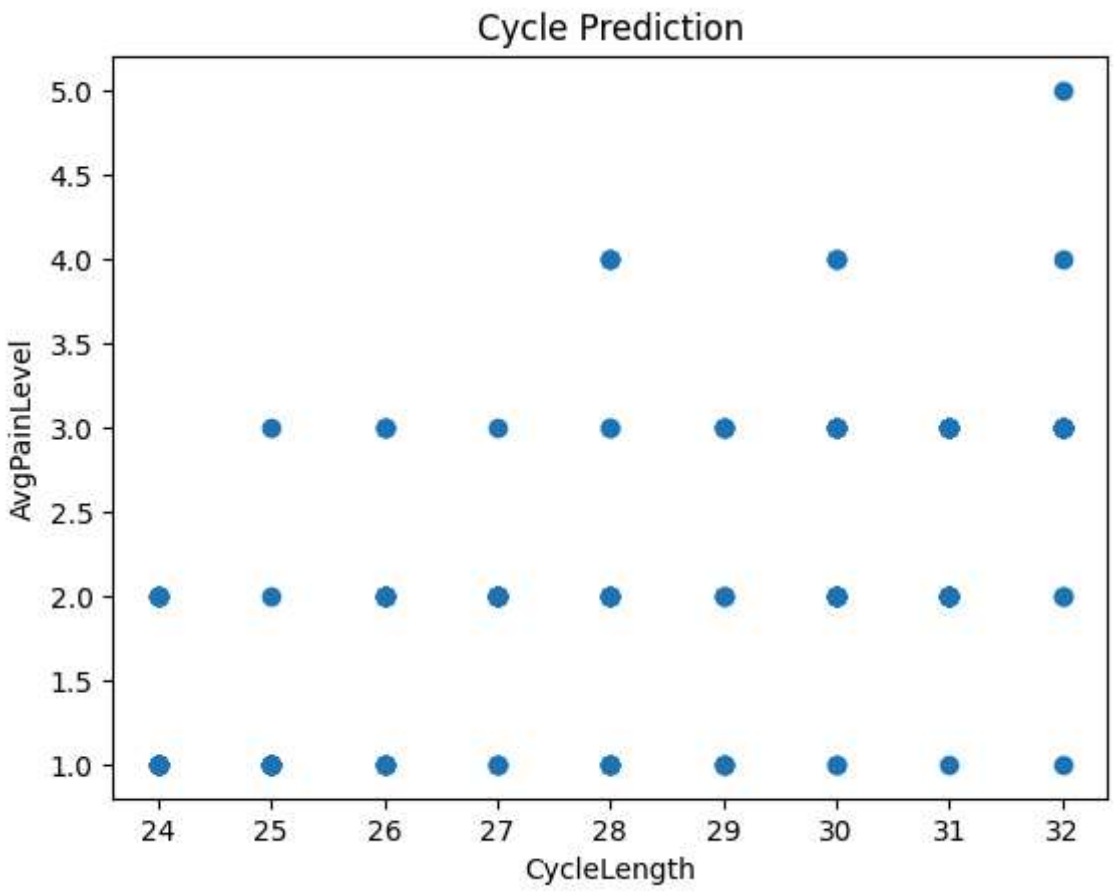
Importing libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
df=pd.read_csv("period_data.csv")
df.head(5)
```

	CycleLength	BleedingDays	AvgPainLevel
0	30	6	3
1	27	4	2
2	31	5	2
3	28	3	1
4	30	7	3

Next steps: Generate code with df | New interactive sheet

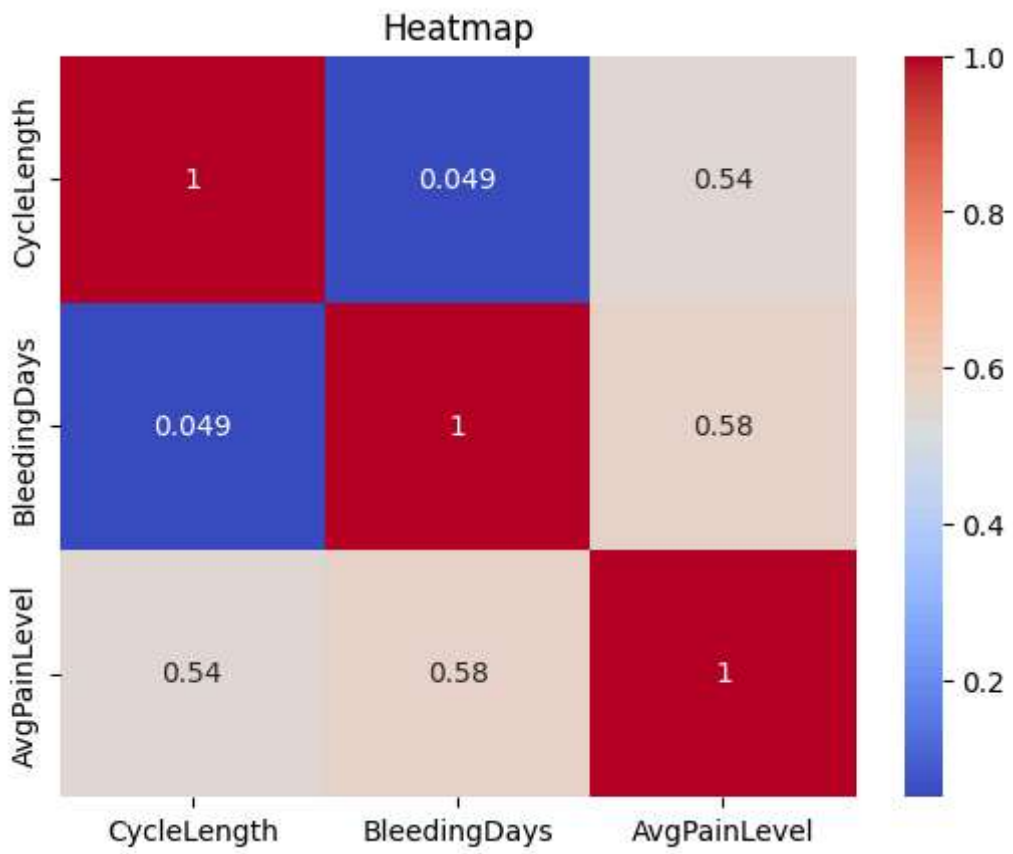
```
plt.scatter(df['CycleLength'],df['AvgPainLevel'])
plt.xlabel('CycleLength')
plt.ylabel('AvgPainLevel')
plt.title('Cycle Prediction')
plt.show()
```



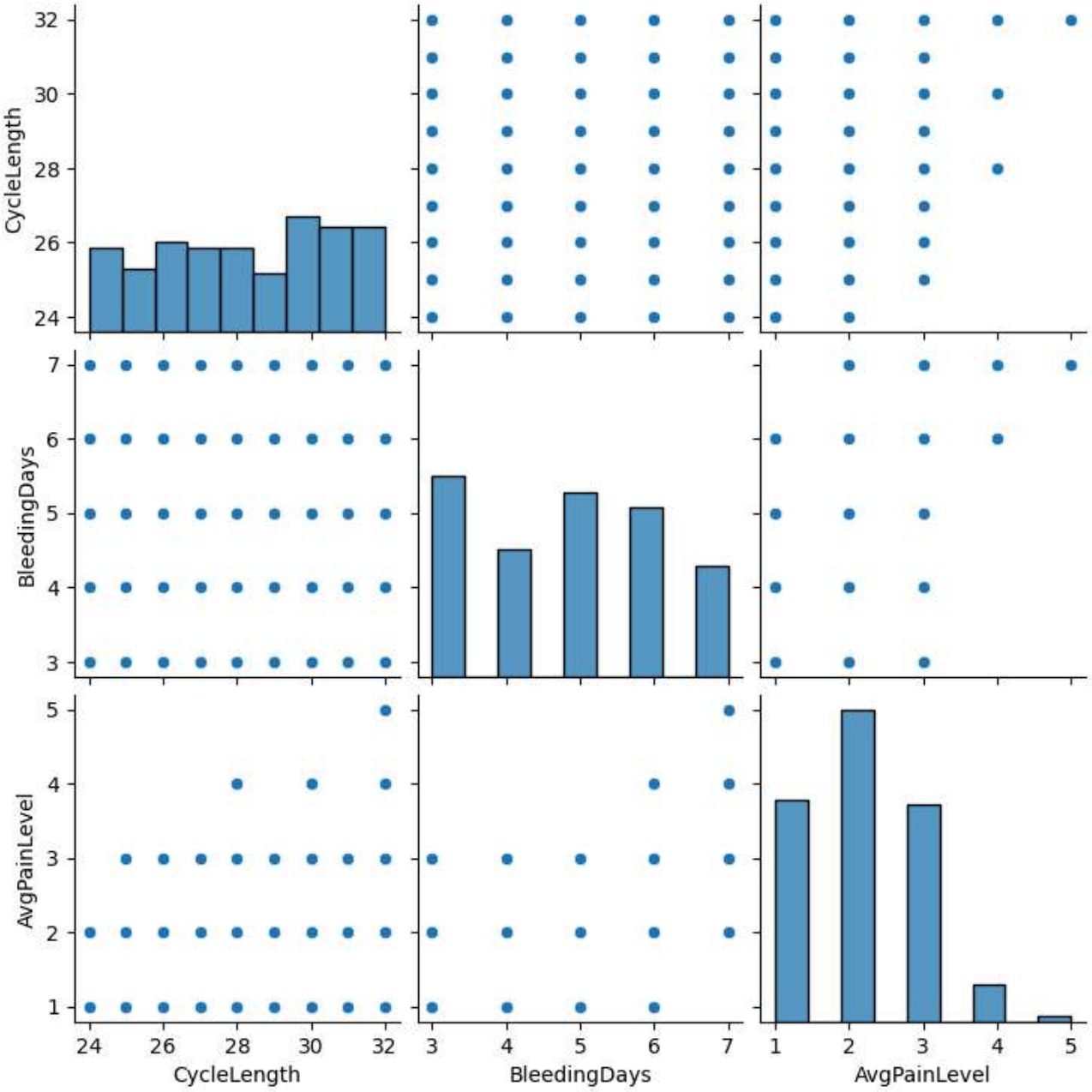

```
df.corr()
```

	CycleLength	BleedingDays	AvgPainLevel
CycleLength	1.000000	0.049166	0.541110
BleedingDays	0.049166	1.000000	0.578987
AvgPainLevel	0.541110	0.578987	1.000000

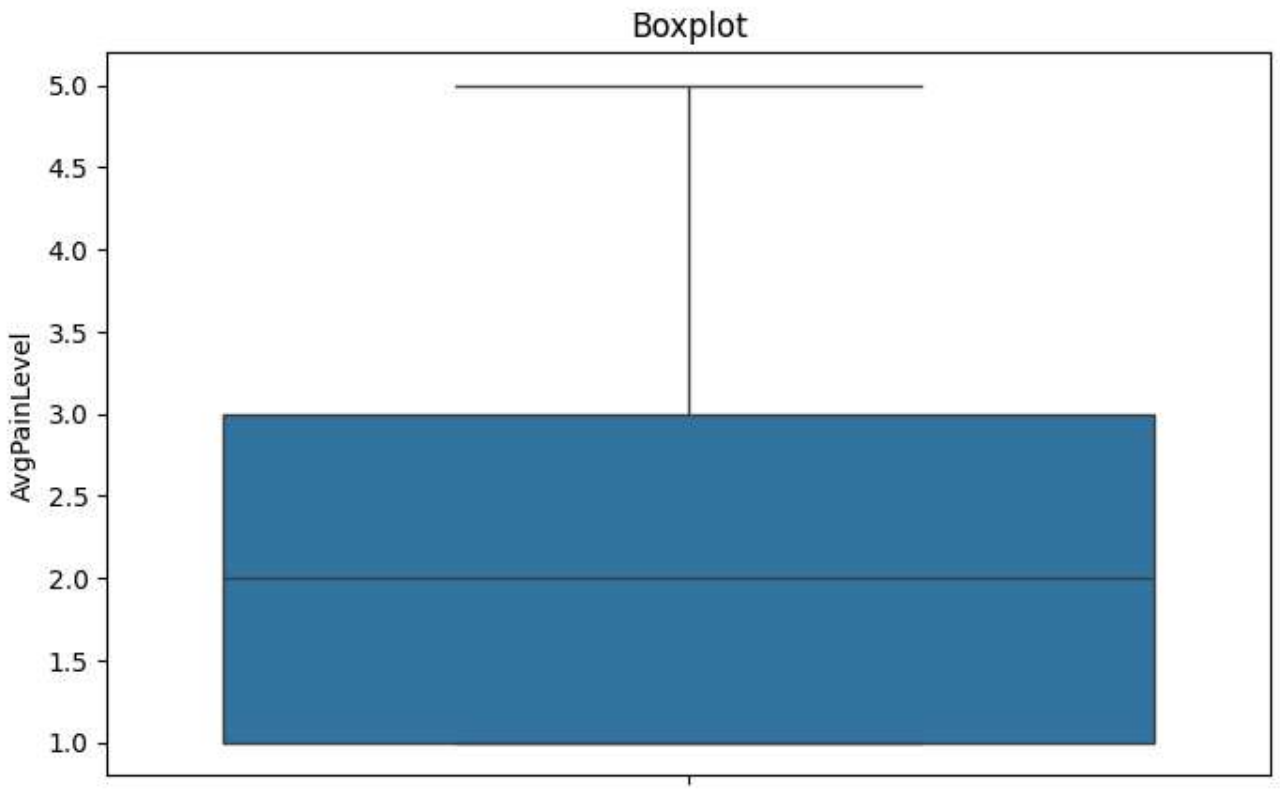
```
sns.heatmap(df.corr(),annot=True,cmap='coolwarm')
plt.title('Heatmap')
plt.show()
```




```
sns.pairplot(df)
plt.show()
```




```
plt.figure(figsize=(8,5))
sns.boxplot(df['AvgPainLevel'])
plt.title('Boxplot')
plt.show()
```




```
X=df[['CycleLength']]
y=df[['AvgPainLevel']]
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42)
X_train.shape,X_test.shape
y_train.shape,y_test.shape
```

```
((105, 1), (45, 1))
```

```
model=LinearRegression()
model.fit(X_train,y_train)
model.intercept_
```

```
array([-3.31492206])
```

```
y_pred=model.predict(X_test)
print(y_test)
print(y_pred)
```

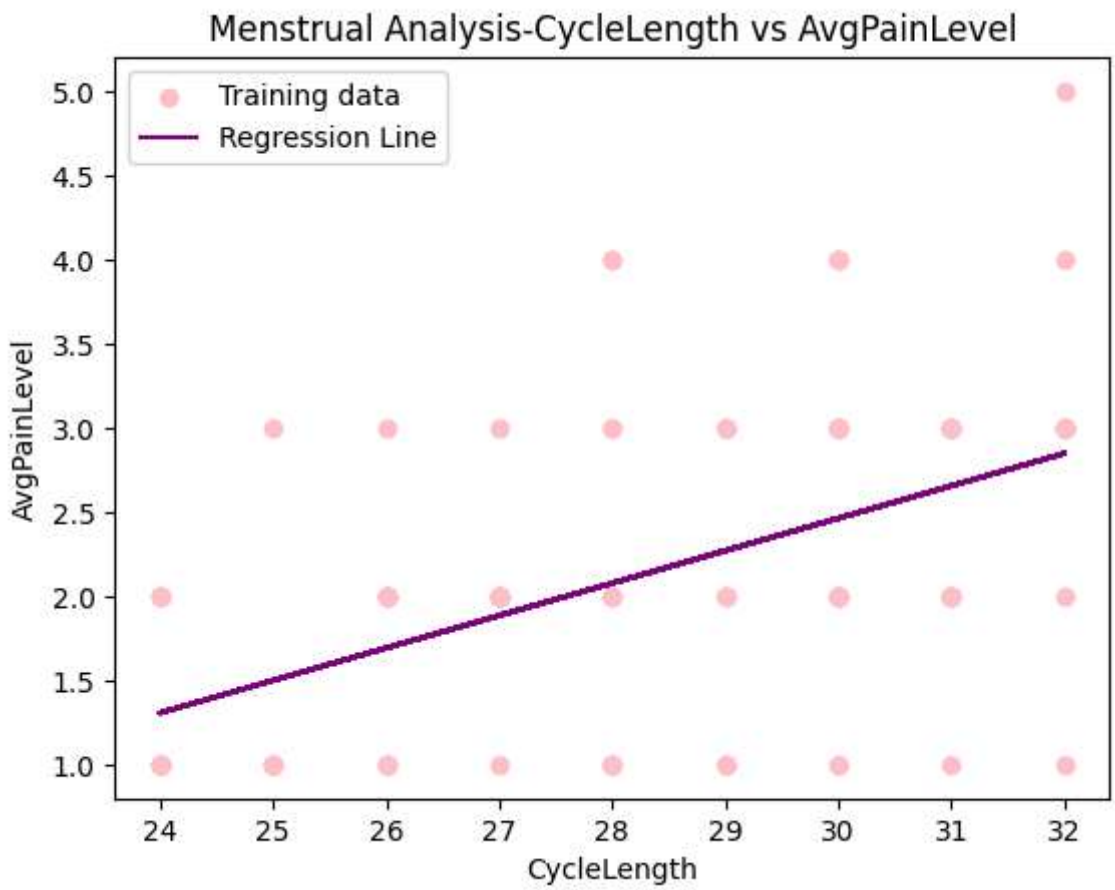
```
142             2
85              3
86              2
16              2
10              2
[[2.65529606]
 [1.49976997]
 [1.69235765]
 [1.30718229]
 [2.65529606]
 [2.46270838]
 [1.49976997]
 [1.69235765]
 [2.84788374]
 [1.30718229]
 [1.30718229]
 [1.69235765]
 [1.88494533]
 [1.88494533]
 [2.07753302]
 [1.49976997]
 [2.07753302]
 [1.30718229]
 [2.2701207 ]
 [1.88494533]
 [2.07753302]
 [2.46270838]
 [1.88494533]
 [1.30718229]
 [2.84788374]
 [2.84788374]
 [1.69235765]
 [2.2701207 ]
 [1.69235765]
 [1.69235765]
 [2.84788374]
 [1.49976997]
 [2.07753302]
 [2.65529606]
 [1.88494533]
 [1.88494533]
 [1.49976997]
 [2.84788374]
 [2.46270838]
 [2.07753302]
 [2.65529606]
 [2.84788374]
 [2.46270838]
 [2.65529606]
 [2.65529606]]
```

```
mae=mean_absolute_error(y_test,y_pred)
mse=mean_squared_error(y_test,y_pred)
r2=r2_score(y_test,y_pred)
print('mae',mae)
print('mse',mse)
```

```
print('r2 score',r2)
```

```
mae 0.5342122513290516
mse 0.43848544794380473
r2 score 0.26858893567857944
```

```
plt.scatter(X_train,y_train,color='pink',label='Training data')
plt.plot(X_train,model.predict(X_train),color='purple',label='Regression Line')
plt.title('Menstrual Analysis-CycleLength vs AvgPainLevel')
plt.xlabel('CycleLength')
plt.ylabel('AvgPainLevel')
plt.legend()
plt.show()
```




```
new_salary=model.predict([[35]])
```