

tache 1 code utiliser :

```
import pandas as pd

def calculer_stats(df, colonnes):
    resultats = {}

    for col in colonnes:
        moyenne = df[col].mean()
        mediane = df[col].median()
        q1 = df[col].quantile(0.25)
        q3 = df[col].quantile(0.75)
        iqr = q3 - q1
        limite_sup = q3 + 1.5 * iqr
        resultats[col] = {
            'Moyenne': moyenne,
            'Mediane': mediane,
            '1er Quartile': q1,
            '3eme Quartile': q3,
            'Limite Superieure': limite_sup
        }

    return resultats

chemin_fichier = '/jfreechart-test-stats.csv'
donnees = pd.read_csv(chemin_fichier)
donnees.columns = donnees.columns.str.strip()
# Métriques à calculer
colonnes = ['TLOC', 'WMC', 'TASSERT']

# Calcul des statistiques
resultats_stats = calculer_stats(donnees, colonnes)
print(resultats_stats)
```

tache 2 code :

```
import pandas as pd
from scipy.stats import pearsonr, linregress

file_path = '/jfreechart-test-stats.csv'
data = pd.read_csv(file_path)
data.rename(columns=lambda x: x.strip(), inplace=True)

pairs = [('TLOC', 'TASSERT'), ('TLOC', 'WMC'), ('WMC', 'TASSERT')]

for x, y in pairs:
    correlation, _ = pearsonr(data[x], data[y])
```

```
regression = linregress(data[x], data[y])

print(f"Corrélation entre {x} et {y}: {correlation:.2f}")
print(f"Régression linéaire pour {x} vs {y}: pente = {regression.slope:.2f},  
intercept = {regression.intercept:.2f}\n")
```