```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
sns.set(color_codes=True)
%matplotlib inline
```

```
df=pd.read_csv('HR Absenteeism data.csv')
```

```
df.head(10)
```

	EmployeeNumber	Surname	GivenName	Gender	City	JobTitle	DepartmentName	StoreLocation	Division	
0	1	Gutierrez	Molly	F	Burnaby	Baker	Bakery	Burnaby	Stores	32
1	2	Hardwick	Stephen	M	Courtenay	Baker	Bakery	Nanaimo	Stores	40
2	3	Delgado	Chester	M	Richmond	Baker	Bakery	Richmond	Stores	48
3	4	Simon	Irene	F	Victoria	Baker	Bakery	Victoria	Stores	44
4	5	Delvalle	Edward	M	New Westminster	Baker	Bakery	New Westminster	Stores	35
5	6	Jones	Ernie	M	Richmond	Baker	Bakery	Richmond	Stores	48
6	7	Buford	Ralph	M	Vancouver	Accounting Clerk	Accounting	Vancouver	FinanceAndAccounting	50
7	8	Lee	Gregory	M	Sechelt	Baker	Bakery	West Vancouver	Stores	36
8	9	Smith	Jerry	M	New Westminster	Baker	Bakery	New Westminster	Stores	58
9	10	Beard	Robert	M	Vancouver	Accounting Clerk	Accounting	Vancouver	FinanceAndAccounting	39

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8336 entries, 0 to 8335
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   EmployeeNumber  8336 non-null   int64  
 1   Surname         8336 non-null   object 
 2   GivenName       8336 non-null   object 
 3   Gender          8336 non-null   object 
 4   City            8336 non-null   object 
 5   JobTitle        8336 non-null   object 
 6   DepartmentName  8336 non-null   object 
 7   StoreLocation   8336 non-null   object 
 8   Division        8336 non-null   object 
 9   Age             8336 non-null   float64
 10  LengthService   8336 non-null   float64
 11  AbsentHours     8336 non-null   float64
 12  BusinessUnit    8336 non-null   object 
dtypes: float64(3), int64(1), object(9)
memory usage: 846.8+ KB
```

```
df.describe()
```

EmployeeNumber	Age	LengthService	AbsentHours

```
df.corr()
```

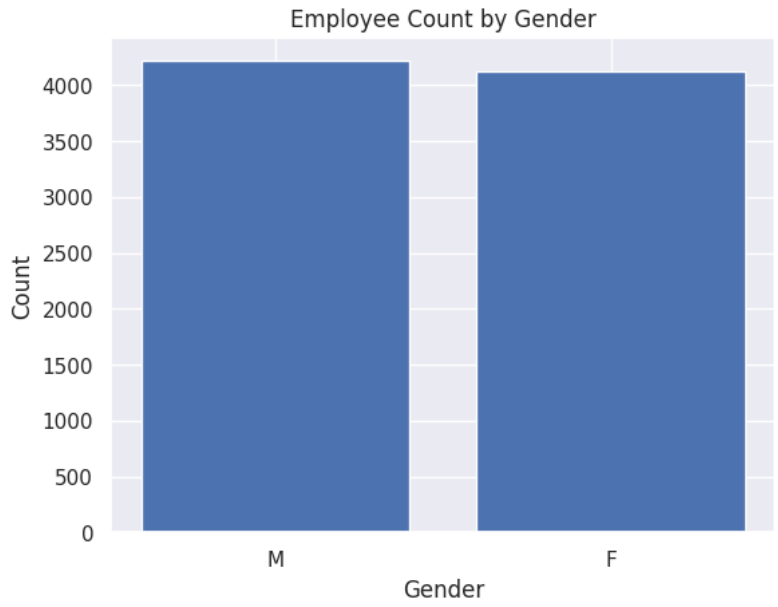
```
<ipython-input-6-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a fu
  df.corr()
```

	EmployeeNumber	Age	LengthService	AbsentHours
EmployeeNumber	1.000000	-0.018445	-0.119631	0.007418
Age	-0.018445	1.000000	0.053104	0.830234
LengthService	-0.119631	0.053104	1.000000	-0.044202
AbsentHours	0.007418	0.830234	-0.044202	1.000000

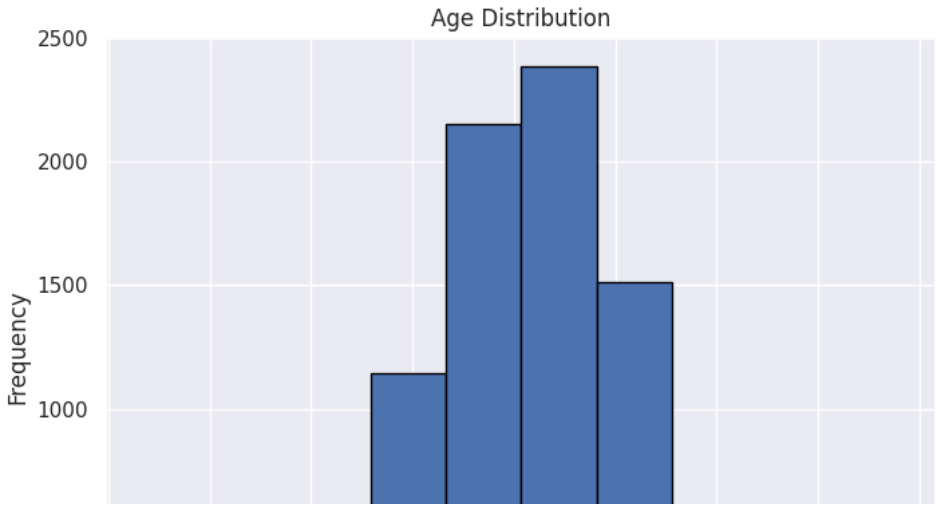
Visualisation

```
# Group the data by gender and get the count
gender_counts = df['Gender'].value_counts()
```

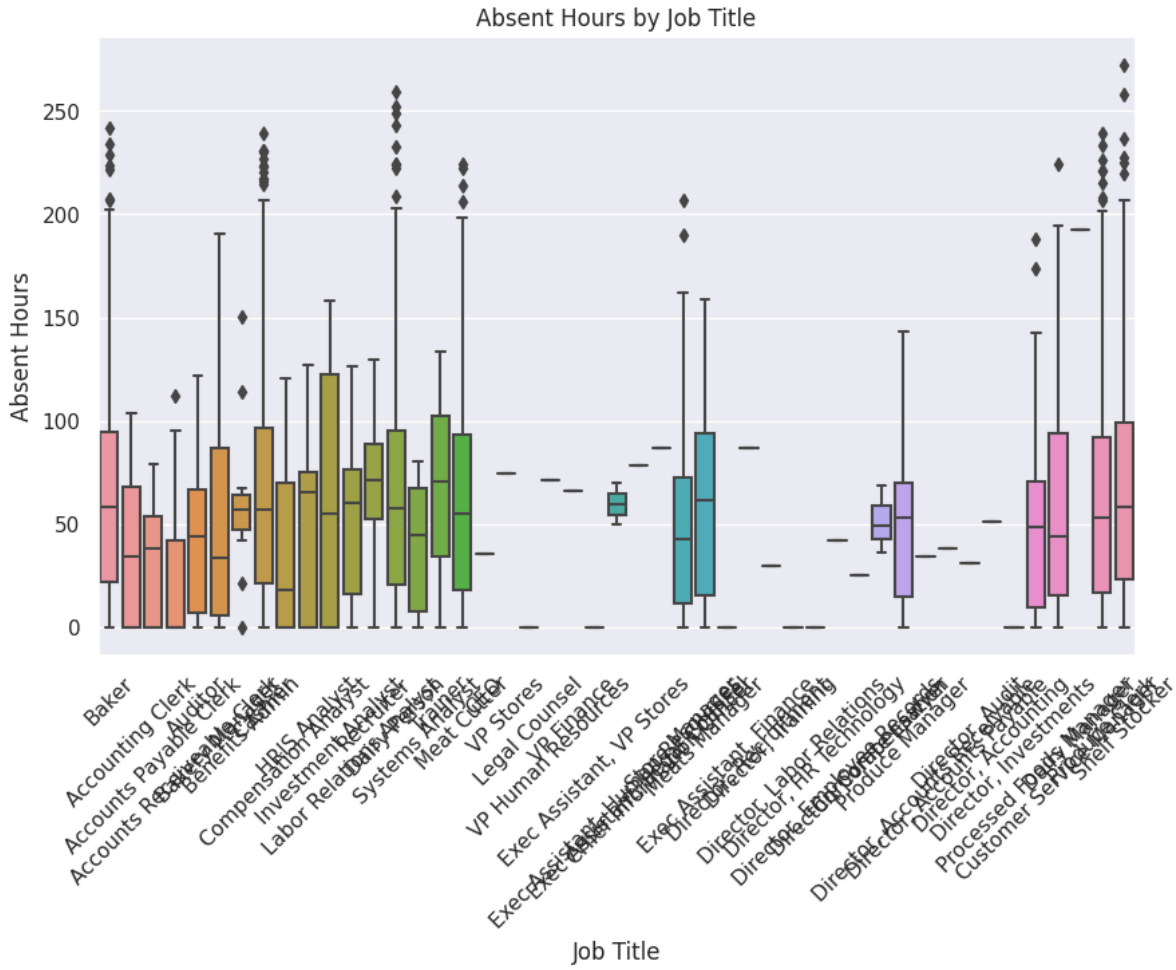
```
plt.bar(gender_counts.index, gender_counts.values)
plt.xlabel('Gender')
plt.ylabel('Count')
plt.title('Employee Count by Gender')
plt.show()
```




```
plt.figure(figsize=(8, 6))
plt.hist(df['Age'], bins=10, edgecolor='black')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Age Distribution')
plt.show()
```

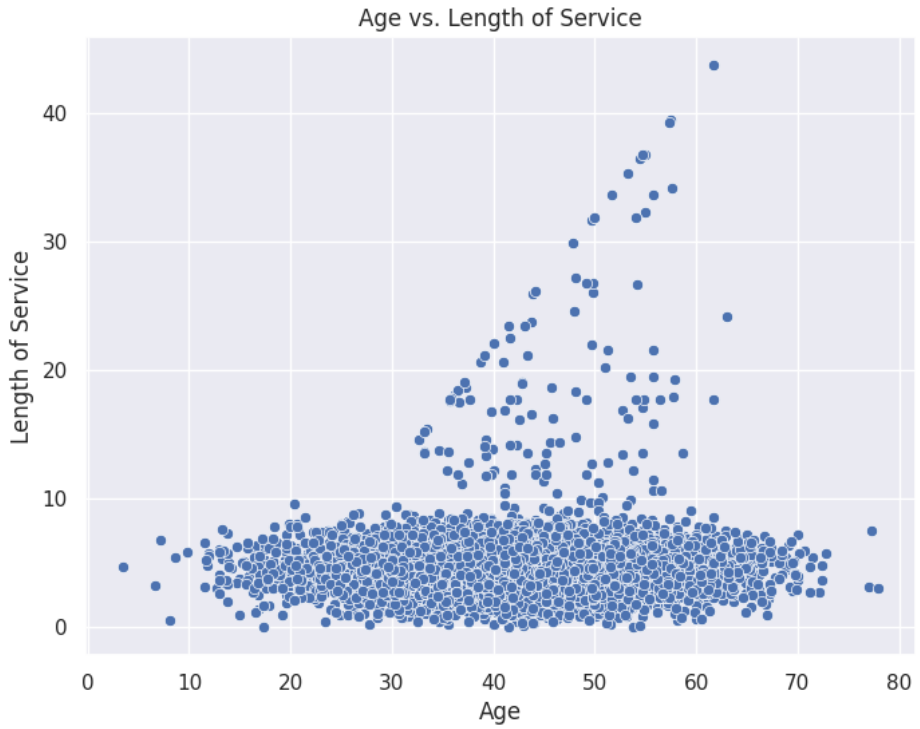



```
# Box Plot - AbsentHours by JobTitle
plt.figure(figsize=(10, 6))
sns.boxplot(x='JobTitle', y='AbsentHours', data=df)
plt.xlabel('Job Title')
plt.ylabel('Absent Hours')
plt.title('Absent Hours by Job Title')
plt.xticks(rotation=45)
plt.show()
```

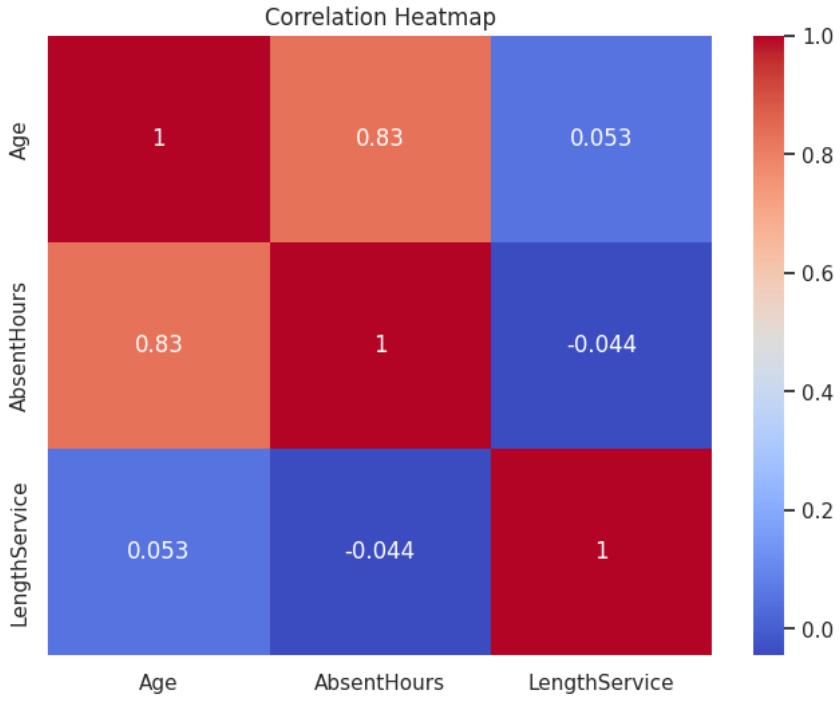



```
# Scatter Plot - Age vs. LengthService
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Age', y='LengthService', data=df)
plt.xlabel('Age')
plt.ylabel('Length of Service')
```

```
plt.title('Age vs. Length of Service')
plt.show()
```

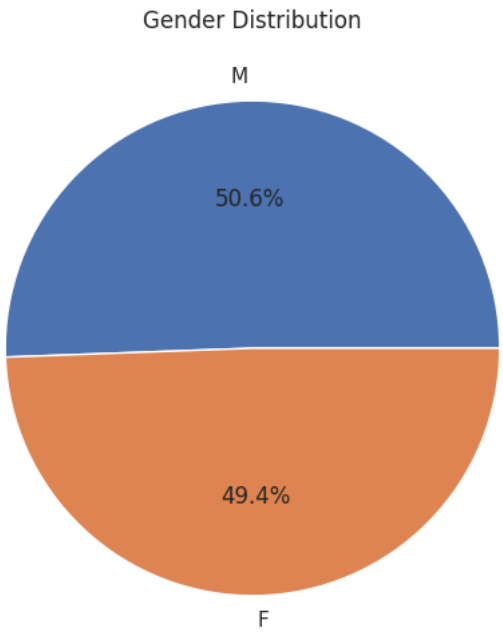



```
# Heatmap - Correlation between Age, AbsentHours, and LengthService
subset = df[['Age', 'AbsentHours', 'LengthService']]
plt.figure(figsize=(8, 6))
sns.heatmap(subset.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



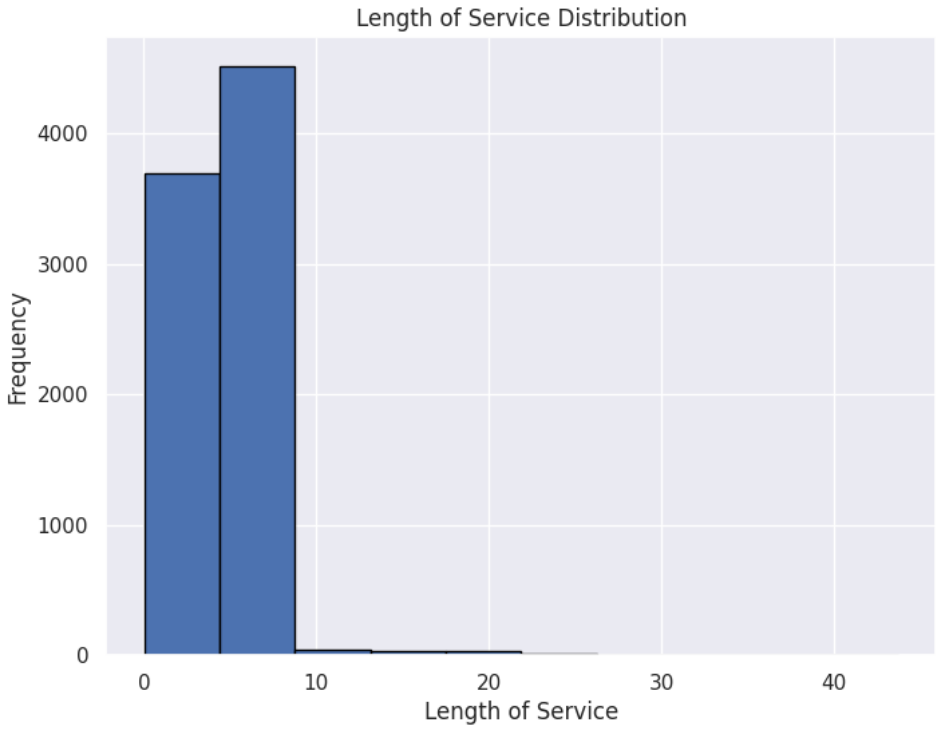

```
# Pie Chart - Gender
gender_counts = df['Gender'].value_counts()
plt.figure(figsize=(8, 6))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%')
plt.title('Gender Distribution')
```

```
plt.show()
```

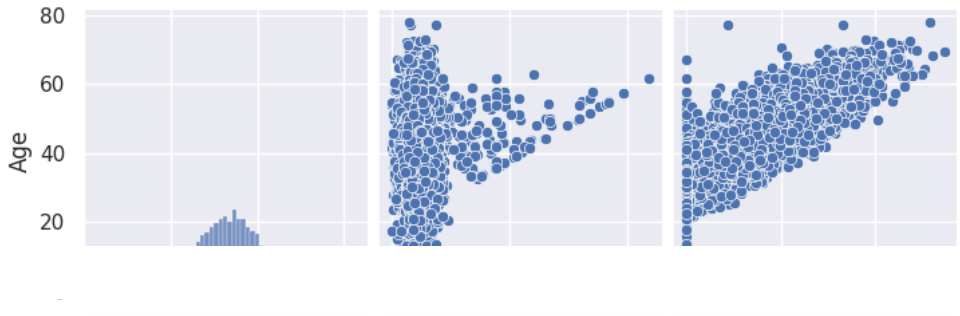



```
# Bar Chart - DepartmentName
department_counts = df['DepartmentName'].value_counts()
plt.figure(figsize=(10, 6))
plt.bar(department_counts.index, department_counts.values)
plt.xlabel('Department Name')
plt.ylabel('Count')
plt.title('Department Distribution')
plt.xticks(rotation=45)
plt.show()
```

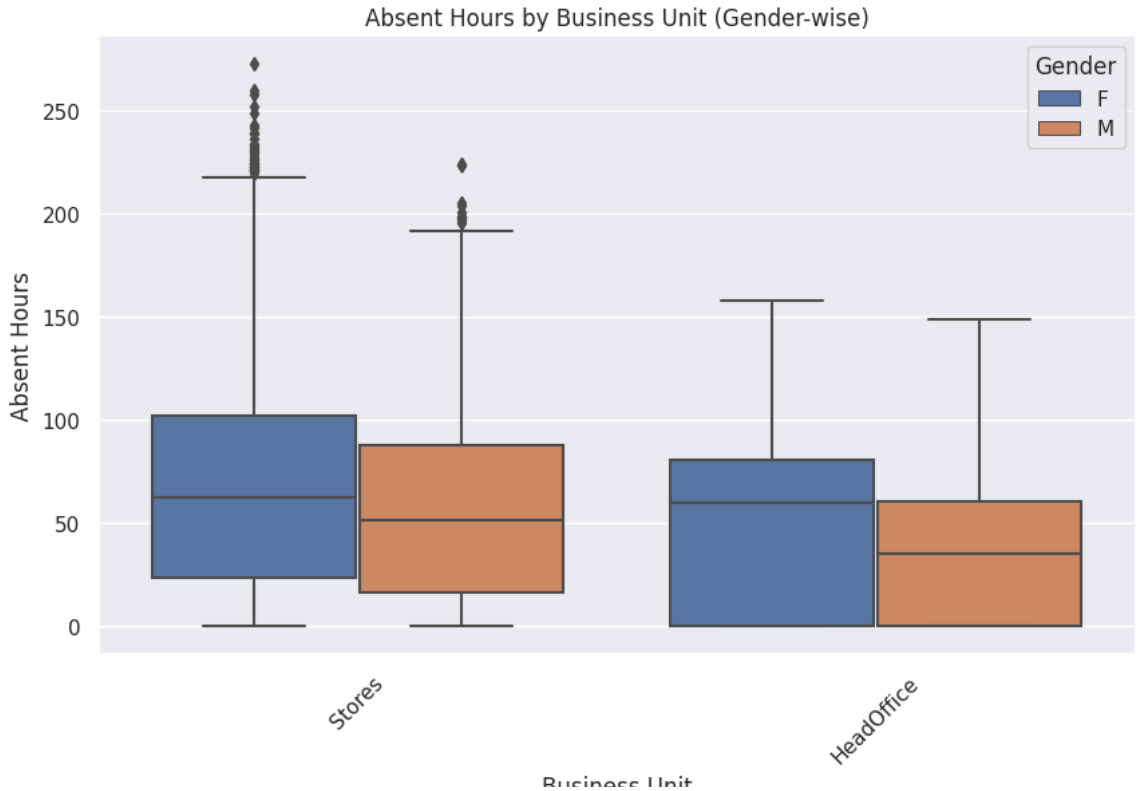
```
# Histogram - LengthService
plt.figure(figsize=(8, 6))
plt.hist(df['LengthService'], bins=10, edgecolor='black')
plt.xlabel('Length of Service')
plt.ylabel('Frequency')
plt.title('Length of Service Distribution')
plt.show()
```



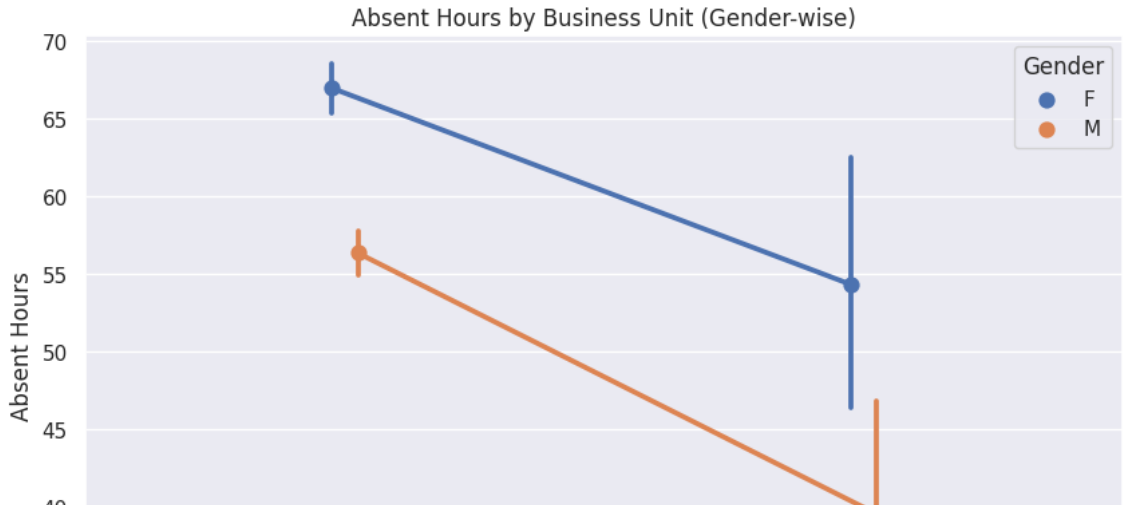

```
# Select numerical columns for the pair plot
numerical_columns = ['Age', 'LengthService', 'AbsentHours']

# Create the pair plot
sns.pairplot(df[numerical_columns])
plt.show()
```




```
# Box Plot - BusinessUnit vs AbsentHours with Gender as hue
plt.figure(figsize=(10, 6))
sns.boxplot(x='BusinessUnit', y='AbsentHours', hue='Gender', data=df)
plt.xlabel('Business Unit')
plt.ylabel('Absent Hours')
plt.title('Absent Hours by Business Unit (Gender-wise)')
plt.xticks(rotation=45)
plt.legend(title='Gender')
plt.show()
```




```
# Point Plot - BusinessUnit vs AbsentHours with Gender
plt.figure(figsize=(10, 6))
sns.pointplot(x='BusinessUnit', y='AbsentHours', hue='Gender', data=df, dodge=True)
plt.xlabel('Business Unit')
plt.ylabel('Absent Hours')
plt.title('Absent Hours by Business Unit (Gender-wise)')
plt.xticks(rotation=45)
plt.legend(title='Gender')
plt.show()
```




```
# Calculate the top absent titles
top_absent_titles = df.groupby('JobTitle')['AbsentHours'].sum().nlargest(10)

# Filter the dataset for the top absent titles
filtered_df = df[df['JobTitle'].isin(top_absent_titles.index)]

# Bar Chart - AbsentHours vs JobTitle (Top Absent Titles)
plt.figure(figsize=(10, 6))
plt.bar(filtered_df['JobTitle'], filtered_df['AbsentHours'])
plt.xlabel('Job Title')
plt.ylabel('Absent Hours')
plt.title('Absent Hours vs Job Title (Top Absent Titles)')
plt.xticks(rotation=45)
plt.show()
```