("Highest value",w4['total sulfur dioxide'].mean() + 3*w4['total sulfur dioxide'].std())
print("Lowest value",w4['total sulfur dioxide'].mean() - 3*w4['total sulfur dioxide'].std())
w4[(w4['total sulfur dioxide'] > 145.15) | (w4['total sulfur dioxide'] < -52.22)]
new3_w4 = w4[(w4['total sulfur dioxide'] < 145.15) & (w4['total sulfur dioxide'] > -52.22)]
new3_w4
sb.boxplot(w4['total sulfur dioxide'])
plt.show()
sb.boxplot(new1_w4['total sulfur dioxide'])
plt.show()
upper_limit = w4['total sulfur dioxide'].mean() + 3*w4['total sulfur dioxide'].std()
lower_limit = w4['total sulfur dioxide'].mean() – 3*w4['total sulfur dioxide'].std()
w4['total sulfur dioxide'] = np.where(
w4['total sulfur dioxide']>upper_limit,
upper_limit,
np.where(
w4['total sulfur dioxide']<lower_limit,
lower_limit,
w4['total sulfur dioxide']))
w4['total sulfur dioxide'].describe()
sb.boxplot(w4['total sulfur dioxide'])
plt.show()
```

```
print("Highest value",w4['pH'].mean() + 3*w4['pH'].std())
print("Lowest value",w4['pH'].mean() - 3*w4['pH'].std())
w4[(w4['pH'] > 3.77) | (w4['pH'] <2.85)]
new4_w4 = w4[(w4['pH'] < 3.77) & (w4['pH'] > 2.85)]
new4_w4
sb.boxplot(w4['pH'])
plt.show()
sb.boxplot(new1_w4['pH'])
plt.show()
upper_limit = w4['pH'].mean() + 3*w4['pH'].std()
lower_limit = w4['pH'].mean() – 3*w4['pH'].std()
w4['pH'] = np.where(
w4['pH']>upper_limit,
upper_limit,
np.where(
w4['pH']<lower_limit,
lower_limit,
w4['pH']))
w4['pH'].describe()
sb.boxplot(w4['pH'])
plt.show()

print("Highest value",w4['sulphates'].mean() + 3*w4['sulphates'].std())
print("Lowest value",w4['sulphates'].mean() - 3*w4['sulphates'].std())
w4[(w4['sulphates'] > 1.17) | (w4['sulphates'] <0.15)]
new5_w4 = w4[(w4['sulphates'] < 1.17) & (w4['sulphates'] > 0.15)]
new5_w4
sb.boxplot(w4['sulphates'])
plt.show()
sb.boxplot(new1_w4['sulphates'])
plt.show()
upper_limit = w4['sulphates'].mean() + 3*w4['sulphates'].std()
lower_limit = w4['sulphates'].mean() – 3*w4['sulphates'].std()
w4['sulphates'] = np.where(
w4['sulphates']>upper_limit,
upper_limit,
np.where(
w4['sulphates']<lower_limit,
lower_limit,
w4['sulphates']))
```

```
w4['sulphates'].describe()
sb.boxplot(w4['sulphates'])
plt.show()

print("Highest value",w4['alcohol'].mean() + 3*w4['alcohol'].std())
print("Lowest value",w4['alcohol'].mean() - 3*w4['alcohol'].std())
w4[(w4['alcohol'] > 13.62) | (w4['alcohol'] < 7.23)]
new6_w4 = w4[(w4['alcohol'] < 13.62) & (w4['alcohol'] > 7.23)]
new6_w4
sb.boxplot(w4['alcohol'])
plt.show()
sb.boxplot(new1_w4['alcohol'])
plt.show()
upper_limit = w4['alcohol'].mean() + 3*w4['alcohol'].std()
lower_limit = w4['alcohol'].mean() – 3*w4['alcohol'].std()
w4['alcohol'] = np.where(
w4['alcohol']>upper_limit,
upper_limit,
np.where(
w4['alcohol']<lower_limit,
lower_limit,
w4['alcohol']))
w4['alcohol'].describe()
sb.boxplot(w4['alcohol'])
plt.show()

print("Highest value",w4['quality'].mean() + 3*w4['quality'].std())
print("Lowest value",w4['quality'].mean() - 3*w4['quality'].std())
w4[(w4['quality'] > 8.06) | (w4['quality'] < 3.21)]
new7_w4 = w4[(w4['quality'] < 8.06) & (w4['quality'] > 3.21)]
new7_w4
sb.boxplot(w4['quality'])
plt.show()
sb.boxplot(new1_w4['quality'])
plt.show()
upper_limit = w4['quality'].mean() + 3*w4['quality'].std()
lower_limit = w4['quality'].mean() – 3*w4['quality'].std()
w4['quality'] = np.where(
w4['quality']>upper_limit,
upper_limit,
```

```
np.where(
w4['quality']<lower_limit,
lower_limit,
w4['quality']))
w4['quality'].describe()
sb.boxplot(w4['quality'])
plt.show()

x2=w4.drop('quality', axis=1)
y2=w4.quality
x2_train,x2_test,y2_train, y2_test= train_test_split(x2,y2, test_size=0.2,
random_state=42)
sb.heatmap(w4.corr() > 0.5, annot=True, cbar=False)
plt.show()
reg = LinearRegression().fit(x2,y2)
reg.score(x2,y2)
```