ACTVITY-3 PYTHON

STUDENT PARENT DATASETS

GROUP :

1:ANUSHA M
2:SOWJANYA
3:ANUSHA G V
4:KRUTHIKA H N

```
mport necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
file_path = '~/Downloads/student_data.csv' # Update this path based on your dataset location
df = pd.read_csv(file_path)

# Display the first few rows of the dataset to verify the data
print(df.head())
```

OUTPUT:

```
School sex  age address famsize Pstatus  Medu  Fedu     Mjob      Fjob  reason guardian  ...  internet
romantic  famrel freetime goout Dalc Walc health absences  G1  G2  G3
0    GP   F   18      U    GT3      A    4    4  at_home   teacher  course   mother ...       no       no
4      3    4    1    1     3       6   5   6   6
1    GP   F   17      U    GT3      T    1    1  at_home    other  course   father ...      yes       no
5      3    3    1    1     3       4   5   5   6
2    GP   F   15      U    LE3      T    1    1  at_home    other   other   mother ...      yes       no
4      3    2    2    3     3      10   7   8  10
3    GP   F   15      U    GT3      T    4    2  health  services   home   mother ...      yes      yes
3      2    2    1    1     5       2  15  14  15
4    GP   F   16      U    GT3      T    3    3  other    other    home   father ...      no       no      4
3    2    1    2     5       4   6  10  10

[5 rows x 33 column]
```

```
# Check for missing values in the dataset
print(df.isnull().sum())
```

output:

```
School       0
sex          0
age          0
address      0
famsize      0
Pstatus      0
Medu         0
Fedu         0
Mjob         0
Fjob         0
reason       0
guardian     0
traveltime   0
studytime    0
failures     0
schoolsup    0
famsup       0
paid         0
activities   0
nursery      0
higher       0
internet     0
romantic     0
famrel       0
freetime     0
goout        0
Dalc         0
Walc         0
health       0
absences     0
G1           0
G2           0
G3           0
dtype: int64
```

```
# Display the column names for reference
print(df.columns)
```

output:

```
Index(['school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Medu', 'Fedu',
       'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime', 'studytime',
       'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery',
       'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc',
       'Walc', 'health', 'absences', 'G1', 'G2', 'G3'],
      dtype='object')
```

```
# Set the style for seaborn
sns.set(style="whitegrid")

# 1. Visualization: Countplot for Gender ('sex' column)
plt.figure(figsize=(8, 6))
sns.countplot(x='sex', data=df, palette='coolwarm')
plt.title('Count of Male and Female Students')
plt.savefig('countplot_sex.png') # Save the figure as PNG
plt.show()

#2.Visualization: Family size distribution ('famsize' column)
plt.figure(figsize=(8, 6))
sns.countplot(x='famsize', data=df, palette='Set2')
plt.title('Family Size Distribution (GT3 vs LE3)')
plt.savefig('countplot_famsize.png')
plt.show()
```

output:

```
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` for the same effect.

  sns.countplot(x='sex', data=df, palette='coolwarm')
/home/anusha/pythonProject/studentData.py:31: 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` for the same effect.

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

```
#3. Visualization: Education of Parents ('Medu' and 'Fedu' columns)
plt.figure(figsize=(8, 6))
sns.histplot(df['Medu'], bins=5, color='blue', label='Mother Education', kde=True)
sns.histplot(df['Fedu'], bins=5, color='red', label='Father Education', kde=True)
plt.title('Distribution of Parental Education (Mother and Father)')
plt.xlabel('Education Level (0-4)')
plt.ylabel('Frequency')
plt.legend()
plt.savefig('histplot_parental_education.png')
plt.show()

# 4. Visualization: Distribution of Occupations ('Mjob' and 'Fjob' columns)
plt.figure(figsize=(10, 6))
sns.countplot(x='Mjob', data=df, palette='viridis')
plt.title('Mother Occupations')
plt.xticks(rotation=45)
plt.savefig('countplot_mother_occupation.png')
plt.show()
```

output:

```
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` for the same effect.

  sns.countplot(x='Mjob', data=df, palette='viridis')
```

```
# plt.figure(figsize=(10, 6))
sns.countplot(x='Fjob', data=df, palette='inferno')
plt.title('Father Occupations')
plt.xticks(rotation=45)
plt.savefig('countplot_father_occupation.png')
plt.show()

# # 5. Visualization: Family Relationship ('famrel' column)
plt.figure(figsize=(8, 6))
sns.countplot(x='famrel', data=df, palette='Spectral')
plt.title('Quality of Family Relationships (1 = very bad, 5 = excellent)')
plt.savefig('countplot_famrel.png')
plt.show()
```

output:

```
 sns.countplot(x='Fjob', data=df, palette='inferno')
/home/anusha/pythonProject/studentData.py:64: 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` for the same effect.

  sns.countplot(x='famrel', data=df, palette='Spectral')
```

```
# # 6. Visualization: Health Status of Students ('health' column)
plt.figure(figsize=(8, 6))
sns.countplot(x='health', data=df, palette='magma')
plt.title('Student Health Status (1 = very bad, 5 = excellent)')
plt.savefig('countplot_health.png')
plt.show()

# # 7. Visualization: Final Grades ('G1', 'G2', 'G3')
plt.figure(figsize=(8, 6))
sns.histplot(df['G1'], color='blue', label='G1', kde=True, bins=10)
sns.histplot(df['G2'], color='orange', label='G2', kde=True, bins=10)
sns.histplot(df['G3'], color='green', label='G3', kde=True, bins=10)
plt.title('Distribution of Grades (G1, G2, G3)')
plt.xlabel('Grades')
plt.ylabel('Frequency')
plt.legend()
plt.savefig('histplot_grades.png')
plt.show()

# # 8. Visualization: Correlation Heatmap (relationships between variables)
```

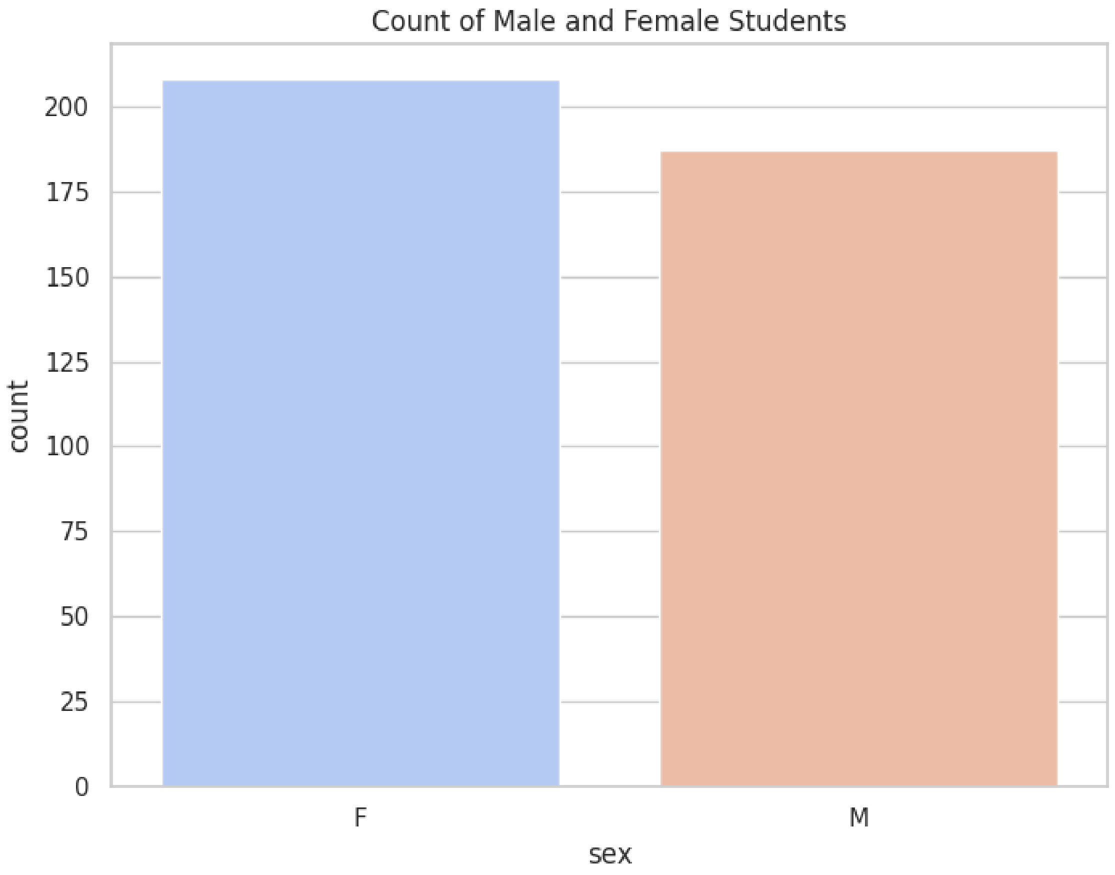
```
plt.figure(figsize=(12, 8))
corr = df.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.savefig('heatmap_correlation.png')
plt.show()
```

output:

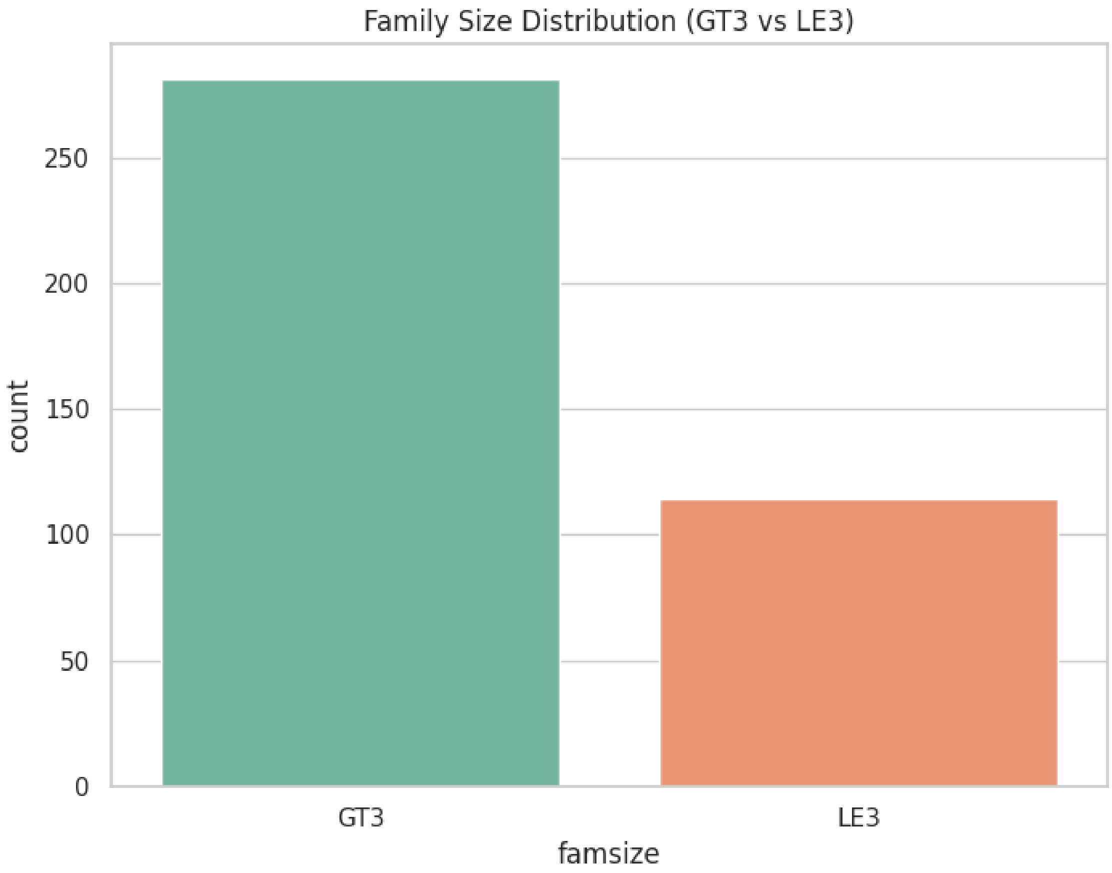
```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the
`x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x='health', data=df, palette='magma')
```

GRAPH

```
# 1. Visualization: Countplot for Gender ('sex' column)
plt.figure(figsize=(8, 6))
sns.countplot(x='sex', data=df, palette='coolwarm')
plt.title('Count of Male and Female Students')
plt.savefig('countplot_sex.png') # Save the figure as PNG
plt.show()
```

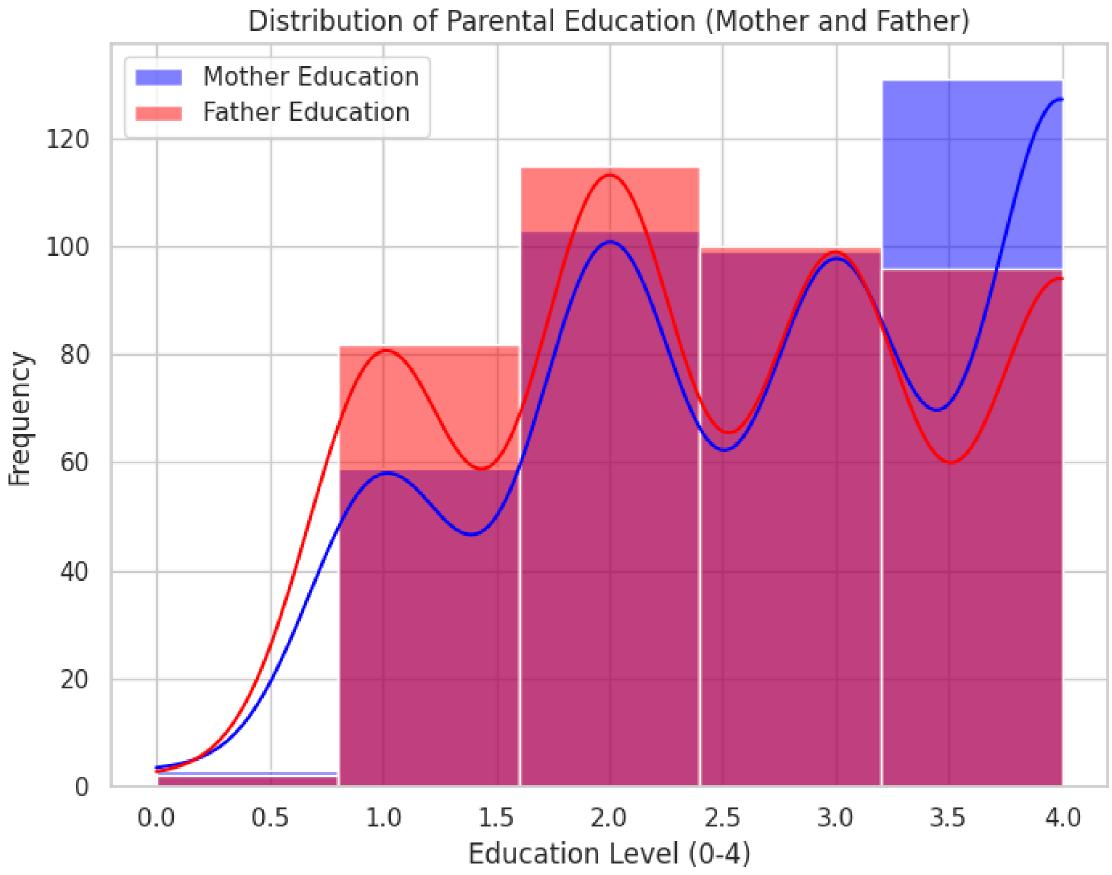


```
#2.Visualization: Family size distribution ('famsize' column)
plt.figure(figsize=(8, 6))
sns.countplot(x='famsize', data=df, palette='Set2')
plt.title('Family Size Distribution (GT3 vs LE3)')
plt.savefig('countplot_famsize.png')
plt.show()
```

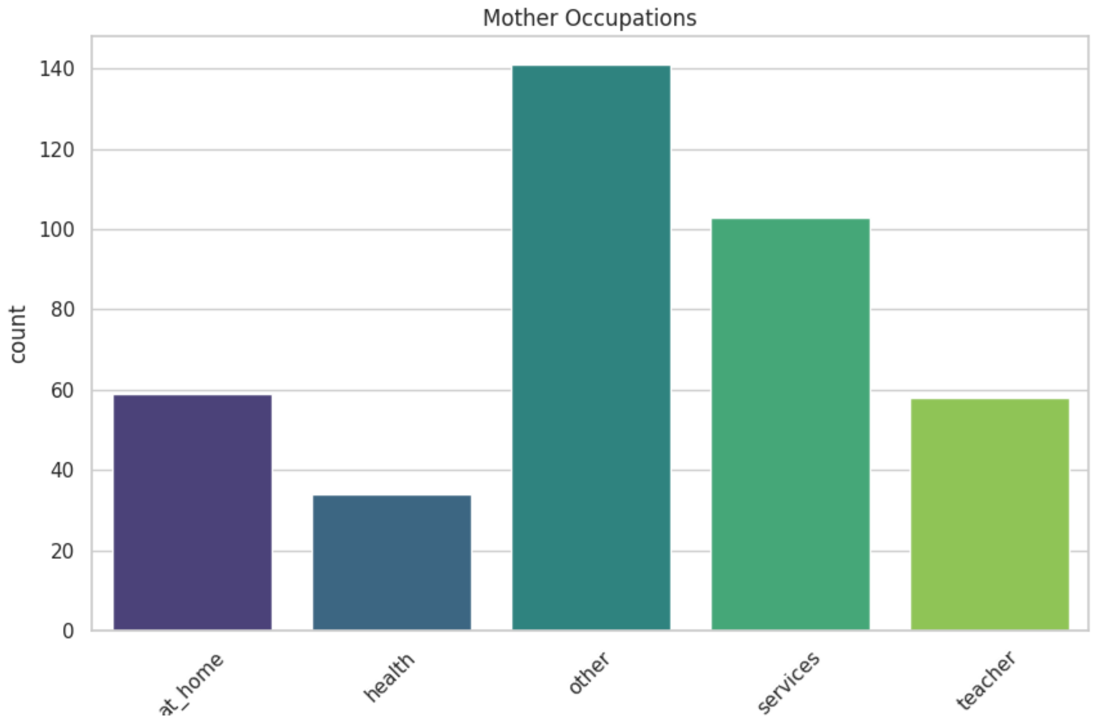


3. Visualization: Education of Parents ('Medu' and 'Fedu' columns)

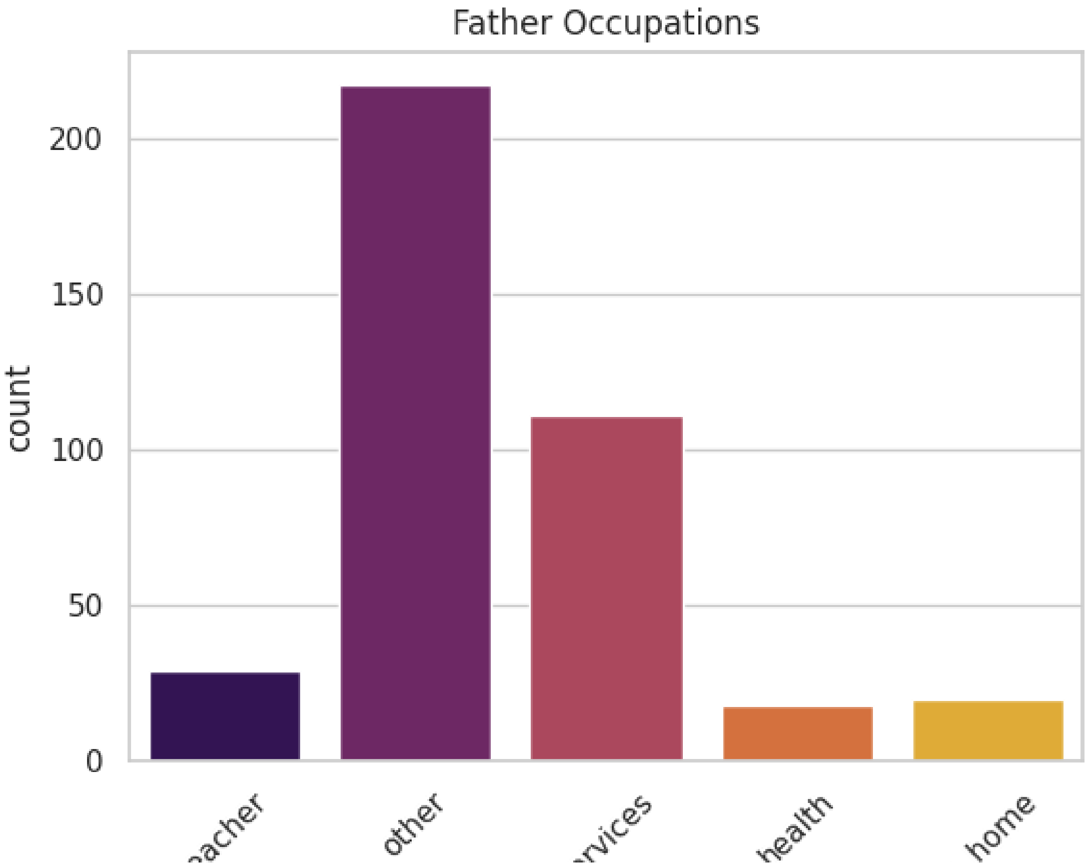
```
plt.figure(figsize=(8, 6))
sns.histplot(df['Medu'], bins=5, color='blue', label='Mother Education', kde=True)
sns.histplot(df['Fedu'], bins=5, color='red', label='Father Education', kde=True)
plt.title('Distribution of Parental Education (Mother and Father)')
plt.xlabel('Education Level (0-4)')
plt.ylabel('Frequency')
plt.legend()
plt.savefig('histplot_parental_education.png')
plt.show()
```



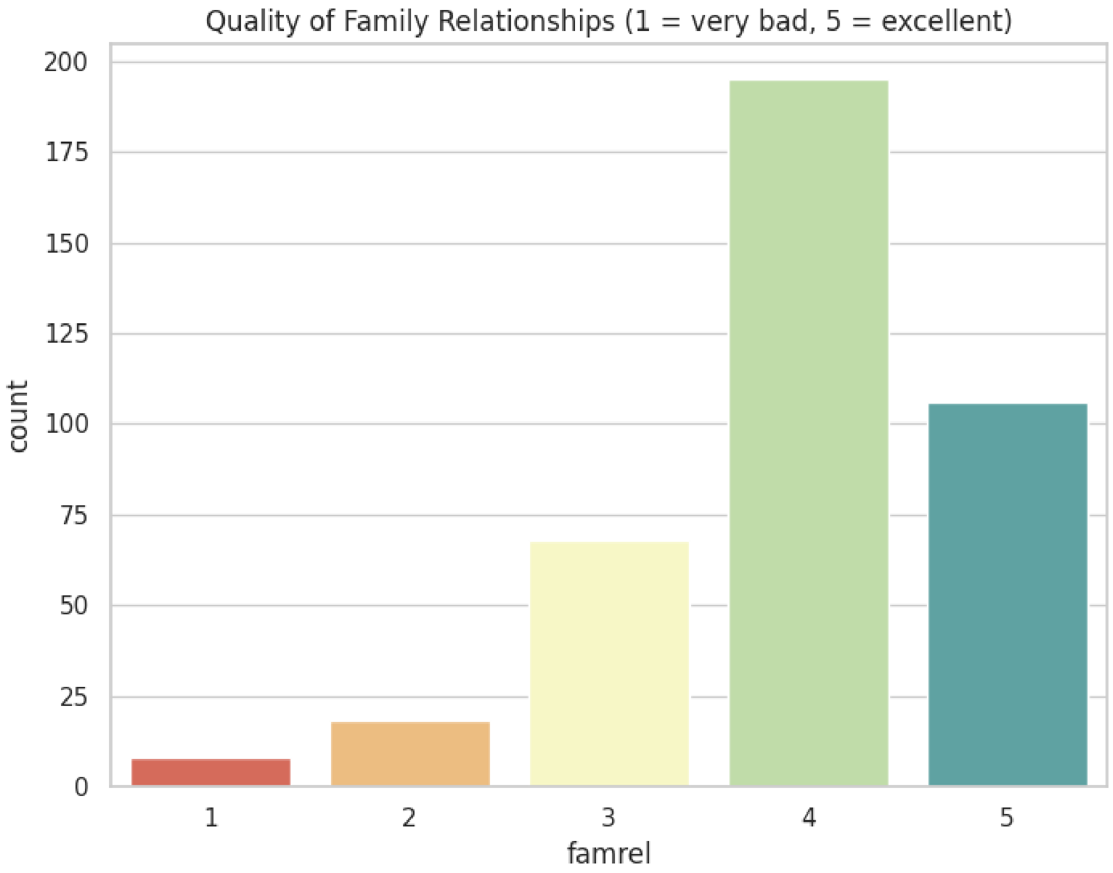
```
# 4. Visualization: Distribution of Occupations ('Mjob' and 'Fjob' columns)
plt.figure(figsize=(10, 6))
sns.countplot(x='Mjob', data=df, palette='viridis')
plt.title('Mother Occupations')
plt.xticks(rotation=45)
plt.savefig('countplot_mother_occupation.png')
plt.show()
```



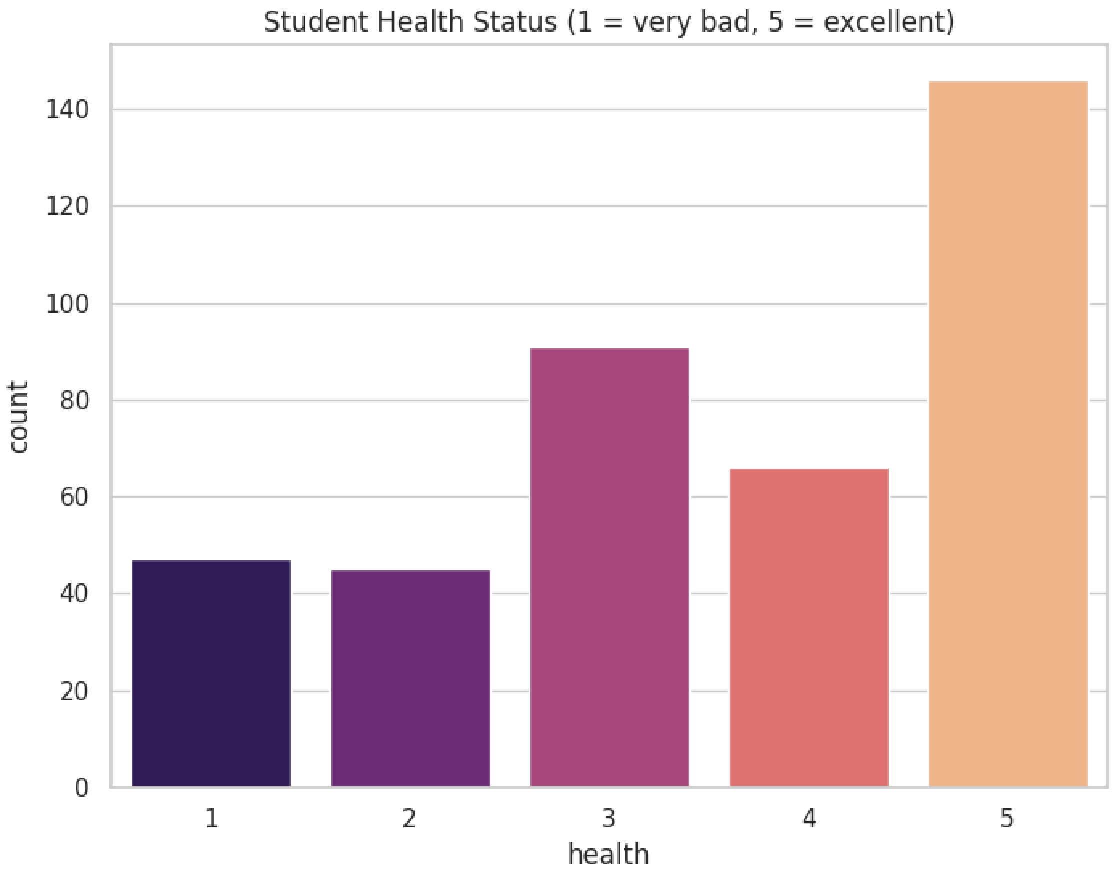
```
# plt.figure(figsize=(10, 6))
sns.countplot(x='Fjob', data=df, palette='inferno')
plt.title('Father Occupations')
plt.xticks(rotation=45)
plt.savefig('countplot_father_occupation.png')
plt.show()
```



```
# 5. Visualization: Family Relationship ('famrel' column)
plt.figure(figsize=(8, 6))
sns.countplot(x='famrel', data=df, palette='Spectral')
plt.title('Quality of Family Relationships (1 = very bad, 5 = excellent)')
plt.savefig('countplot_famrel.png')
plt.show()
```

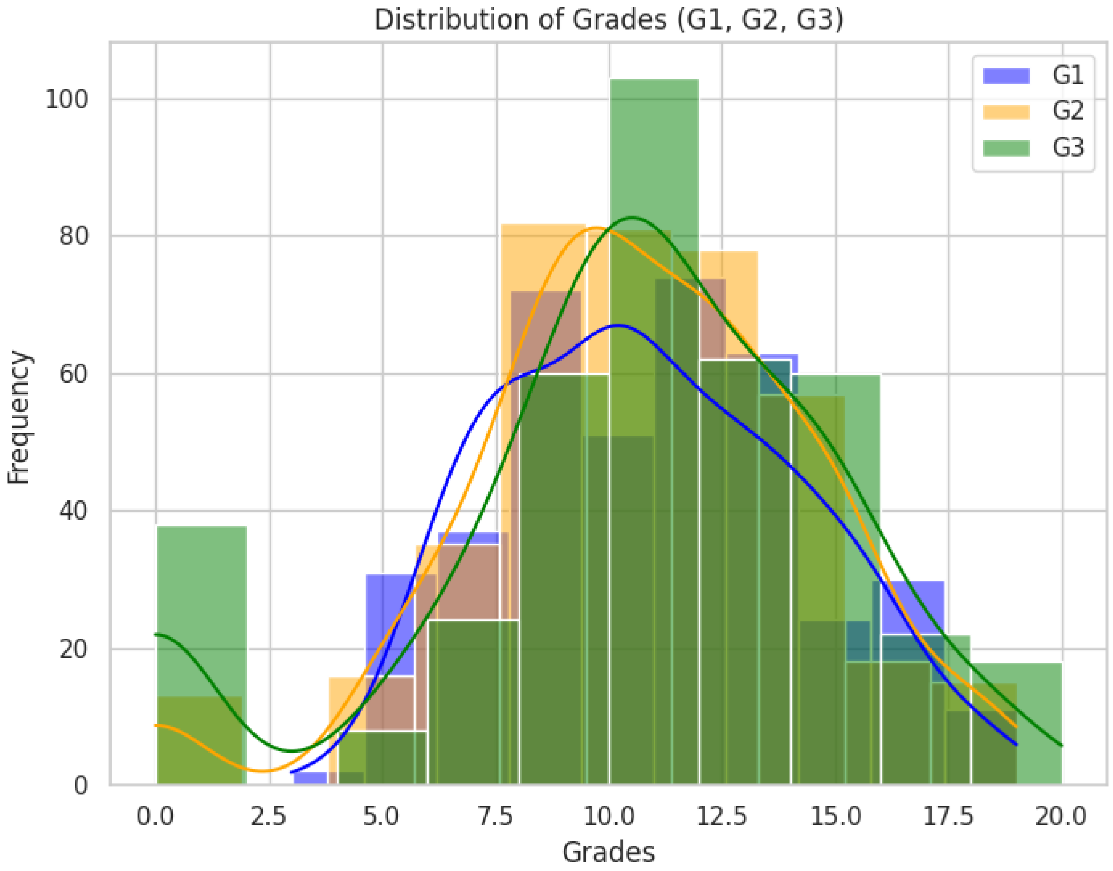


```
 # # 6. Visualization: Health Status of Students ('health' column)
 plt.figure(figsize=(8, 6))
sns.countplot(x='health', data=df, palette='magma')
plt.title('Student Health Status (1 = very bad, 5 = excellent)')
 plt.savefig('countplot_health.png')
plt.show()
```



7. Visualization: Final Grades ('G1', 'G2', 'G3')

```
plt.figure(figsize=(8, 6))
sns.histplot(df['G1'], color='blue', label='G1', kde=True, bins=10)
sns.histplot(df['G2'], color='orange', label='G2', kde=True, bins=10)
sns.histplot(df['G3'], color='green', label='G3', kde=True, bins=10)
plt.title('Distribution of Grades (G1, G2, G3)')
plt.xlabel('Grades')
plt.ylabel('Frequency')
plt.legend()
plt.savefig('histplot_grades.png')
plt.show()
```



FULL CODE FOR EXECUTE

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
file_path = '~/Downloads/student_data.csv' # Update this path based on your dataset location
df = pd.read_csv(file_path)

# Display the first few rows of the dataset to verify the data
print(df.head())
```

```
# Check for missing values in the dataset
print(df.isnull().sum())

# Display the column names for reference
print(df.columns)

# Set the style for seaborn
sns.set(style="whitegrid")

# 1. Visualization: Countplot for Gender ('sex' column)
plt.figure(figsize=(8, 6))
sns.countplot(x='sex', data=df, palette='coolwarm')
plt.title('Count of Male and Female Students')
plt.savefig('countplot_sex.png') # Save the figure as PNG
plt.show()

# # #2.Visualization: Family size distribution ('famsize' column)
plt.figure(figsize=(8, 6))
sns.countplot(x='famsize', data=df, palette='Set2')
plt.title('Family Size Distribution (GT3 vs LE3)')
plt.savefig('countplot_famsize.png')
plt.show()

#3. Visualization: Education of Parents ('Medu' and 'Fedu' columns)
plt.figure(figsize=(8, 6))
sns.histplot(df['Medu'], bins=5, color='blue', label='Mother Education', kde=True)
sns.histplot(df['Fedu'], bins=5, color='red', label='Father Education', kde=True)
plt.title('Distribution of Parental Education (Mother and Father)')
plt.xlabel('Education Level (0-4)')
plt.ylabel('Frequency')
plt.legend()
plt.savefig('histplot_parental_education.png')
plt.show()

# # 4. Visualization: Distribution of Occupations ('Mjob' and 'Fjob' columns)
plt.figure(figsize=(10, 6))
sns.countplot(x='Mjob', data=df, palette='viridis')
plt.title('Mother Occupations')
plt.xticks(rotation=45)
plt.savefig('countplot_mother_occupation.png')
plt.show()

# plt.figure(figsize=(10, 6))
sns.countplot(x='Fjob', data=df, palette='inferno')
plt.title('Father Occupations')
plt.xticks(rotation=45)
plt.savefig('countplot_father_occupation.png')
plt.show()

# # # 5. Visualization: Family Relationship ('famrel' column)
plt.figure(figsize=(8, 6))
sns.countplot(x='famrel', data=df, palette='Spectral')
```

```
plt.title('Quality of Family Relationships (1 = very bad, 5 = excellent)')
plt.savefig('countplot_famrel.png')
plt.show()

# # # 6. Visualization: Health Status of Students ('health' column)
plt.figure(figsize=(8, 6))
sns.countplot(x='health', data=df, palette='magma')
plt.title('Student Health Status (1 = very bad, 5 = excellent)')
plt.savefig('countplot_health.png')
plt.show()

# # 7. Visualization: Final Grades ('G1', 'G2', 'G3')
plt.figure(figsize=(8, 6))
sns.histplot(df['G1'], color='blue', label='G1', kde=True, bins=10)
sns.histplot(df['G2'], color='orange', label='G2', kde=True, bins=10)
sns.histplot(df['G3'], color='green', label='G3', kde=True, bins=10)
plt.title('Distribution of Grades (G1, G2, G3)')
plt.xlabel('Grades')
plt.ylabel('Frequency')
plt.legend()
plt.savefig('histplot_grades.png')
plt.show()

# 8. Visualization: Correlation Heatmap (relationships between variables)
plt.figure(figsize=(12, 8))
corr = df.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.savefig('heatmap_correlation.png')
plt.show()
```