In [1]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Student Performance EDA

📝 Introduction

- Dataset: Student performance data with ~395 records.
- Objective: Understand student demographics,family,study habits etc

In [3]:
```
df=pd.read_csv('student_mat.csv')
```

In [4]:
```
df.head()
```

Out[4]:

	school;sex;age;address;famsize;Pstatus;Medu;Fedu;Mjob;Fjob;reason;guardian;traveltime;s
0	
1	
2	
3	
4	

In [5]:
```
df=pd.read_csv('student_mat.csv',sep=';',quotechar='"')
```

In [6]:
```
df.shape
```

Out[6]:
```
(395, 33)
```

In [7]:
```
df.head()
```

Out[7]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	fan
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	
3	GP	F	15	U	GT3	T	4	2	health	services	...	
4	GP	F	16	U	GT3	T	3	3	other	other	...	

5 rows × 33 columns

📊 Data Overview

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   school      395 non-null    object
 1   sex         395 non-null    object
 2   age         395 non-null    int64 
 3   address     395 non-null    object
 4   famsize     395 non-null    object
 5   Pstatus     395 non-null    object
 6   Medu        395 non-null    int64 
 7   Fedu        395 non-null    int64 
 8   Mjob        395 non-null    object
 9   Fjob        395 non-null    object
 10  reason      395 non-null    object
 11  guardian    395 non-null    object
 12  traveltime  395 non-null    int64 
 13  studytime   395 non-null    int64 
 14  failures    395 non-null    int64 
 15  schoolsup   395 non-null    object
 16  famsup      395 non-null    object
 17  paid        395 non-null    object
 18  activities  395 non-null    object
 19  nursery     395 non-null    object
 20  higher      395 non-null    object
 21  internet    395 non-null    object
 22  romantic    395 non-null    object
 23  famrel      395 non-null    int64 
 24  freetime    395 non-null    int64 
 25  goout       395 non-null    int64 
 26  Dalc        395 non-null    int64 
 27  Walc        395 non-null    int64 
 28  health      395 non-null    int64 
 29  absences    395 non-null    int64 
 30  G1          395 non-null    int64 
 31  G2          395 non-null    int64 
 32  G3          395 non-null    int64 
dtypes: int64(16), object(17)
memory usage: 102.0+ KB
```

In [9]:
```
df.describe()
```

Out[9]:

	age	Medu	Fedu	traveltime	studytime	failures	fam
count	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.0000
mean	16.696203	2.749367	2.521519	1.448101	2.035443	0.334177	3.9443
std	1.276043	1.094735	1.088201	0.697505	0.839240	0.743651	0.8966
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.0000
25%	16.000000	2.000000	2.000000	1.000000	1.000000	0.000000	4.0000
50%	17.000000	3.000000	2.000000	1.000000	2.000000	0.000000	4.0000
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.000000	5.0000
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.000000	5.0000

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
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
```

In [11]:
```
cat_col=df.select_dtypes(include=['object']).columns.tolist()
num_col=df.select_dtypes(exclude=['object']).columns.tolist()
```

In [12]:
```
print(cat_col)
```

```
['school', 'sex', 'address', 'famsize', 'Pstatus', 'Mjob', 'Fjob', 'reason', 'gua
rdian', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'intern
et', 'romantic']
```

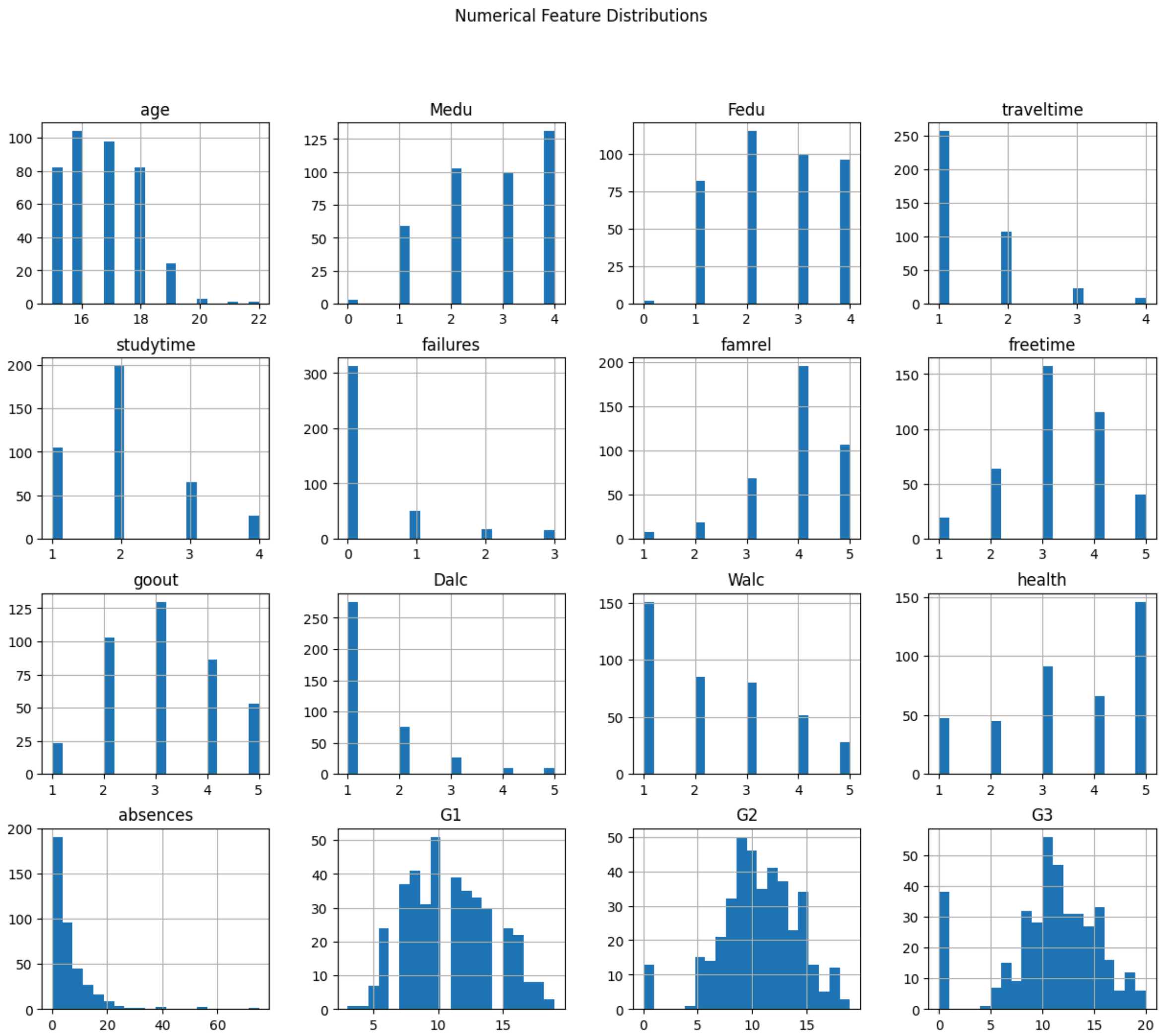
In [13]:
```
print(num_col)
```

```
['age', 'Medu', 'Fedu', 'traveltime', 'studytime', 'failures', 'famrel', 'freetim
e', 'goout', 'Dalc', 'Walc', 'health', 'absences', 'G1', 'G2', 'G3']
```

In [14]:
```
for col in cat_col[:5]:
    print(f"\nValuecounts for {col}:")
    print(df[col].value_counts())
```

```
Valuecounts for school:
school
GP    349
MS     46
Name: count, dtype: int64

Valuecounts for sex:
sex
F    208
M    187
Name: count, dtype: int64

Valuecounts for address:
address
U    307
R     88
Name: count, dtype: int64

Valuecounts for famsize:
famsize
GT3    281
LE3    114
Name: count, dtype: int64

Valuecounts for Pstatus:
Pstatus
T    354
A     41
Name: count, dtype: int64
```

In [15]:
```
df[num_col].hist(figsize=(15, 12), bins=20)
plt.suptitle("Numerical Feature Distributions")
plt.show()
```



- Total records: 395
- Columns: 33
- Data types: Mix of int and object
- Missing values: No missing values

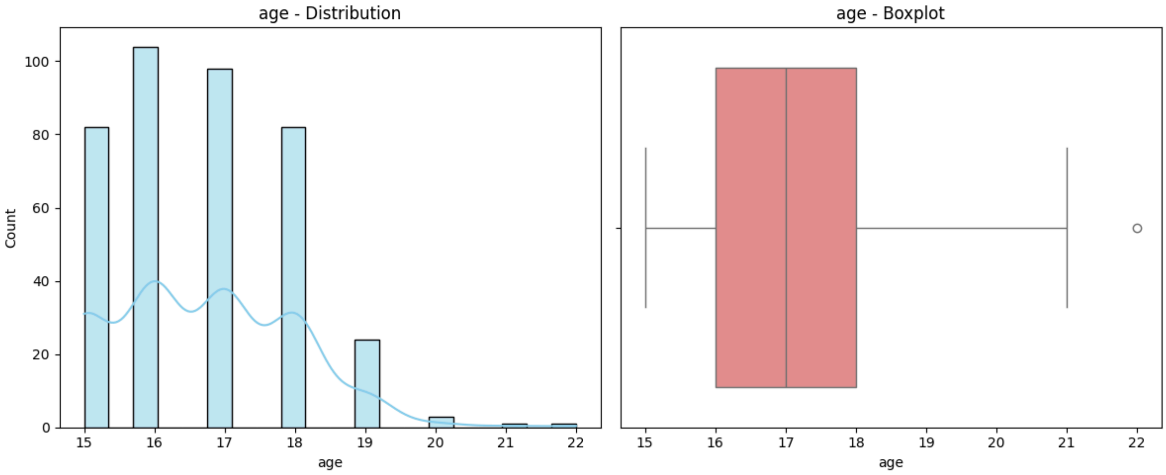
📊 Univariate Analysis

In [16]:

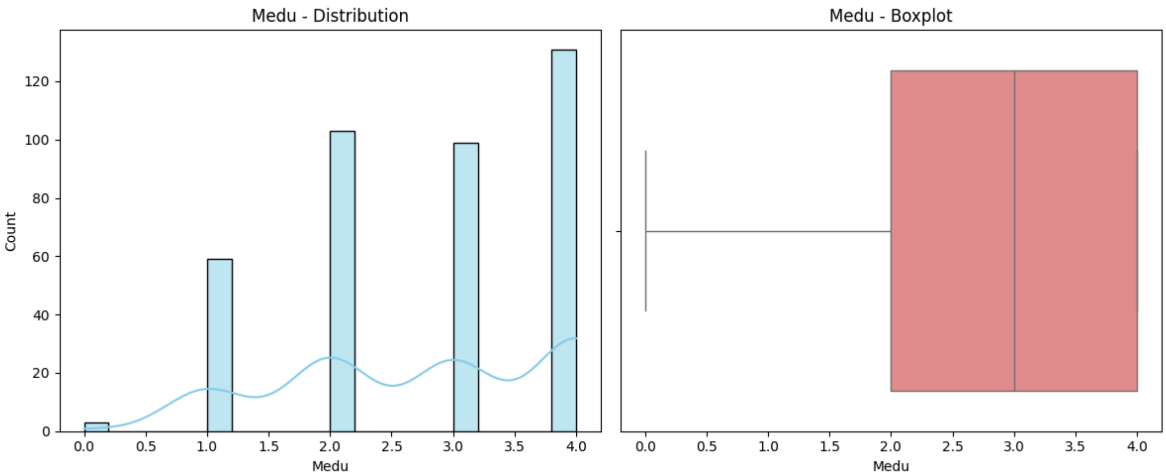
```
#Numerical columns
print("📌 Numerical Features Analysis\n")

for col in num_col:
    plt.figure(figsize=(12,5))

    #Histogram
    plt.subplot(1,2,1)
    sns.histplot(df[col],kde=True,bins=20,color='skyblue')
    plt.title(f"{col} - Distribution")

    #Boxplot
    plt.subplot(1,2,2)
    sns.boxplot(x=df[col],color='lightcoral')
    plt.title(f"{col} - Boxplot")

    plt.tight_layout()
```

```python
    plt.show()

    print(f"{col} Summary Stats:\n", df[col].describe(), "\n")
```

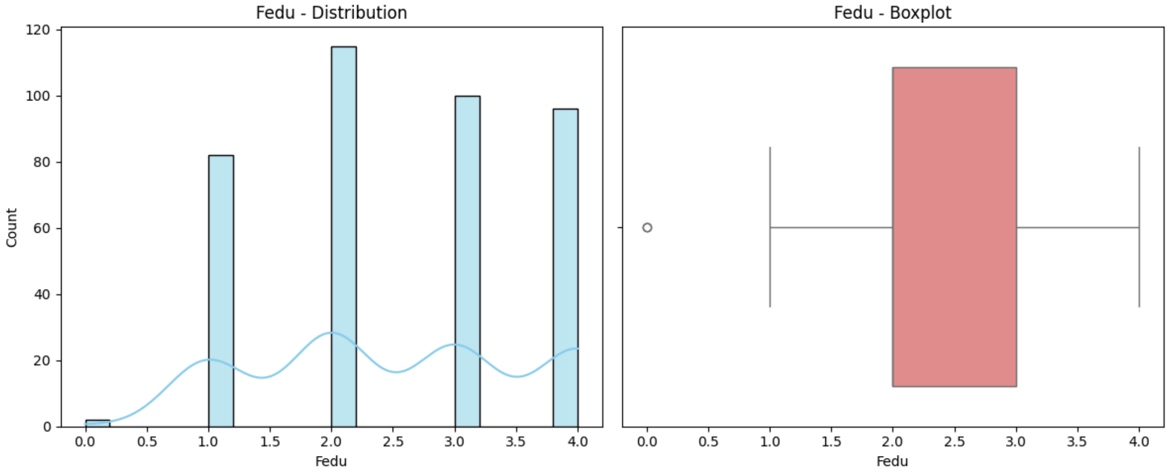
📌 Numerical Features Analysis



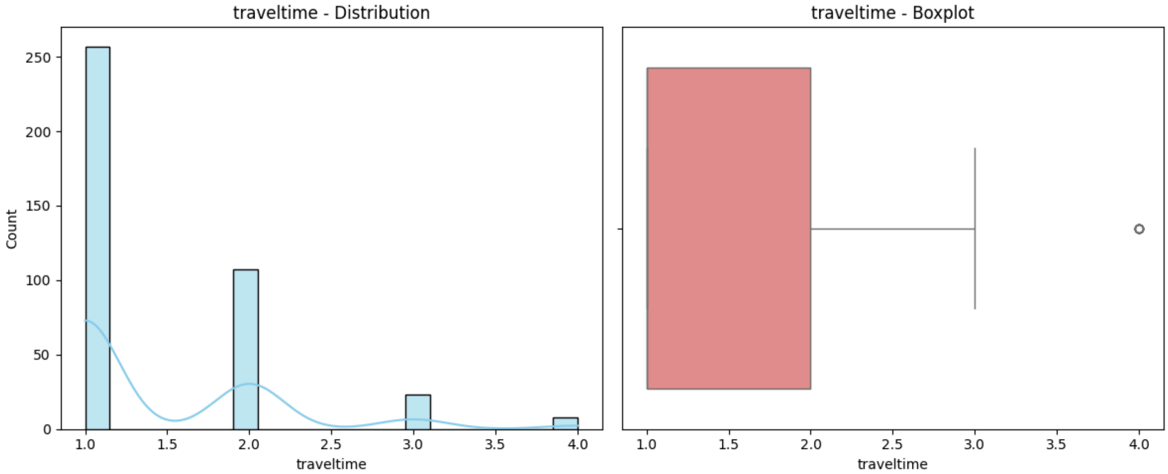

```
age Summary Stats:
 count    395.000000
mean      16.696203
std        1.276043
min       15.000000
25%       16.000000
50%       17.000000
75%       18.000000
max       22.000000
Name: age, dtype: float64
```



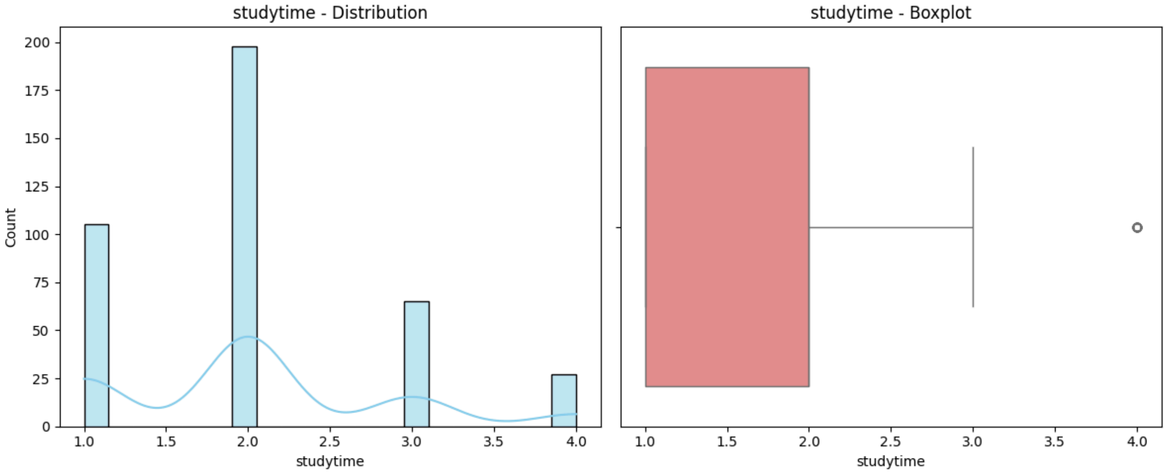

```
Medu Summary Stats:
 count    395.000000
mean       2.749367
std        1.094735
min        0.000000
25%        2.000000
50%        3.000000
75%        4.000000
max        4.000000
Name: Medu, dtype: float64
```



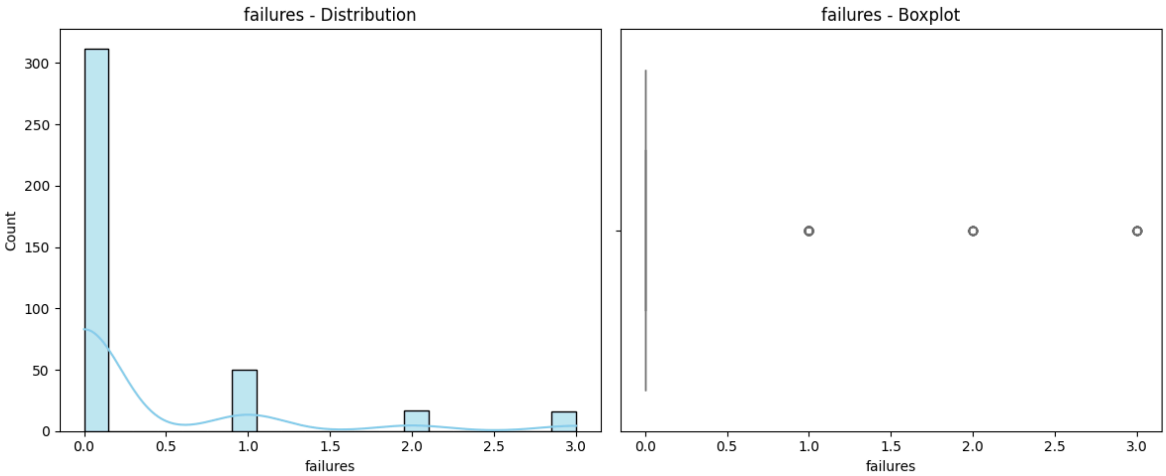

```
Fedu Summary Stats:
 count    395.000000
mean       2.521519
std        1.088201
min        0.000000
25%        2.000000
50%        2.000000
75%        3.000000
max        4.000000
Name: Fedu, dtype: float64
```



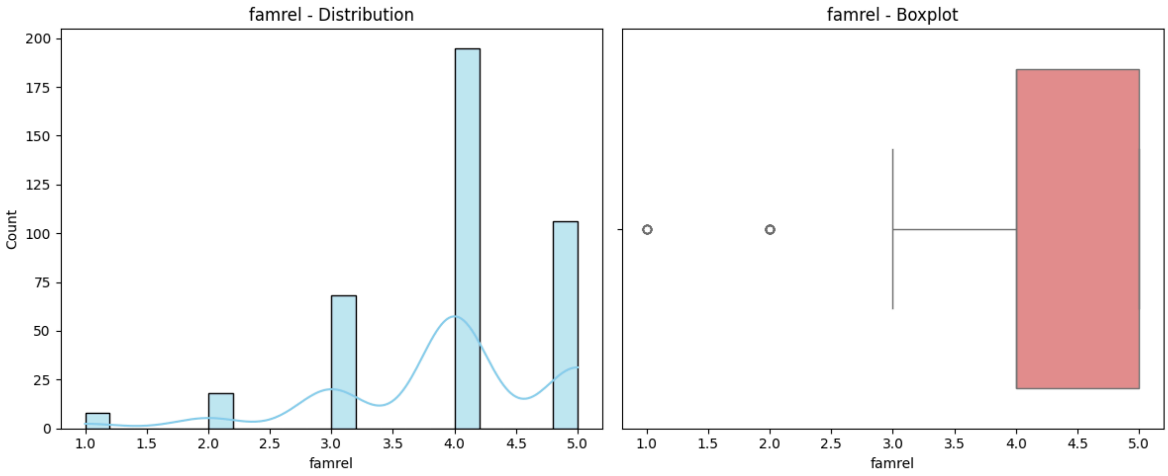

```
traveltime Summary Stats:
 count    395.000000
mean       1.448101
std        0.697505
min        1.000000
25%        1.000000
50%        1.000000
75%        2.000000
max        4.000000
Name: traveltime, dtype: float64
```



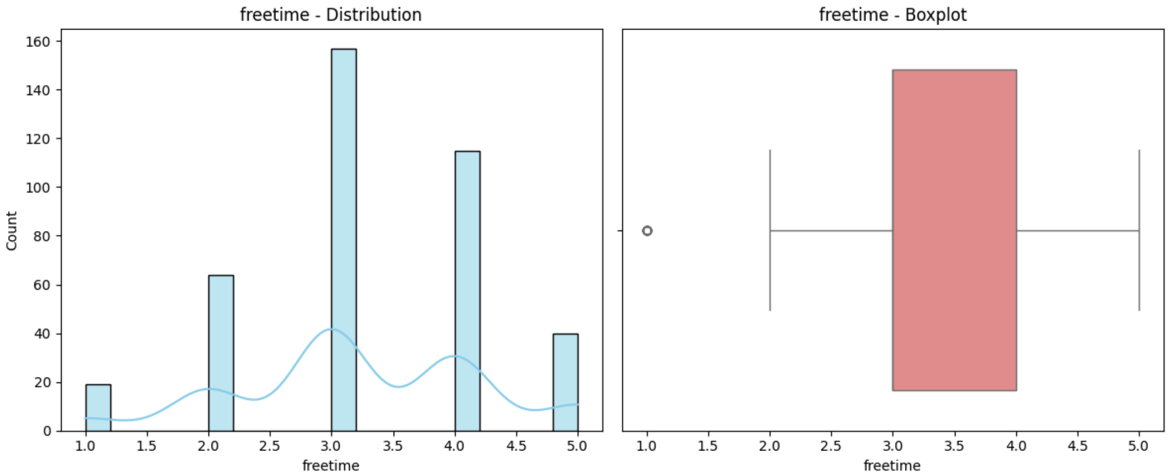

```
studytime Summary Stats:
 count    395.000000
mean       2.035443
std        0.839240
min        1.000000
25%        1.000000
50%        2.000000
75%        2.000000
max        4.000000
Name: studytime, dtype: float64
```



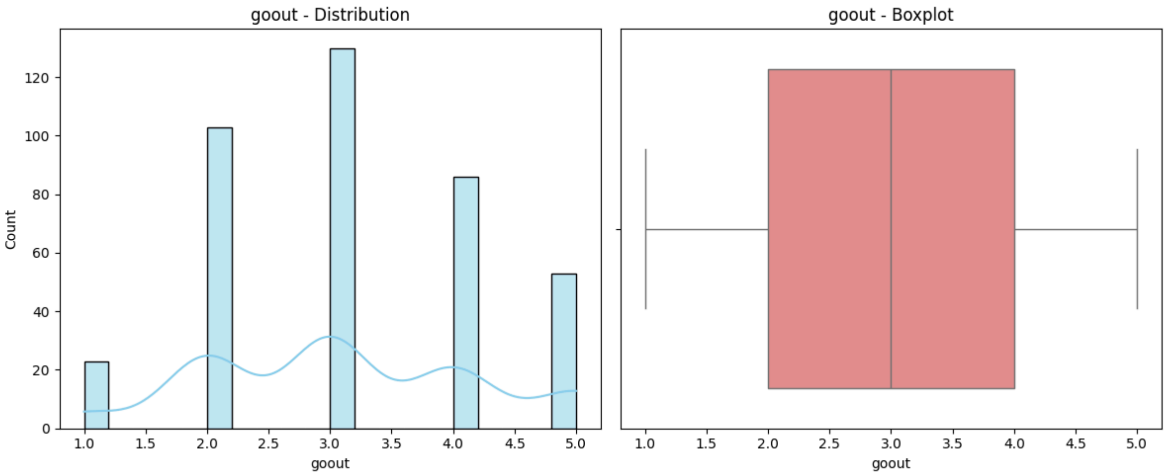

```
failures Summary Stats:
 count    395.000000
mean       0.334177
std        0.743651
min        0.000000
25%        0.000000
50%        0.000000
75%        0.000000
max        3.000000
Name: failures, dtype: float64
```



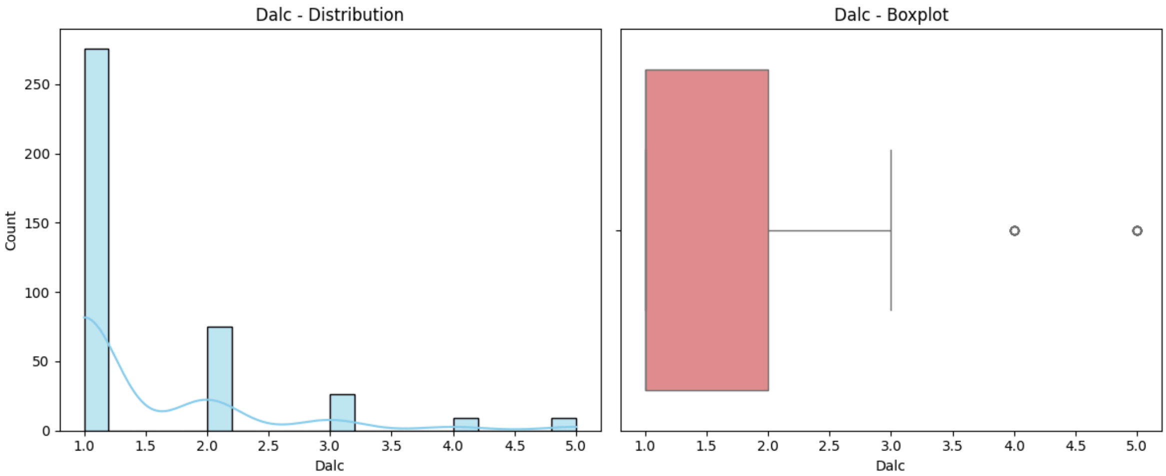

```
famrel Summary Stats:
 count    395.000000
mean       3.944304
std        0.896659
min        1.000000
25%        4.000000
50%        4.000000
75%        5.000000
max        5.000000
Name: famrel, dtype: float64
```



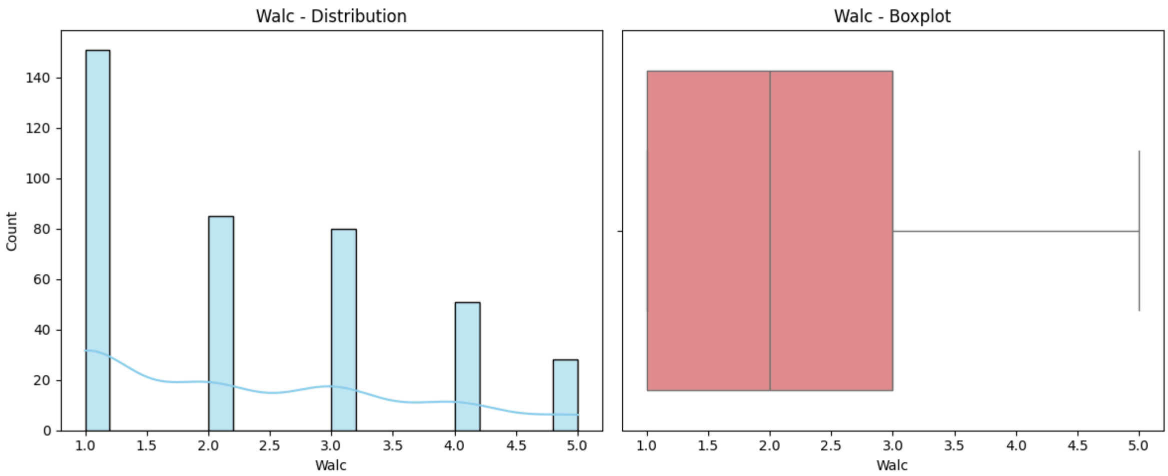

```
freetime Summary Stats:
 count    395.000000
mean       3.235443
std        0.998862
min        1.000000
25%        3.000000
50%        3.000000
75%        4.000000
max        5.000000
Name: freetime, dtype: float64
```



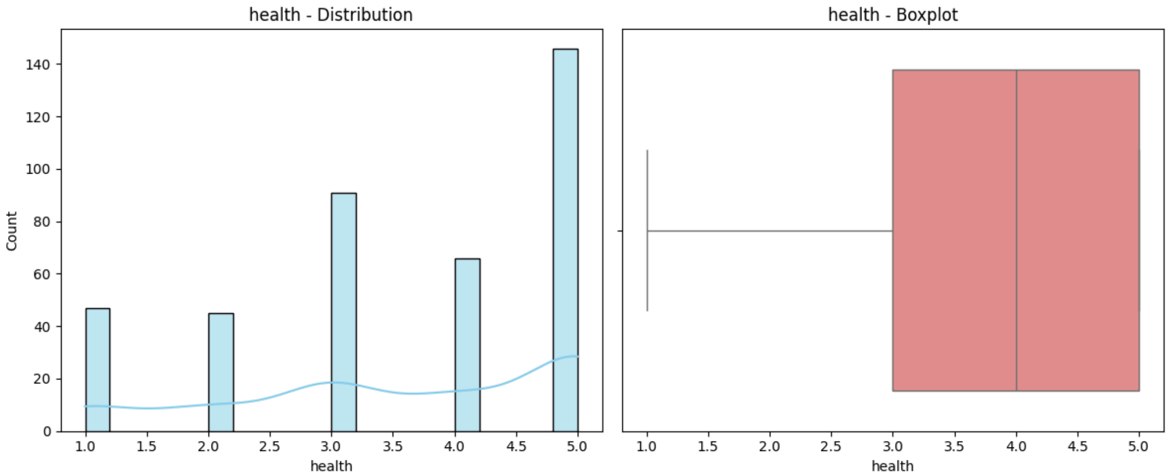

```
goout Summary Stats:
 count    395.000000
mean       3.108861
std        1.113278
min        1.000000
25%        2.000000
50%        3.000000
75%        4.000000
max        5.000000
Name: goout, dtype: float64
```



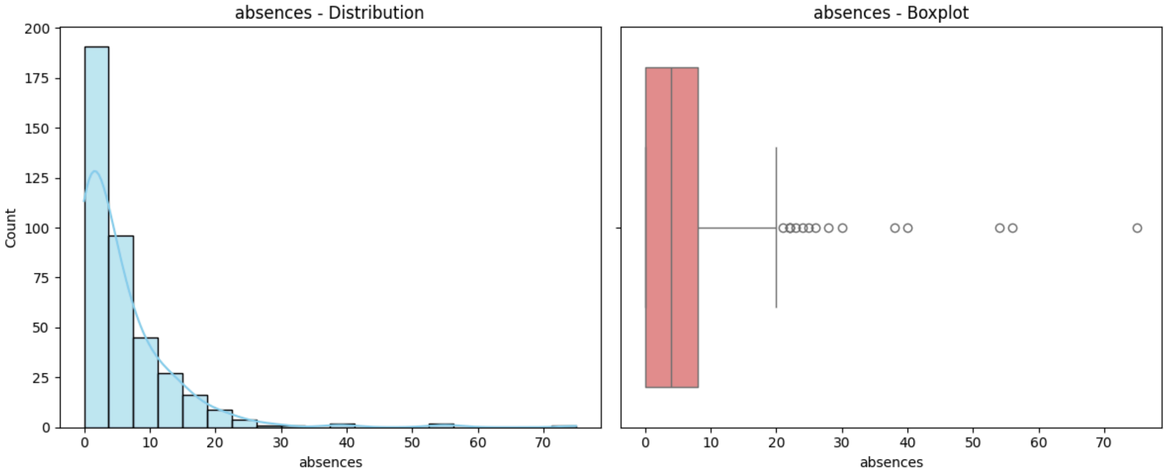

```
Dalc Summary Stats:
 count    395.000000
mean       1.481013
std        0.890741
min        1.000000
25%        1.000000
50%        1.000000
75%        2.000000
max        5.000000
Name: Dalc, dtype: float64
```



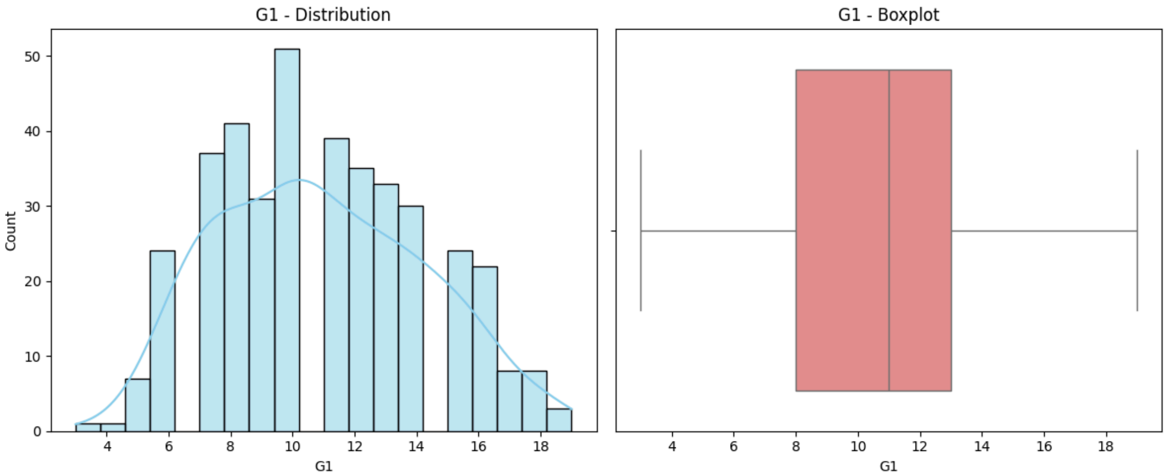

```
Walc Summary Stats:
 count    395.000000
mean       2.291139
std        1.287897
min        1.000000
25%        1.000000
50%        2.000000
75%        3.000000
max        5.000000
Name: Walc, dtype: float64
```



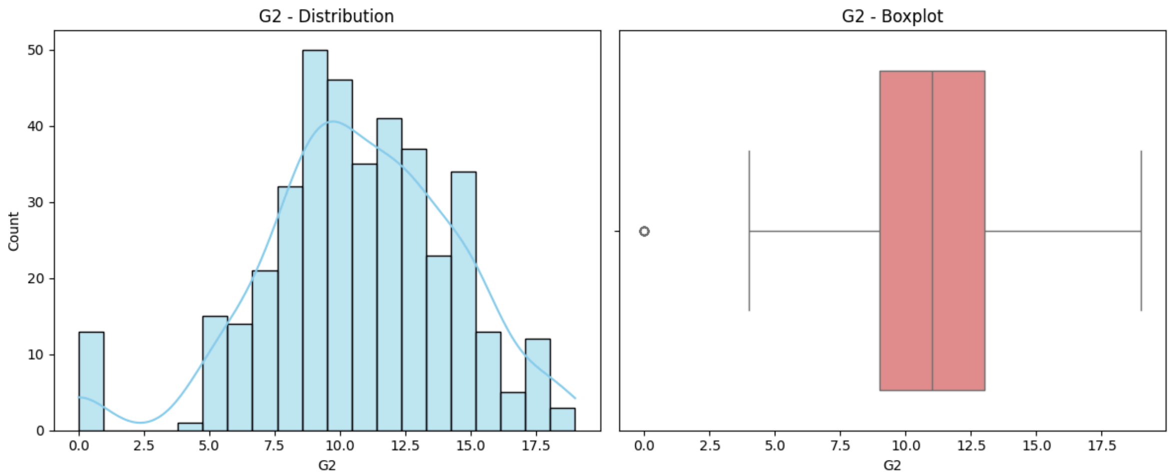

```
health Summary Stats:
 count    395.000000
mean       3.554430
std        1.390303
min        1.000000
25%        3.000000
50%        4.000000
75%        5.000000
max        5.000000
Name: health, dtype: float64
```



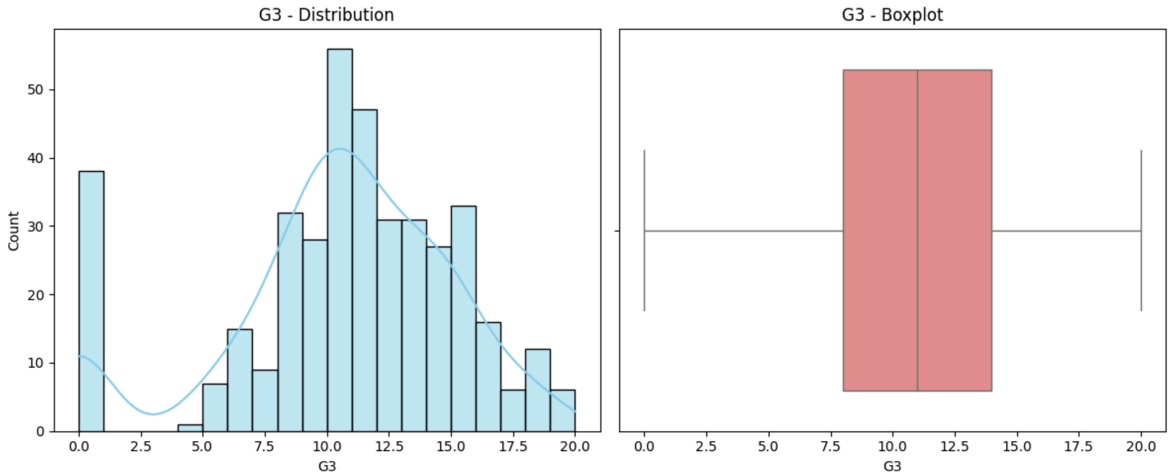

```
absences Summary Stats:
 count    395.000000
mean       5.708861
std        8.003096
min        0.000000
25%        0.000000
50%        4.000000
75%        8.000000
max       75.000000
Name: absences, dtype: float64
```




```
G1 Summary Stats:
 count    395.000000
mean      10.908861
std        3.319195
min        3.000000
25%        8.000000
50%       11.000000
75%       13.000000
max       19.000000
Name: G1, dtype: float64
```




```
G2 Summary Stats:
 count    395.000000
mean      10.713924
std        3.761505
min        0.000000
25%        9.000000
50%       11.000000
75%       13.000000
max       19.000000
Name: G2, dtype: float64
```



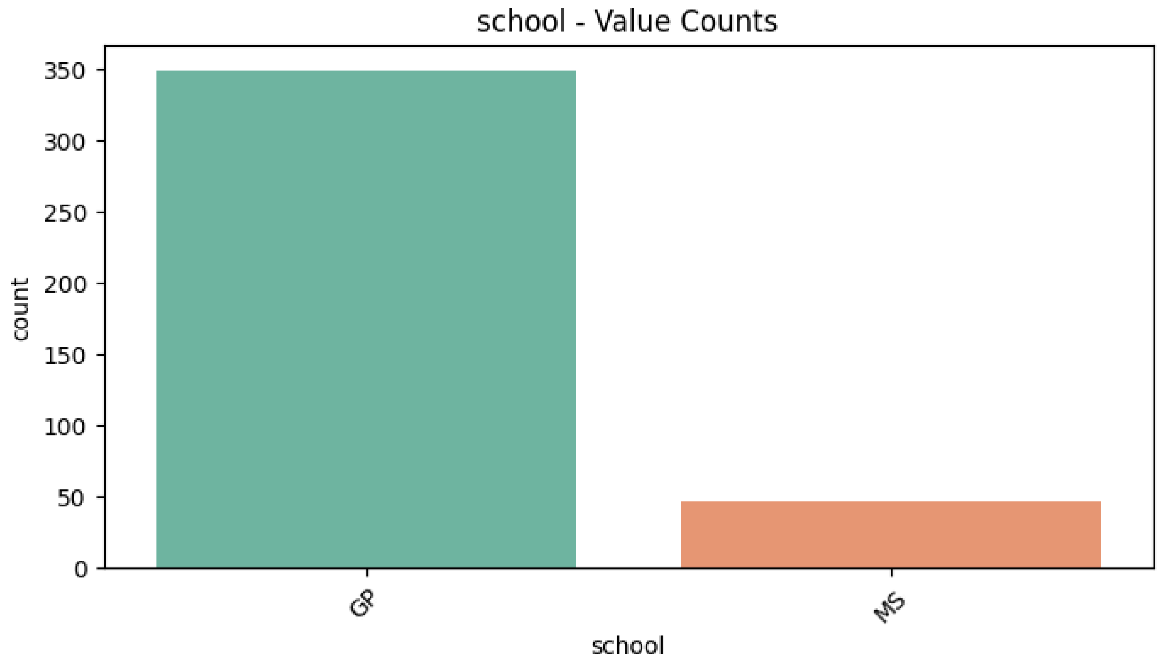

```
G3 Summary Stats:
 count    395.000000
mean      10.415190
std        4.581443
min        0.000000
25%        8.000000
50%       11.000000
75%       14.000000
max       20.000000
Name: G3, dtype: float64
```

In [17]:

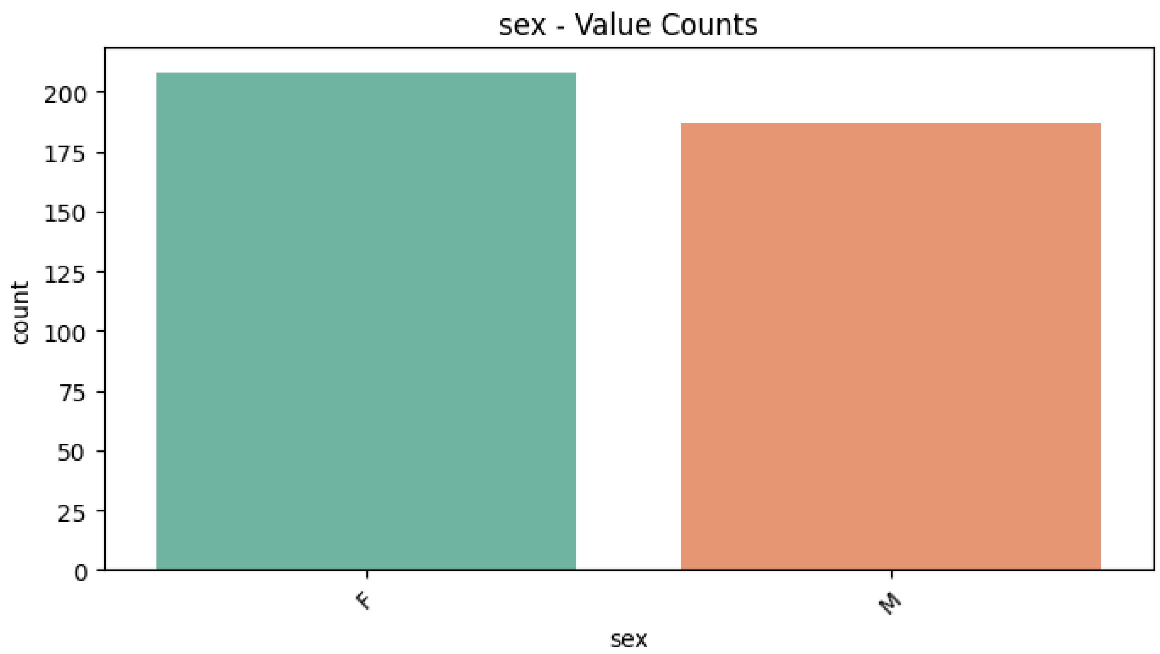
```
# 2. Categorical Features
print("📌 Categorical Features Analysis\n")

for col in cat_col:
    plt.figure(figsize=(8,4))
    sns.countplot(x=df[col], order=df[col].value_counts().index, palette="Set2")
    plt.title(f"{col} - Value Counts")
    plt.xticks(rotation=45)
```

```
    plt.show()

    print(f"Value Counts for {col}:\n", df[col].value_counts(), "\n")
```

📌 Categorical Features Analysis

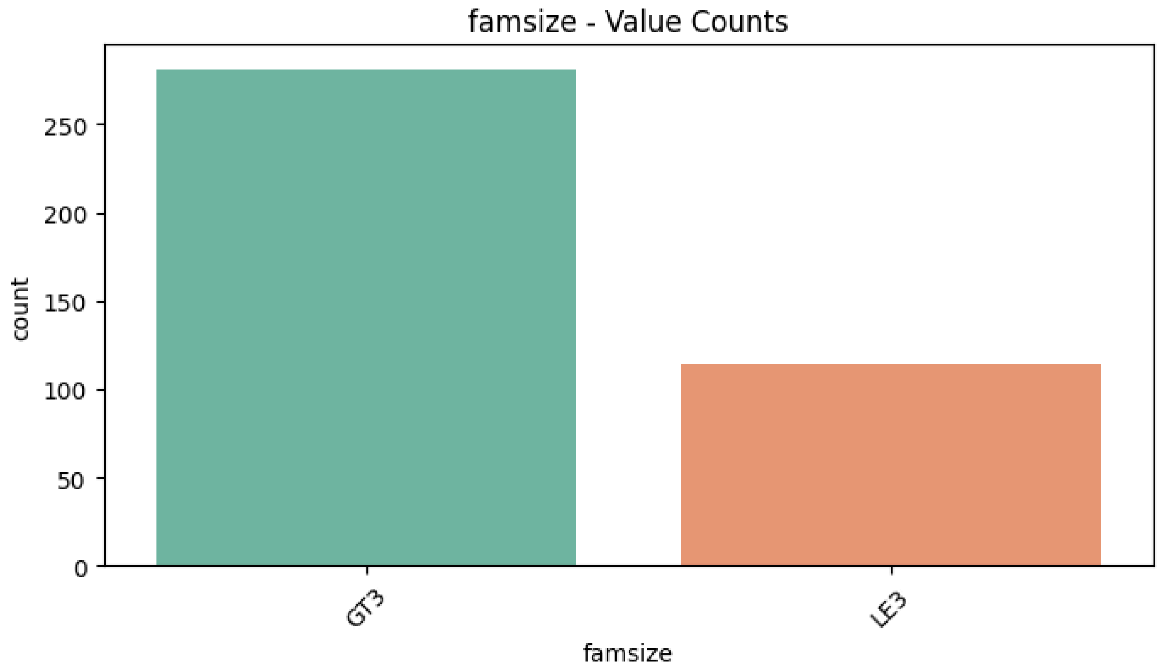



```
Value Counts for school:
 school
GP    349
MS     46
Name: count, dtype: int64
```

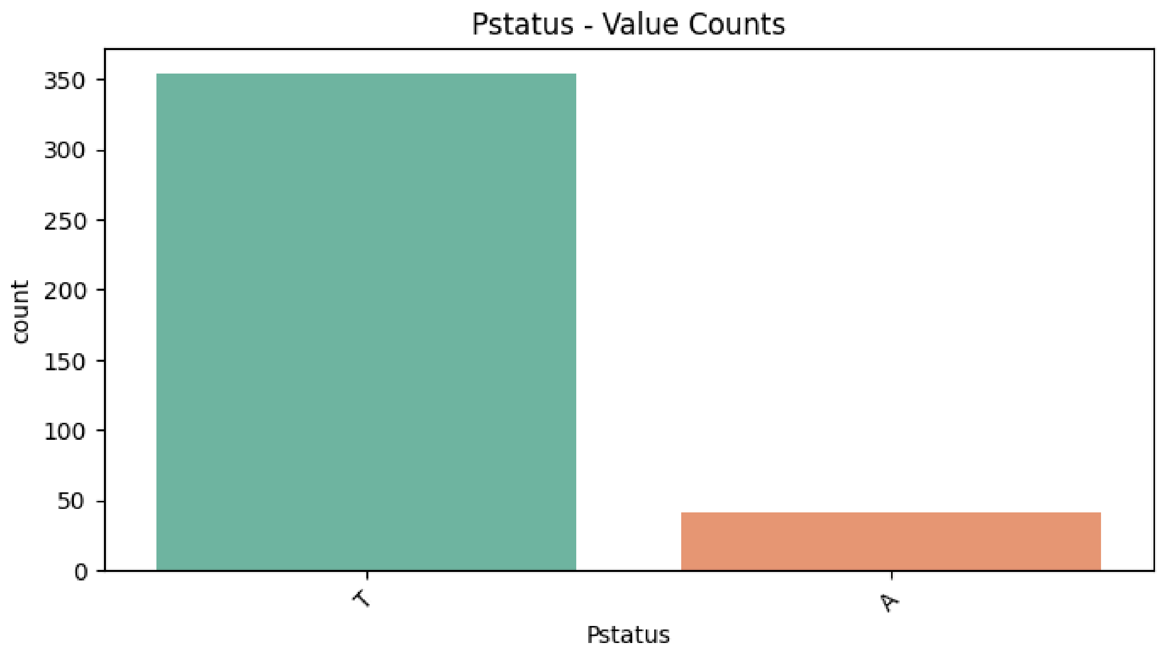



```
Value Counts for sex:
 sex
F    208
M    187
Name: count, dtype: int64
```

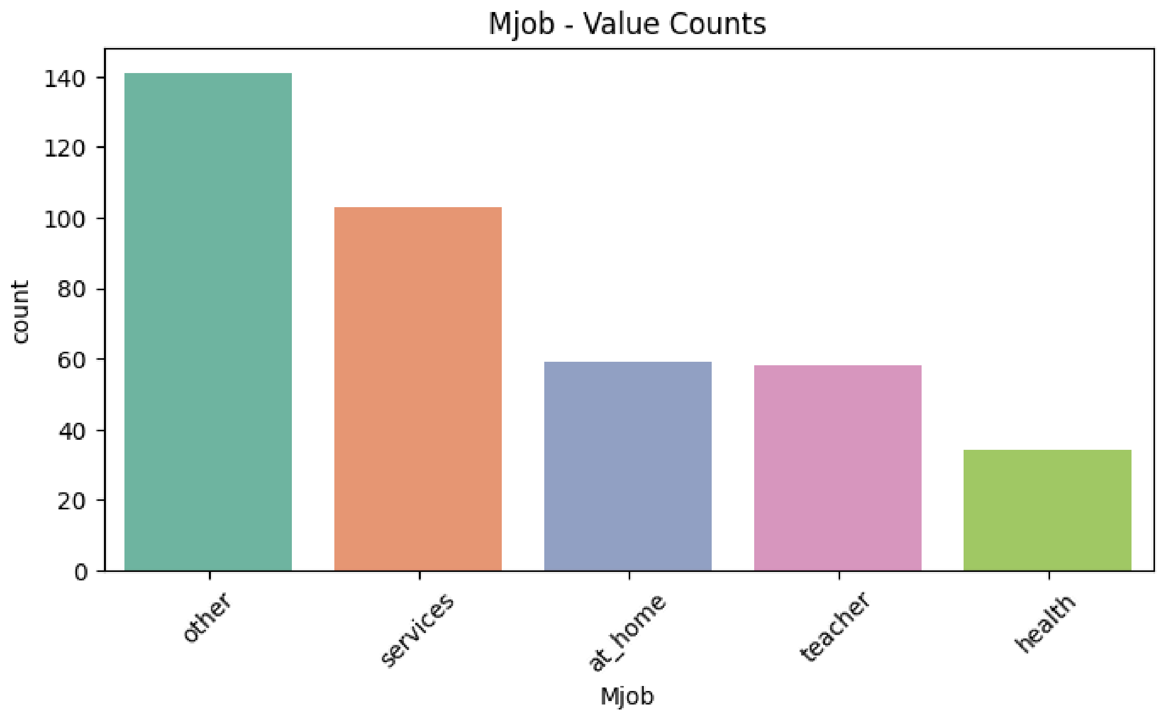

```
Value Counts for address:
 address
U    307
R     88
Name: count, dtype: int64
```



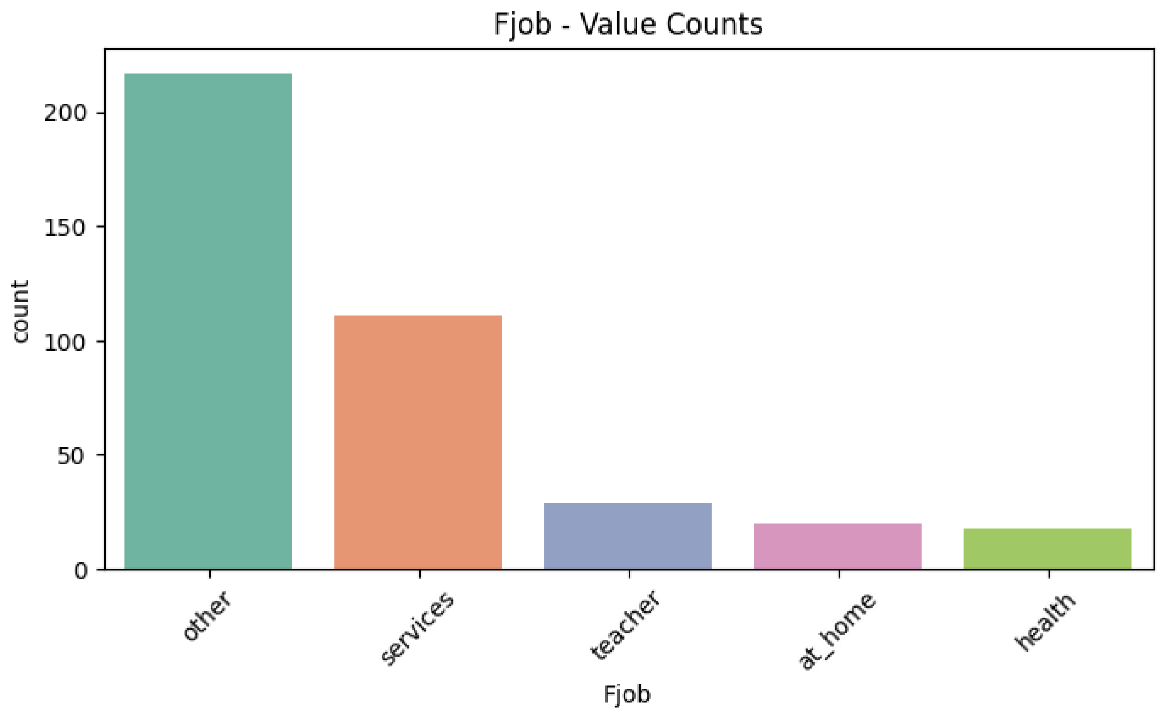

```
Value Counts for famsize:
 famsize
GT3    281
LE3    114
Name: count, dtype: int64
```



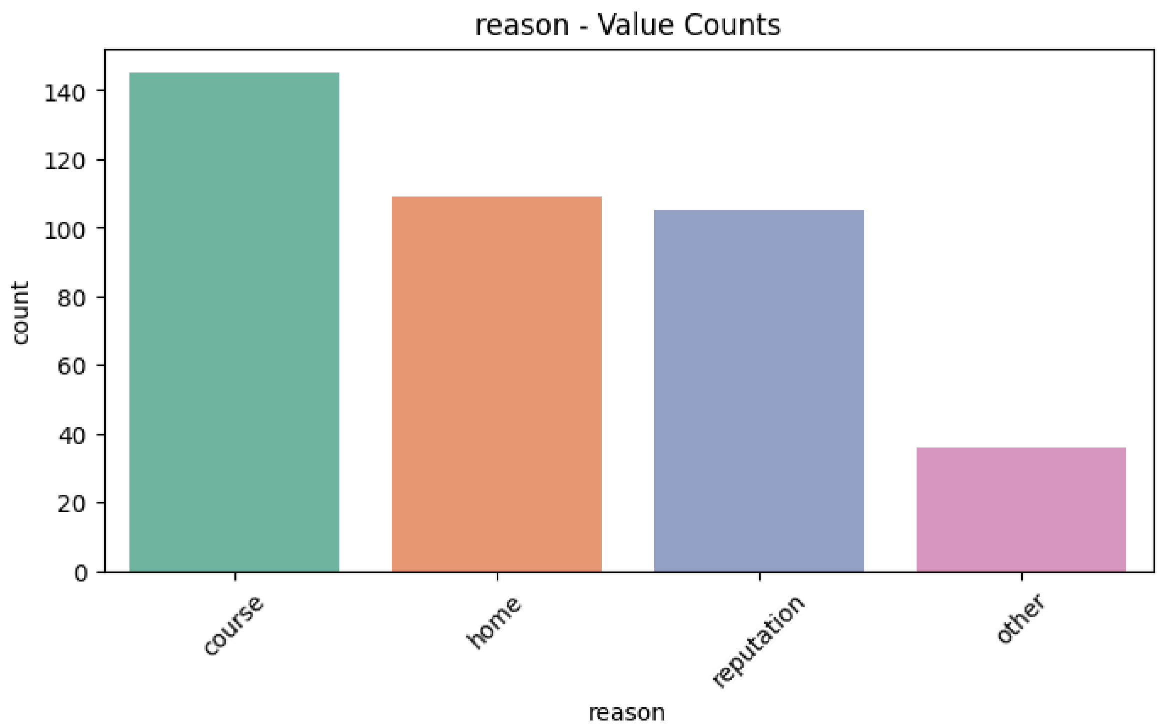

```
Value Counts for Pstatus:
 Pstatus
T    354
A     41
Name: count, dtype: int64
```



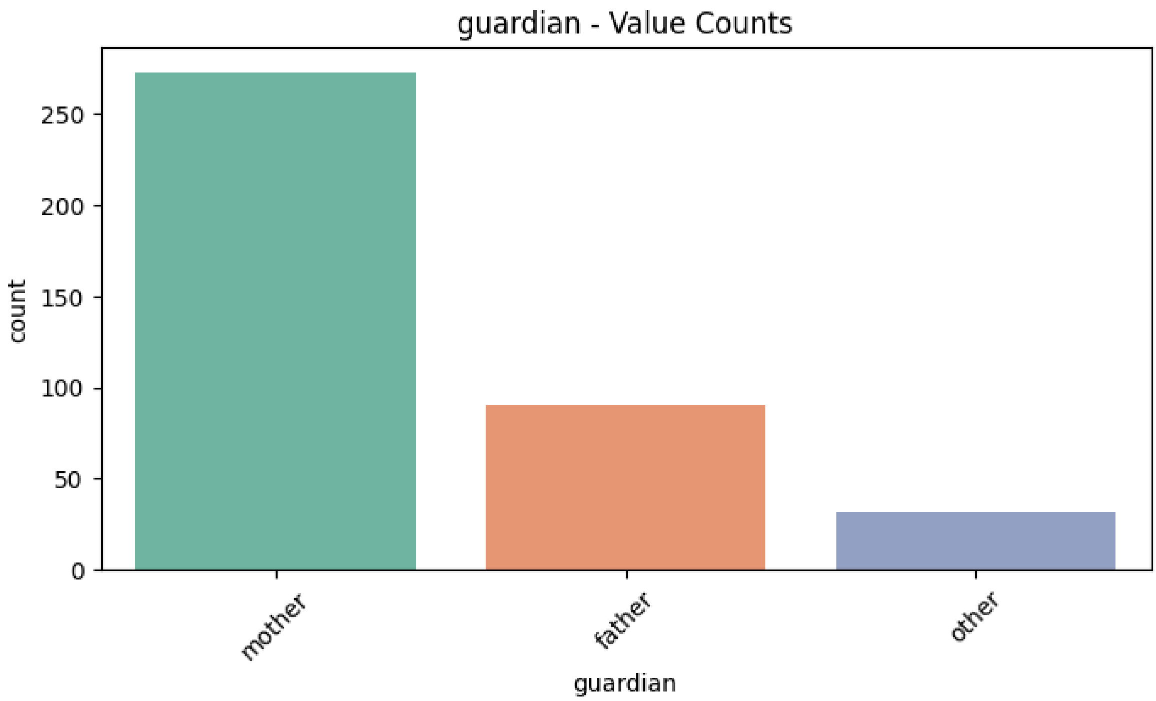

```
Value Counts for Mjob:
 Mjob
other       141
services    103
at_home      59
teacher      58
health       34
Name: count, dtype: int64
```



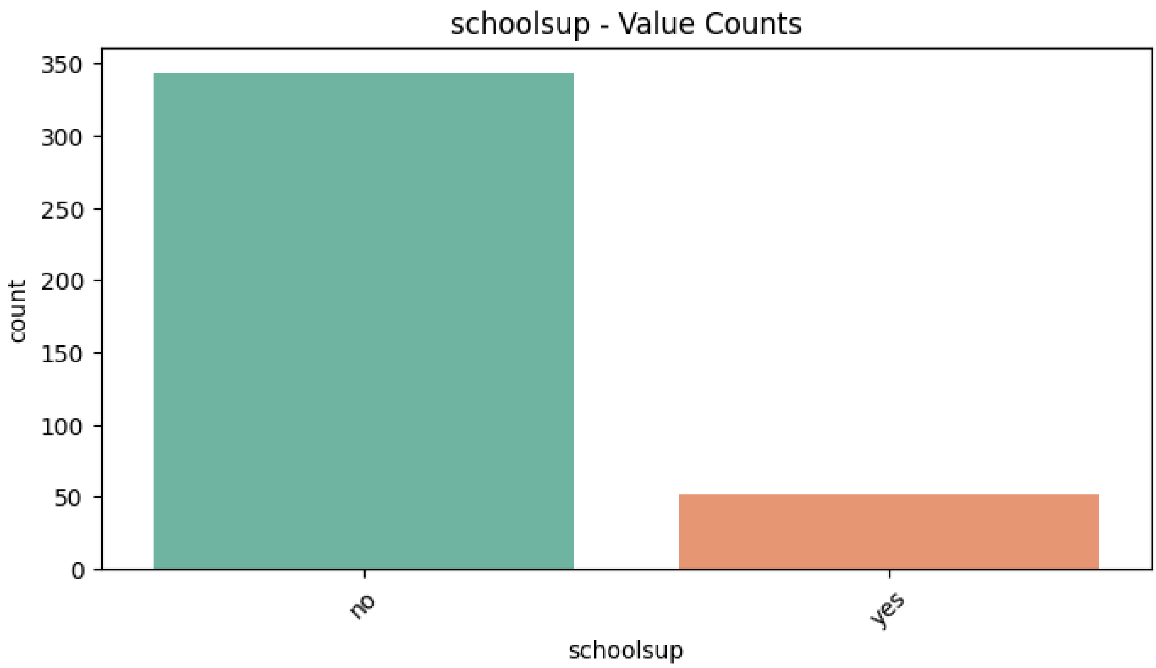

```
Value Counts for Fjob:
 Fjob
other       217
services    111
teacher      29
at_home      20
health       18
Name: count, dtype: int64
```



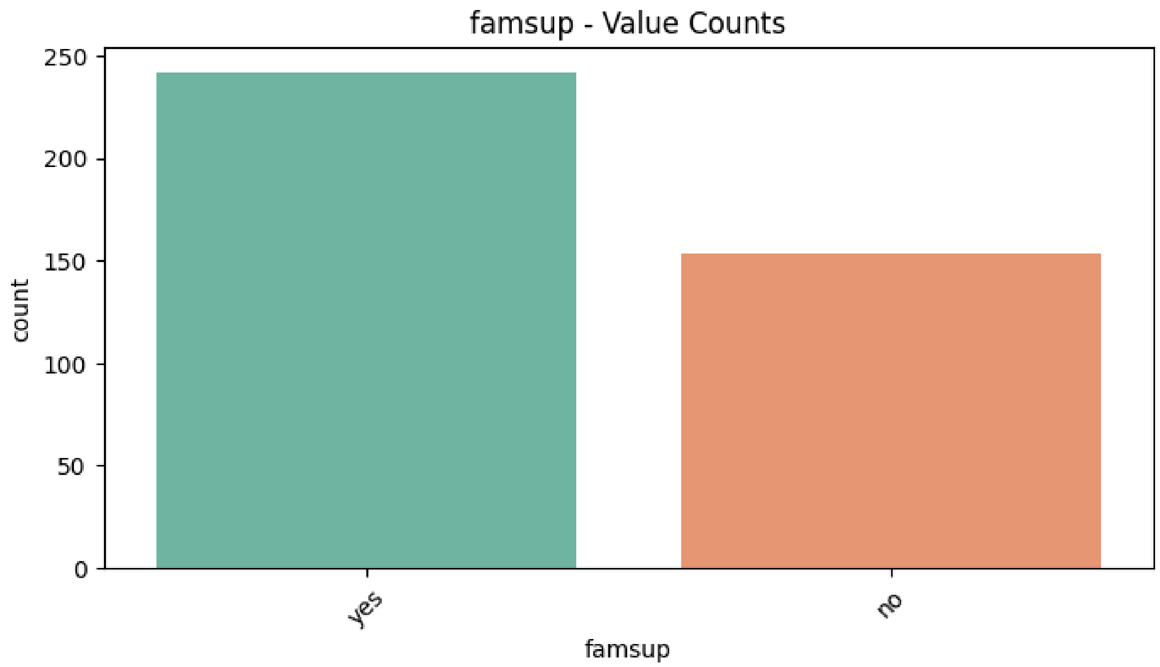
```
Value Counts for reason:
 reason
course        145
home          109
reputation    105
other          36
Name: count, dtype: int64
```



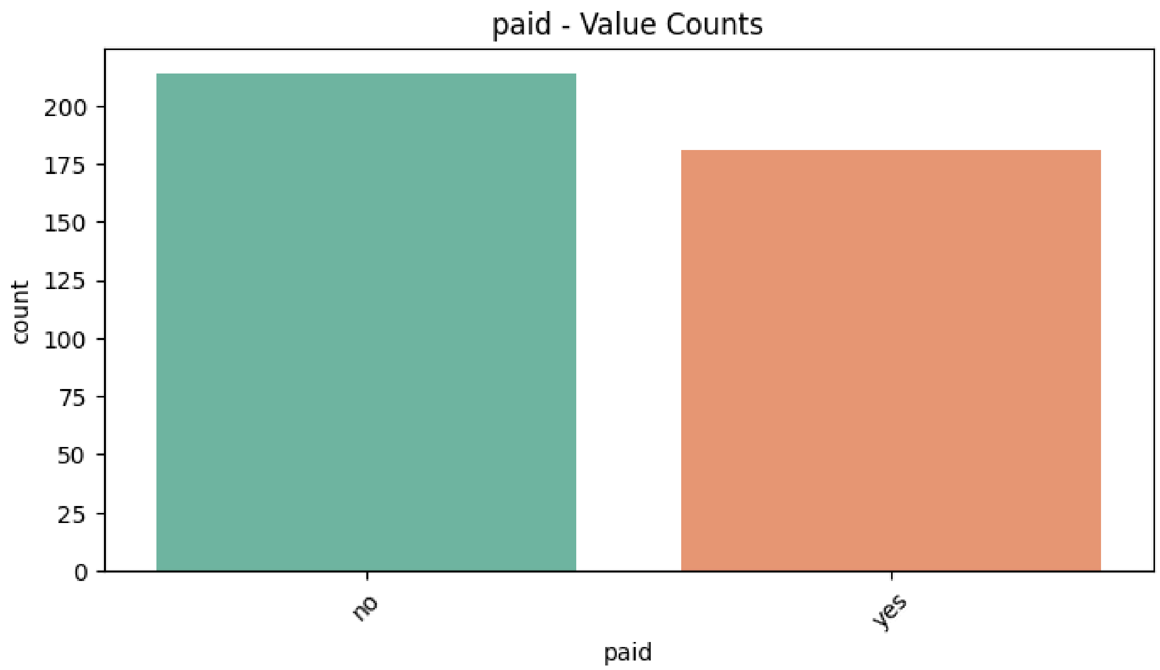

```
Value Counts for guardian:
 guardian
mother    273
father     90
other      32
Name: count, dtype: int64
```



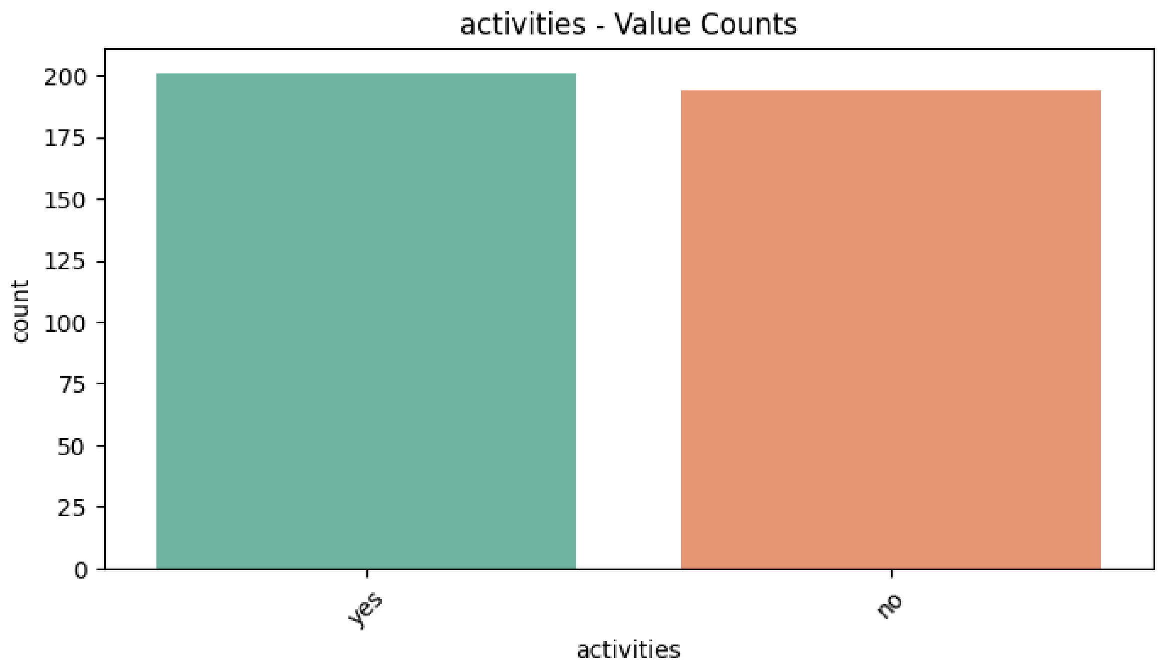
```
Value Counts for schoolsup:
 schoolsup
no     344
yes     51
Name: count, dtype: int64
```



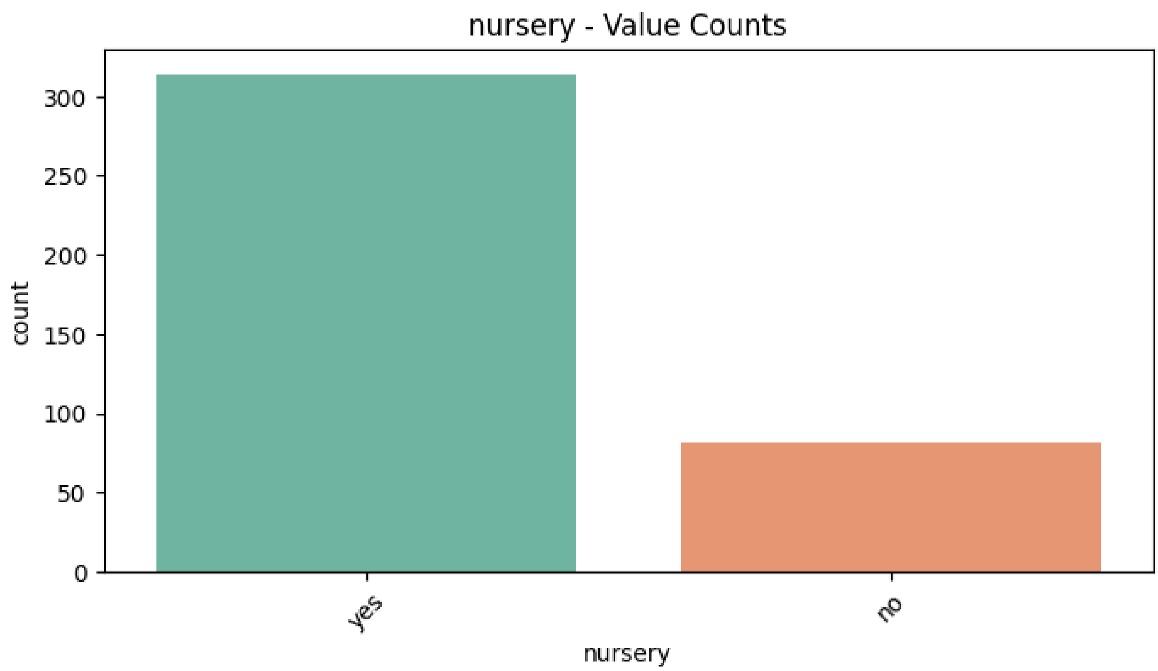

```
Value Counts for famsup:
 famsup
yes    242
no     153
Name: count, dtype: int64
```



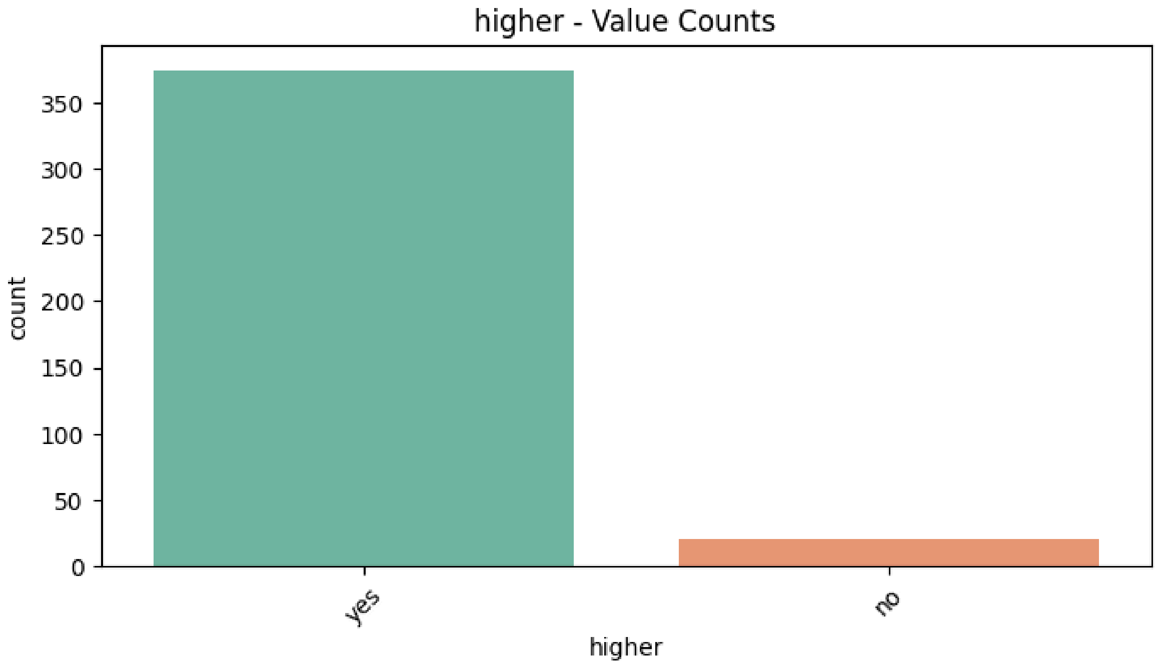

```
Value Counts for paid:
 paid
no     214
yes    181
Name: count, dtype: int64
```



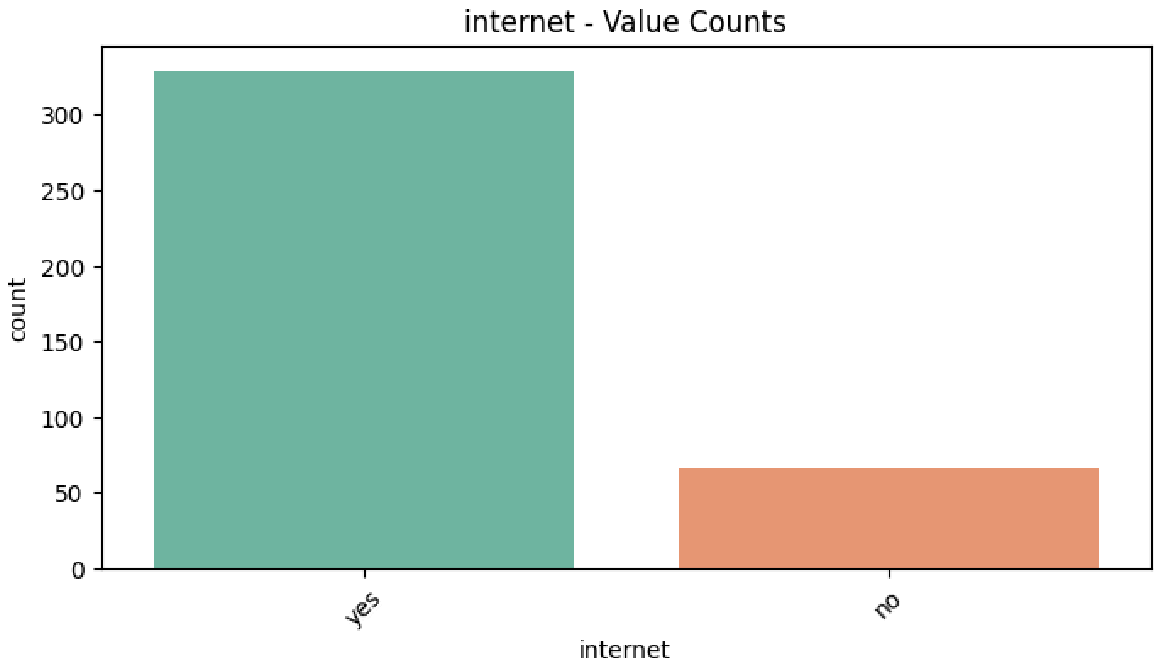

```
Value Counts for activities:
 activities
yes    201
no     194
Name: count, dtype: int64
```



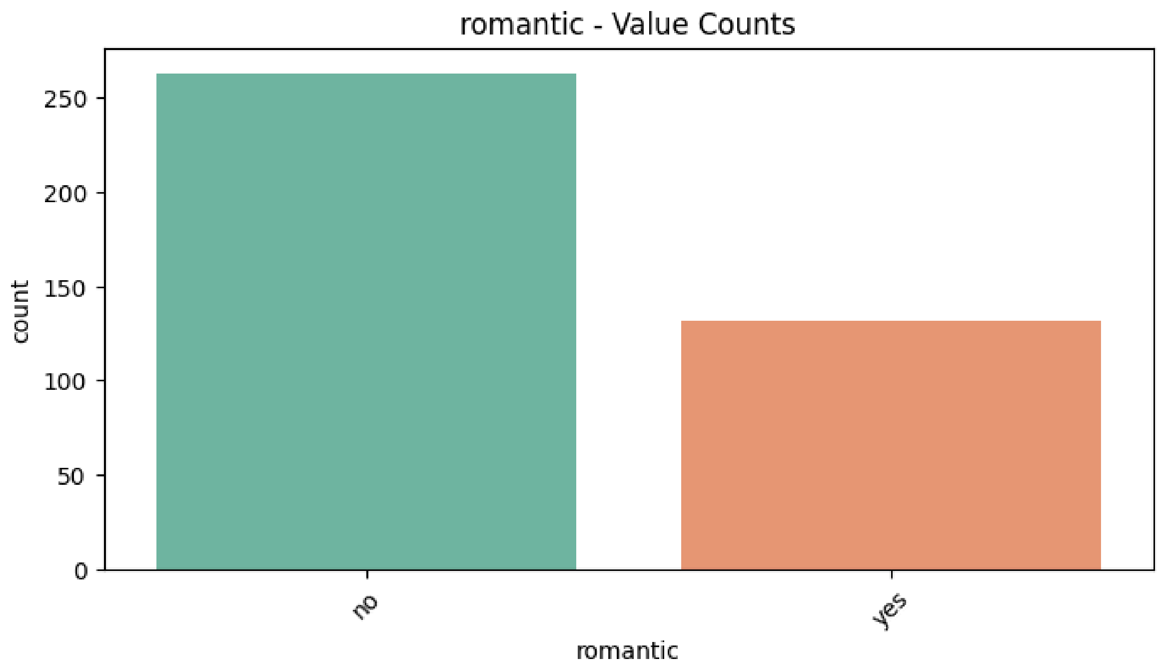

```
Value Counts for nursery:
 nursery
yes    314
no      81
Name: count, dtype: int64
```




```
Value Counts for higher:
 higher
yes    375
no      20
Name: count, dtype: int64
```




```
Value Counts for internet:
 internet
yes    329
no      66
Name: count, dtype: int64
```




```
Value Counts for romantic:
 romantic
no     263
yes    132
Name: count, dtype: int64
```

✅ Observations (Univariate EDA)

Categorical

- School → Majority students belong to GP, fewer from MS.
- Sex → Fairly balanced, but females slightly more than males.
- Address → More students live in Urban (U) areas compared to Rural (R).
- Family Size → Larger families (GT3) more common than smaller (LE3).
- Parent Status (Pstatus) → More students' parents are together (T) than apart (A).
- Guardian → Mother is most common guardian, followed by father, then other.
- Activities, Nursery, Internet →
- Participation in activities: almost equal yes/no
- Nursery attendance: more had nursery schooling
- Internet access: more have it than not

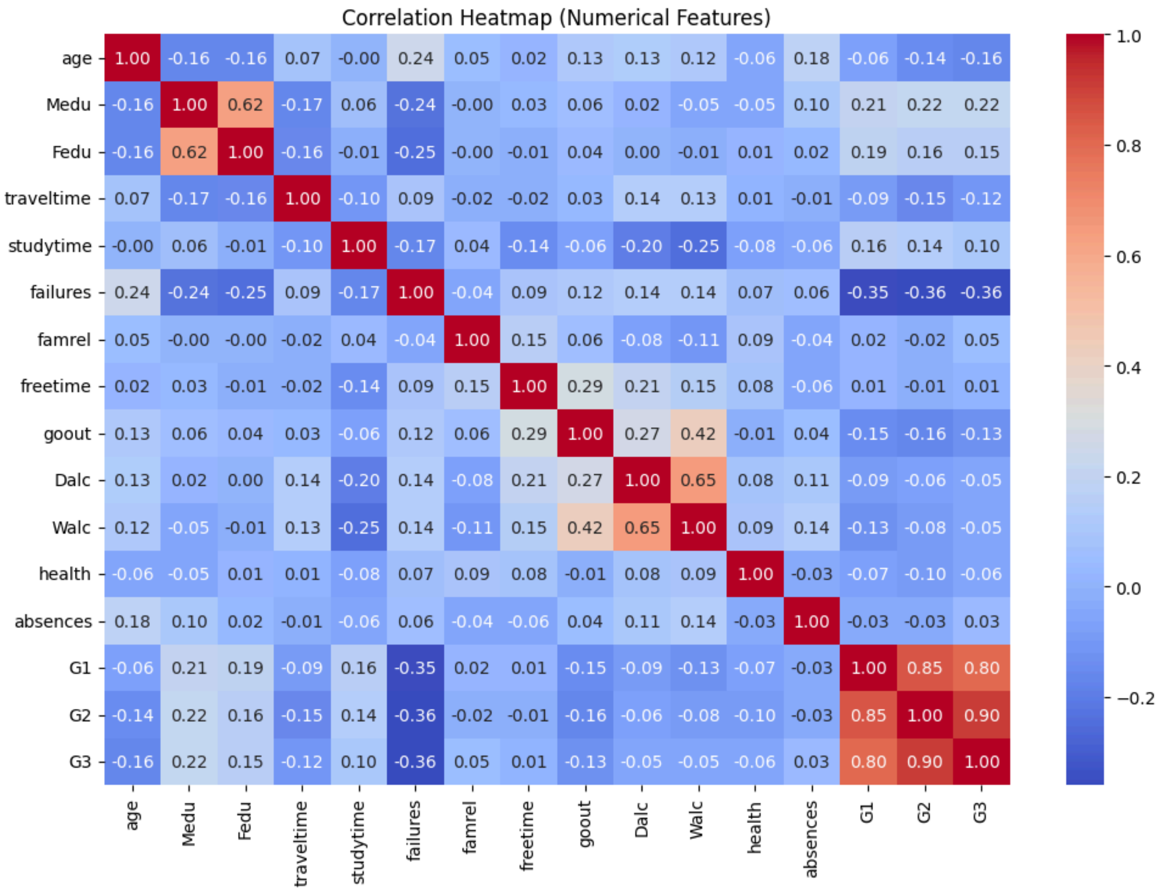
Numerical

- Age → Concentrated around mid–teen range (15–18).

- Absences → Heavily skewed with a few extreme outliers (some students missing dozens of classes).
- Grades (G1, G2, G3) → Not normally distributed, tend to cluster toward the lower end (many students performing below average).
- Other numeric features (studytime, freetime, goout, Dalc, Walc, health) are ordinal scales (discrete 1–5 ratings) rather than continuous numbers — so treat them almost like categorical when analyzing.

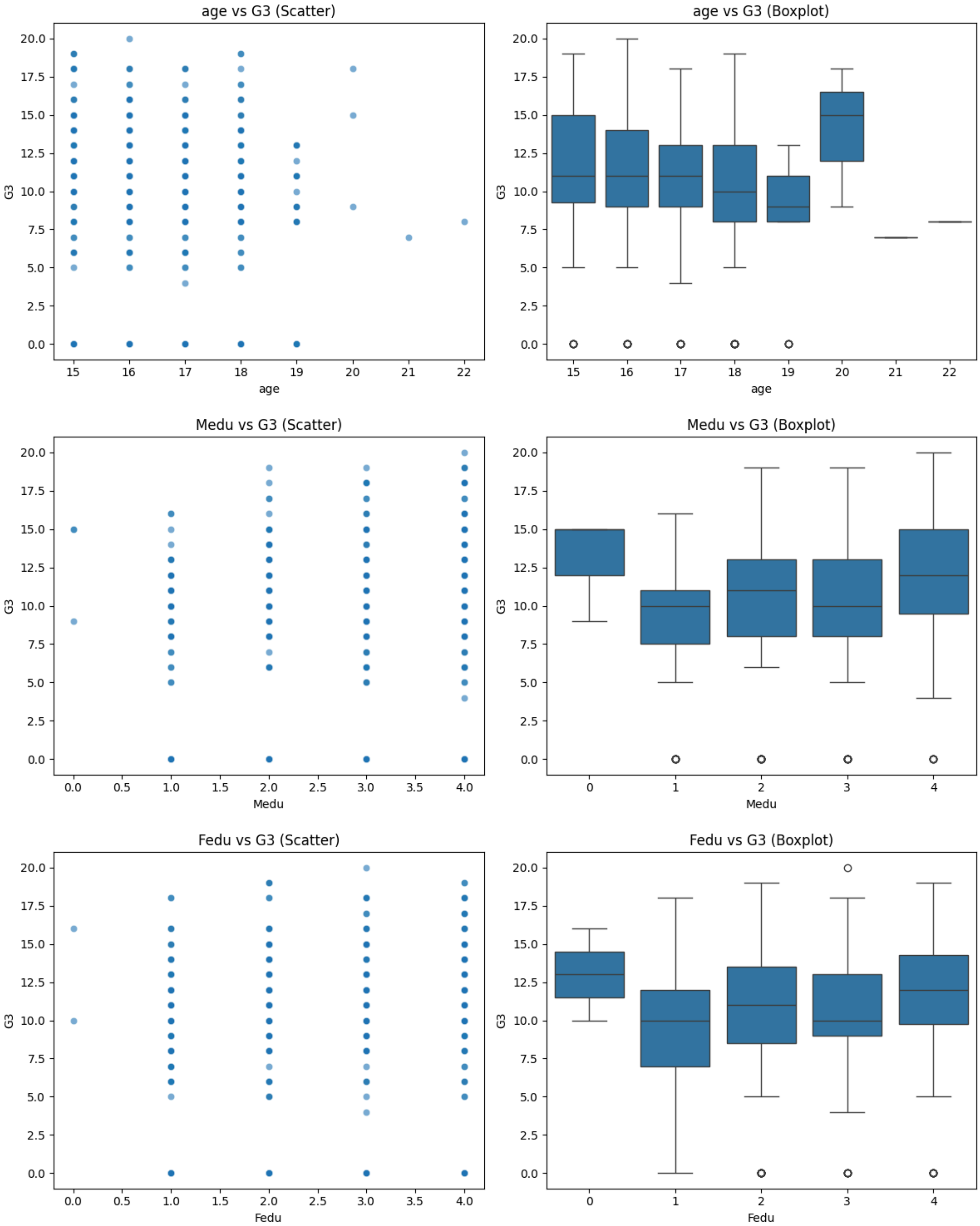
Bivariate Analysis

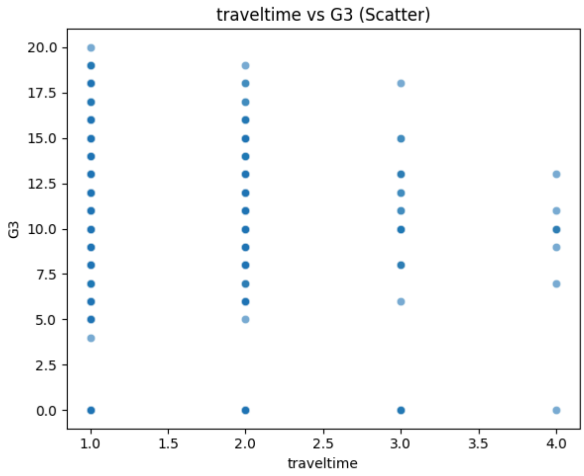
In [18]:
```
target = "G3"
# 1. Correlation Heatmap (numerical vs numerical, including target)
plt.figure(figsize=(12,8))
sns.heatmap(df[num_col].corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap (Numerical Features)")
plt.show()
```

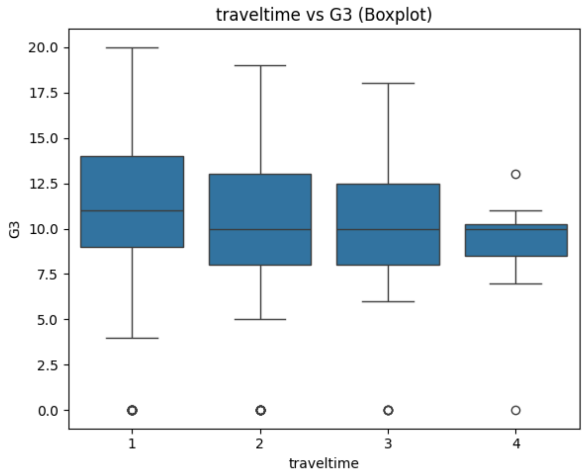


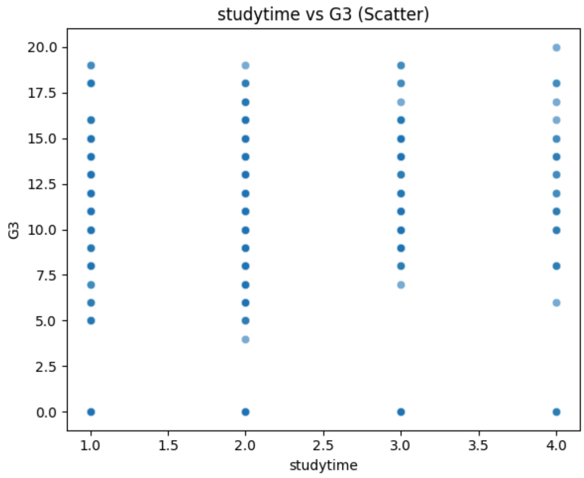
In [19]:
```
# 2. Numerical Features vs Target (scatter + boxplot)
for col in [c for c in num_col if c != target]:
    plt.figure(figsize=(12,5))

    # Scatterplot
    plt.subplot(1,2,1)
    sns.scatterplot(x=df[col], y=df[target], alpha=0.6)
    plt.title(f"{col} vs {target} (Scatter)")

    # Boxplot (discretize col if it's ordinal like studytime, freetime, etc.)
```

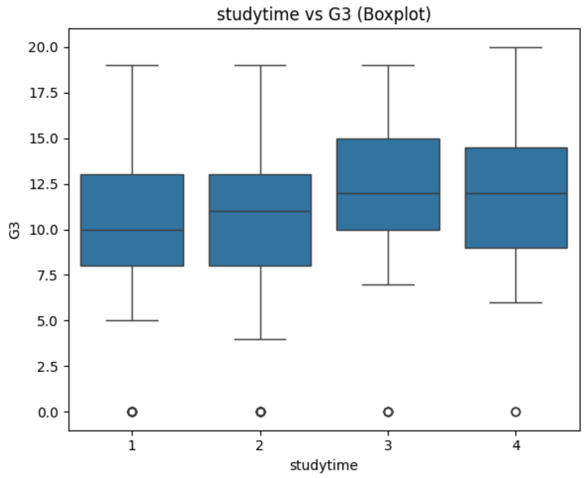
```
    plt.subplot(1,2,2)
    sns.boxplot(x=df[col], y=df[target])
    plt.title(f"{col} vs {target} (Boxplot)")

    plt.tight_layout()
    plt.show()
```

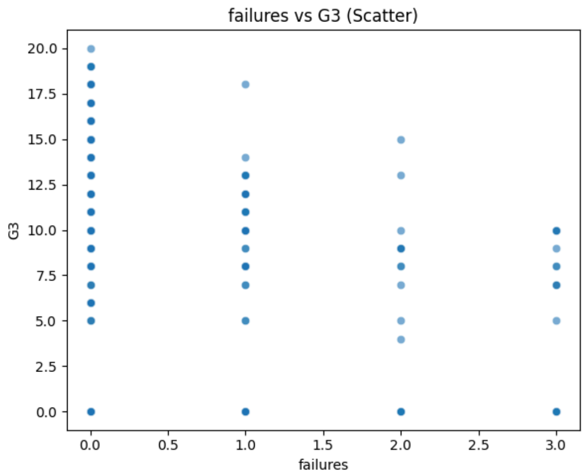


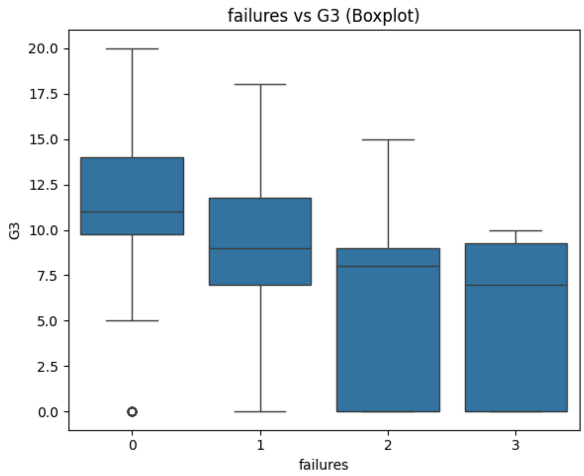

traveltime vs G3 (Scatter)
G3
traveltime

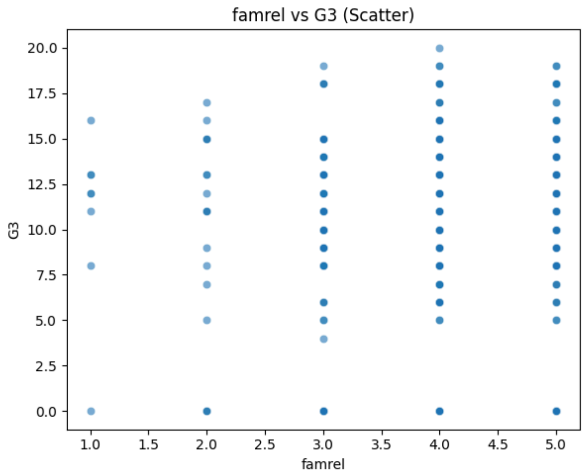

traveltime vs G3 (Boxplot)
G3
traveltime

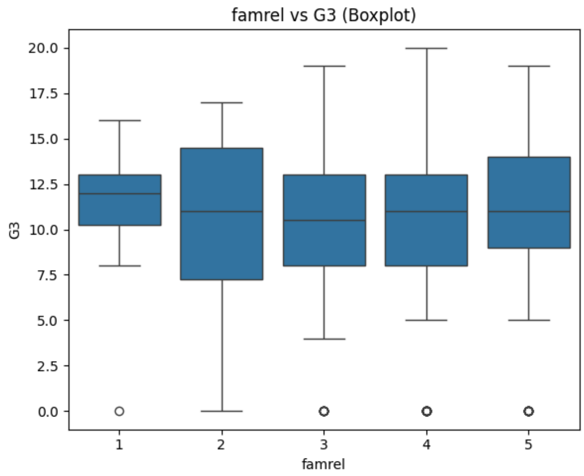

studytime vs G3 (Scatter)
G3
studytime

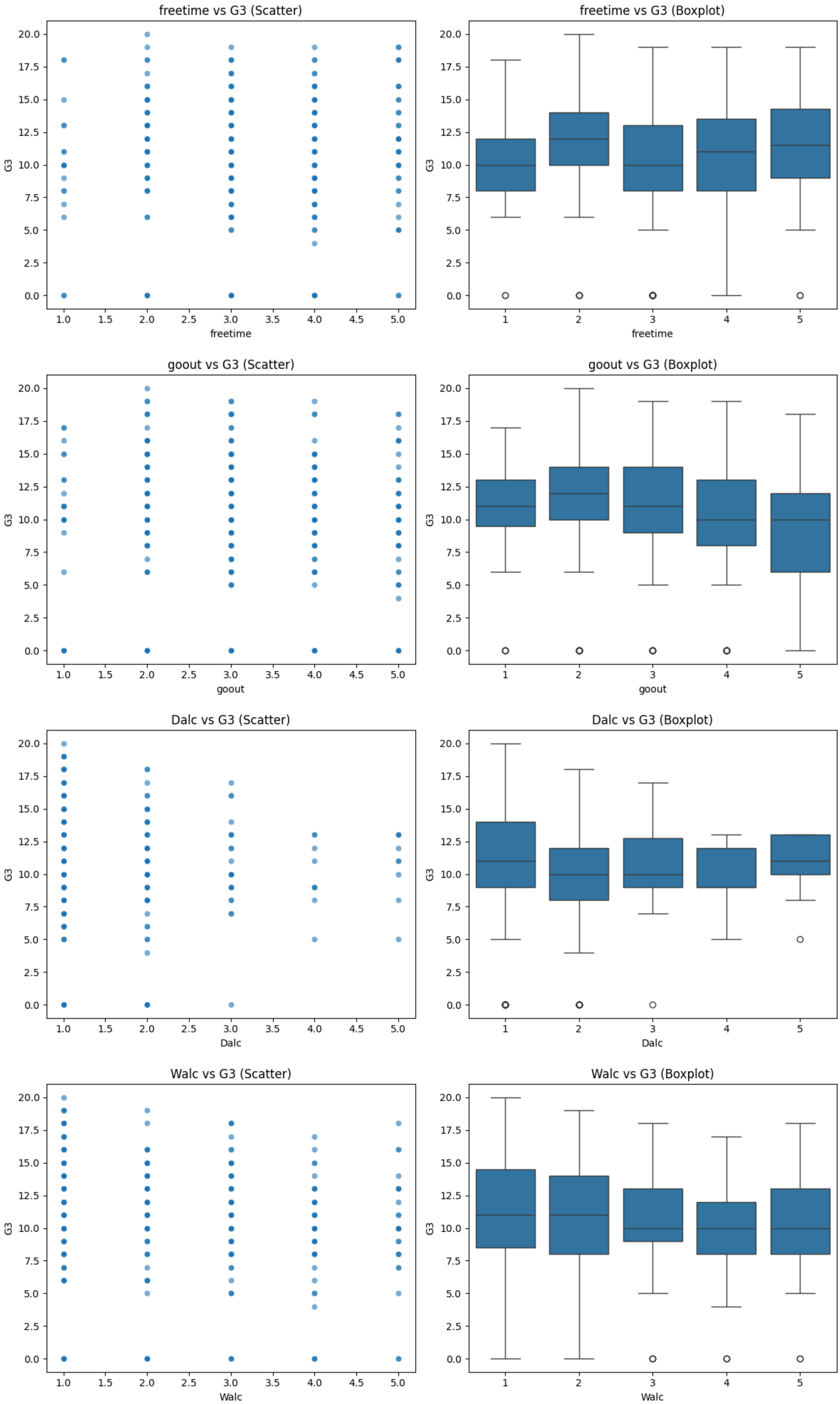

studytime vs G3 (Boxplot)
G3
studytime

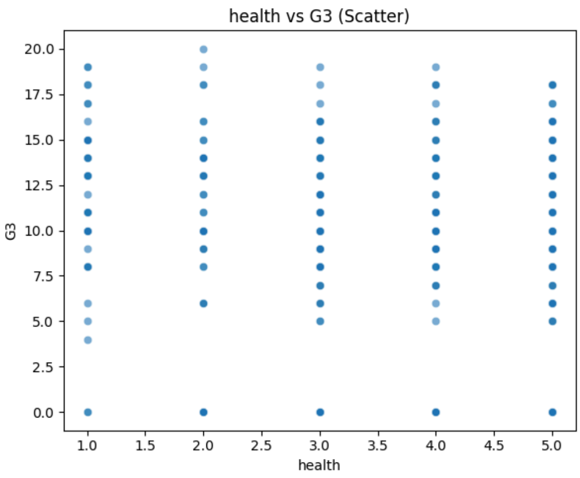

failures vs G3 (Scatter)
G3
failures

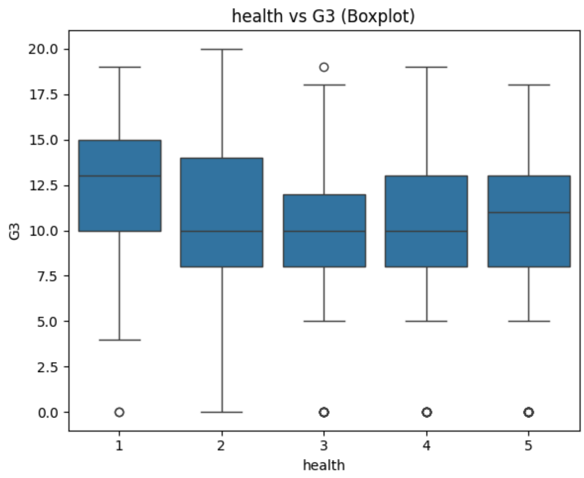

failures vs G3 (Boxplot)
G3
failures

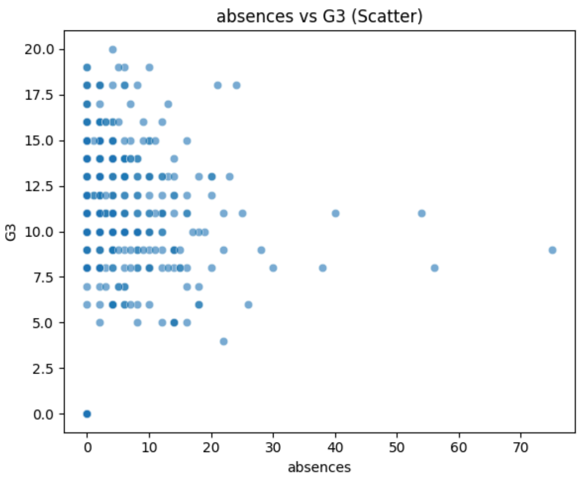

famrel vs G3 (Scatter)
G3
famrel

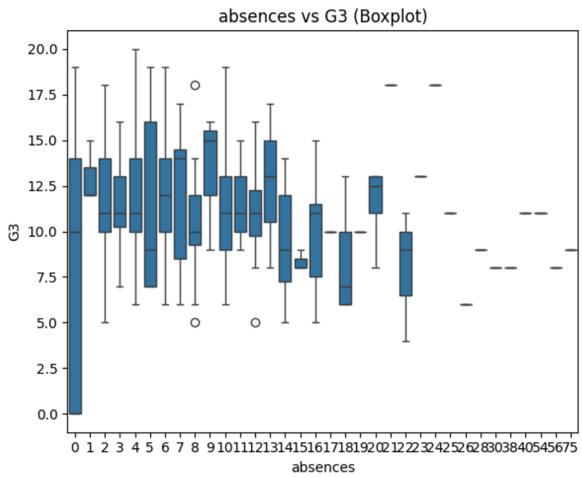

famrel vs G3 (Boxplot)
G3
famrel

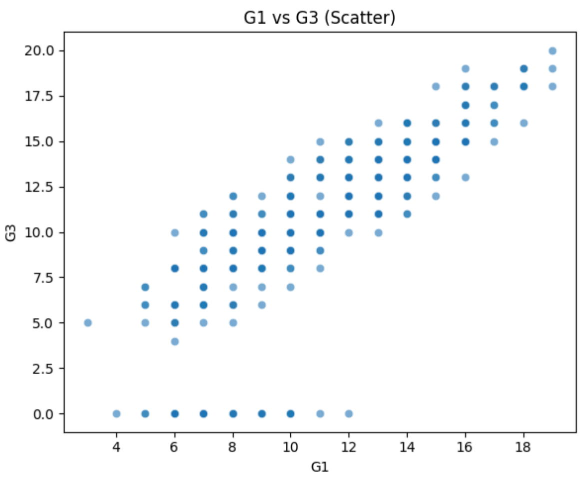

freetime vs G3 (Scatter)
freetime vs G3 (Boxplot)
goout vs G3 (Scatter)
goout vs G3 (Boxplot)
Dalc vs G3 (Scatter)
Dalc vs G3 (Boxplot)
Walc vs G3 (Scatter)
Walc vs G3 (Boxplot)
G3
freetime
goout
Dalc
Walc

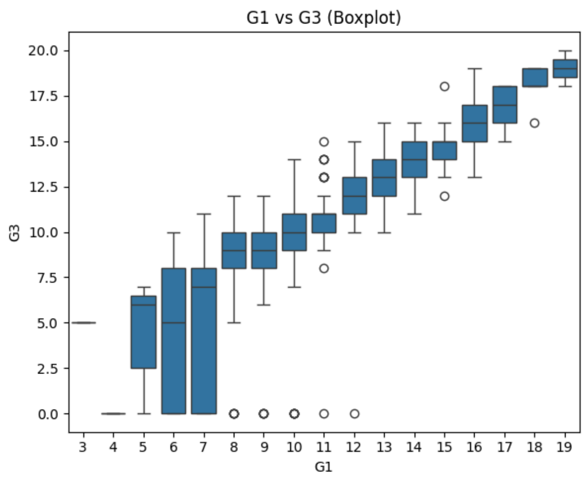

health vs G3 (Scatter)
G3
health

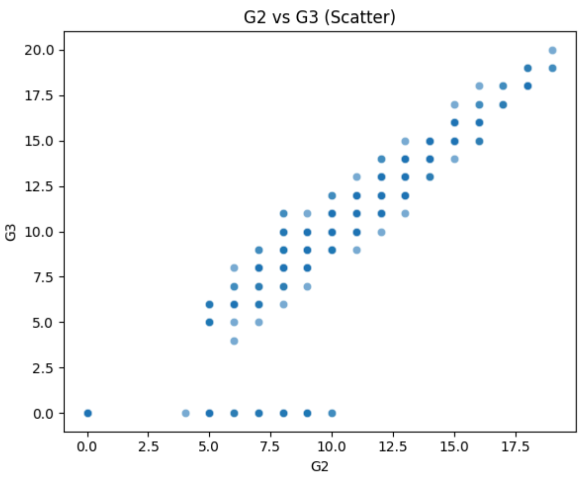

health vs G3 (Boxplot)
G3
health

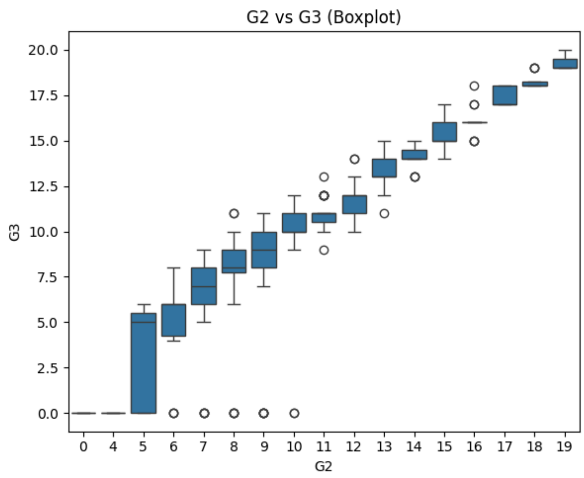

absences vs G3 (Scatter)
G3
absences


absences vs G3 (Boxplot)
G3
absences

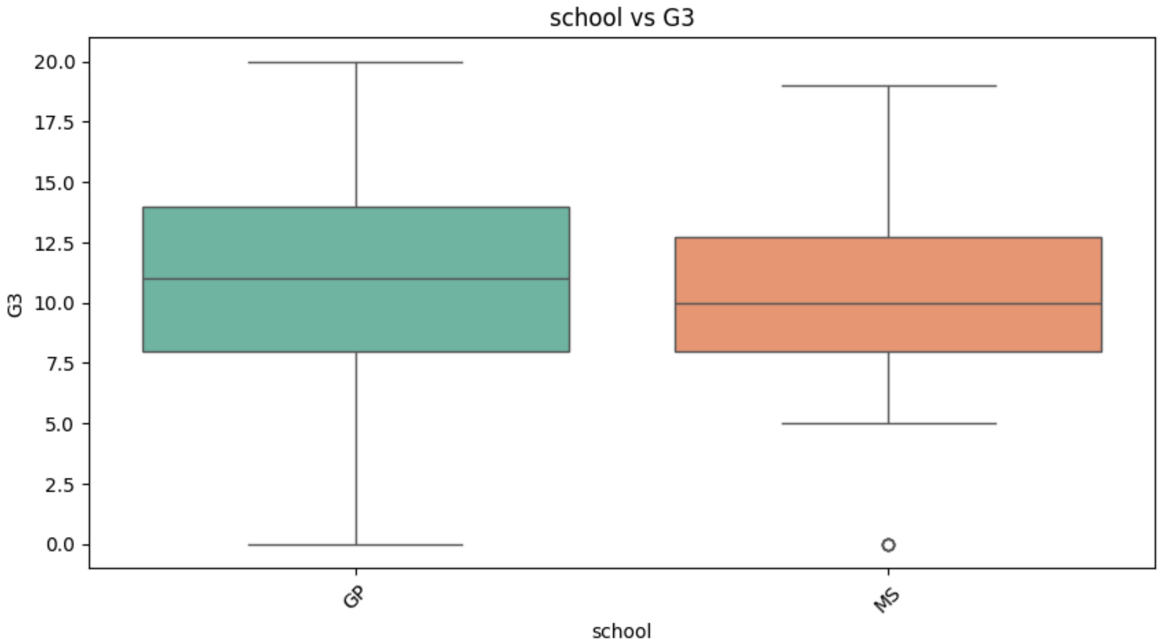

G1 vs G3 (Scatter)
G3
G1

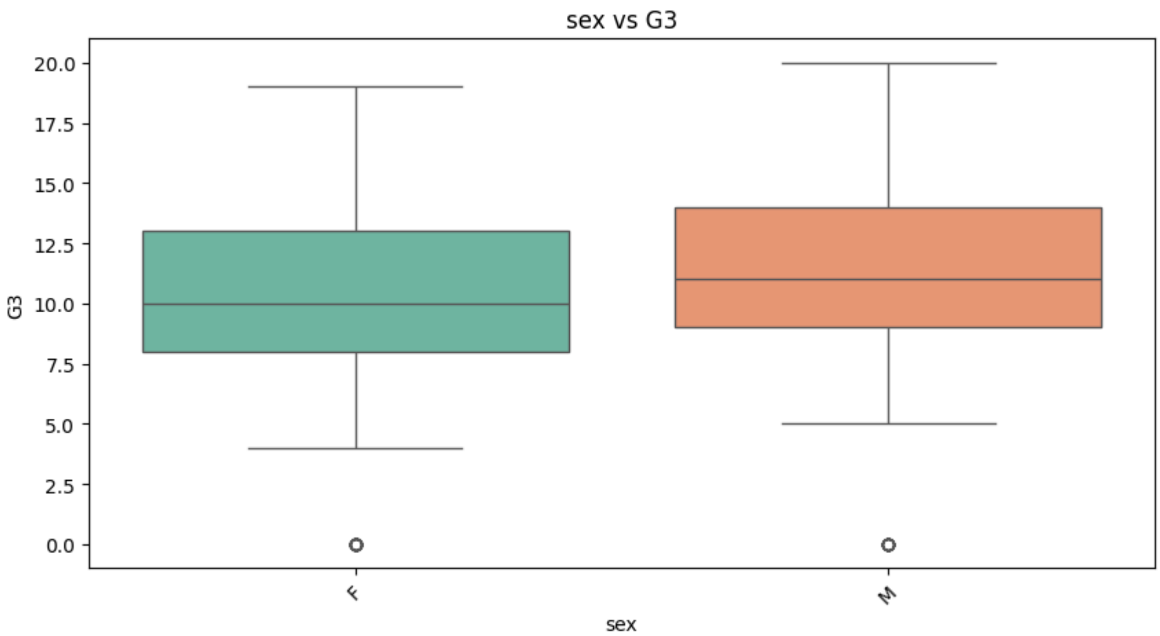

G1 vs G3 (Boxplot)
G3
G1

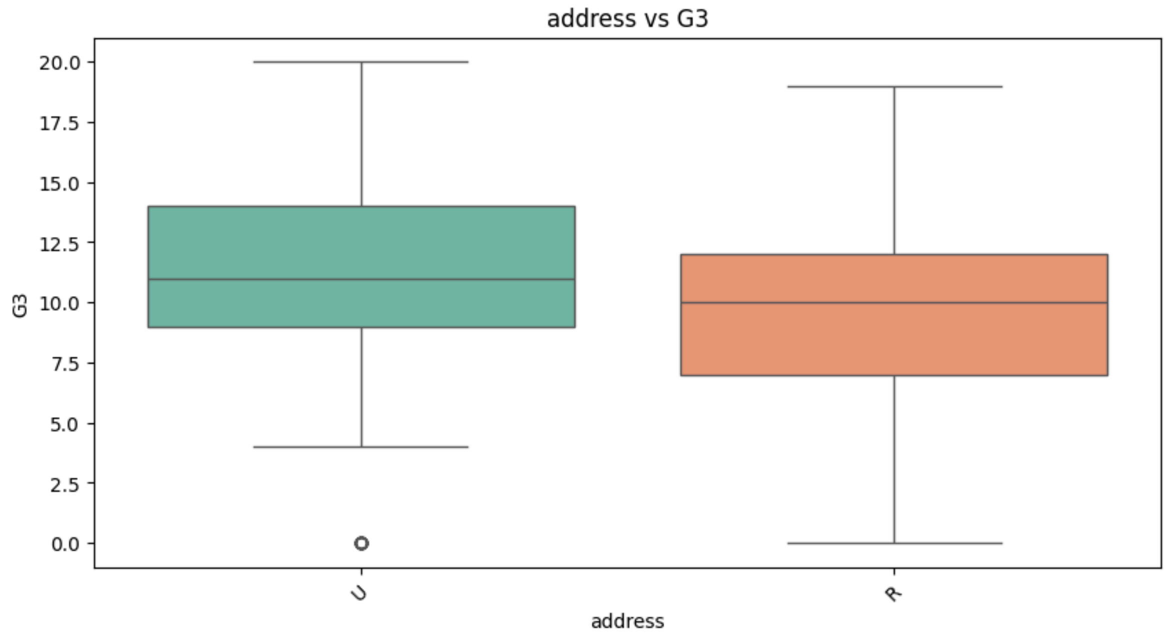

G2 vs G3 (Scatter)
G3
G2

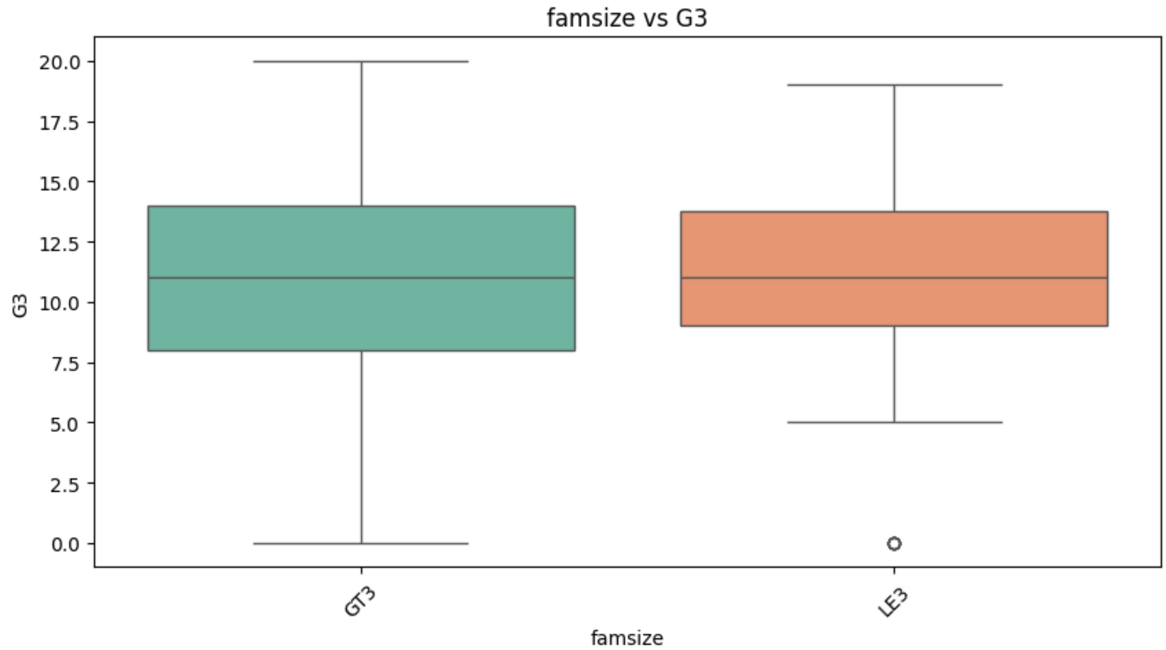

G2 vs G3 (Boxplot)
G3
G2

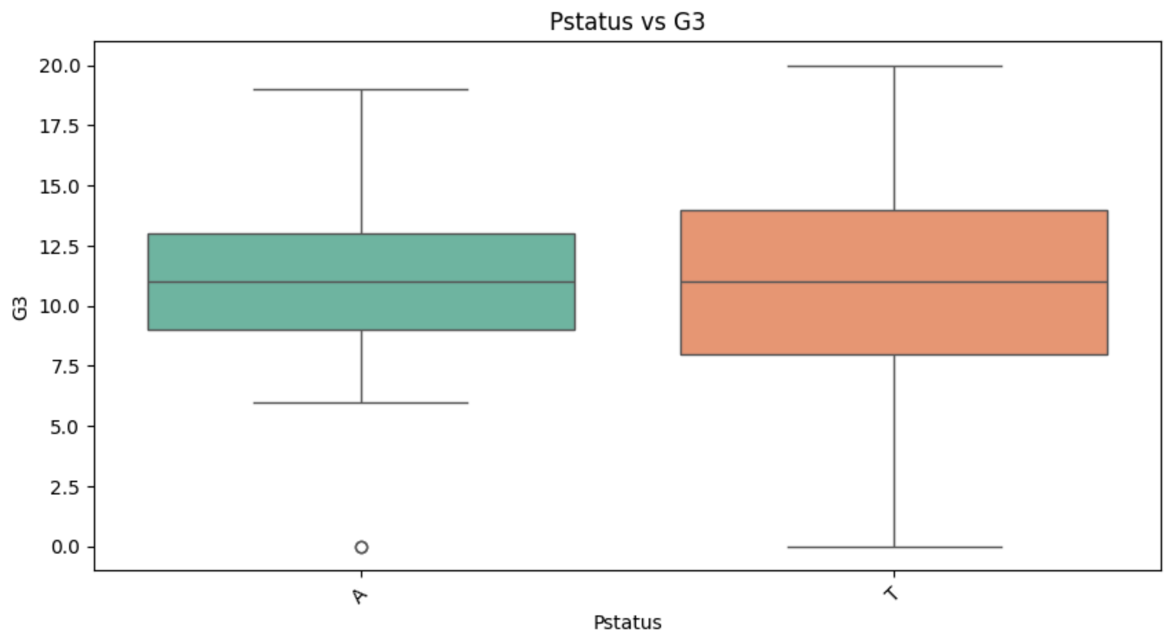
In [20]:
```
for col in cat_col:
    plt.figure(figsize=(10,5))
    sns.boxplot(x=df[col], y=df[target], palette="Set2")
    plt.title(f"{col} vs {target}")
    plt.xticks(rotation=45)
    plt.show()
```

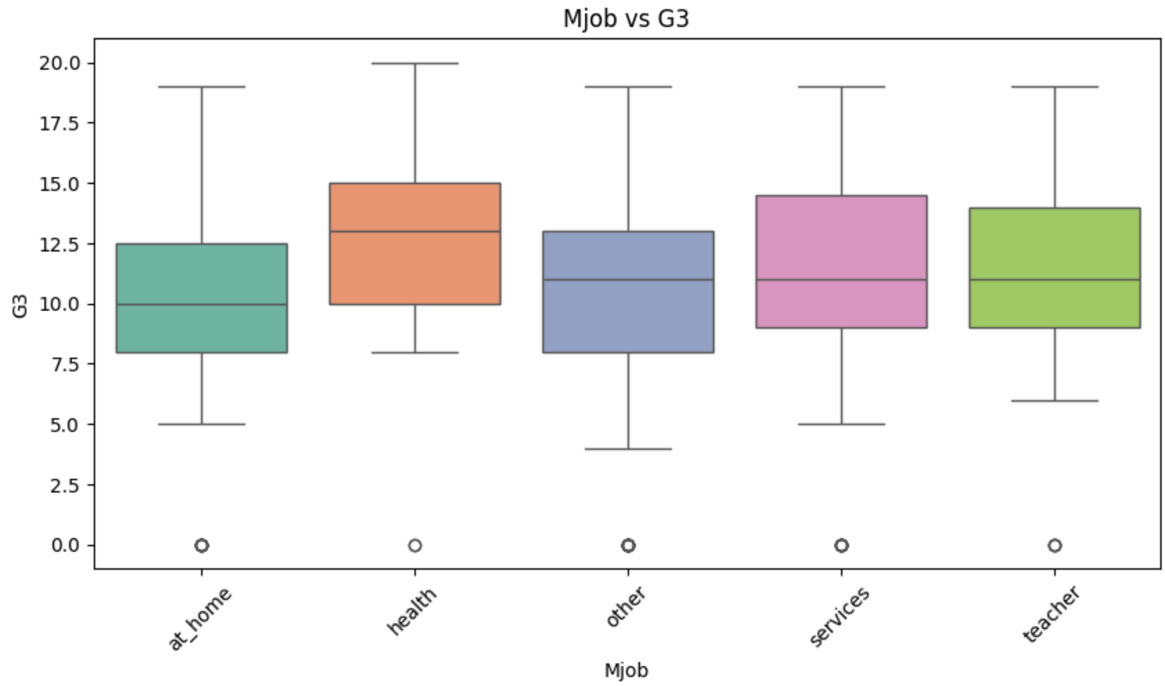


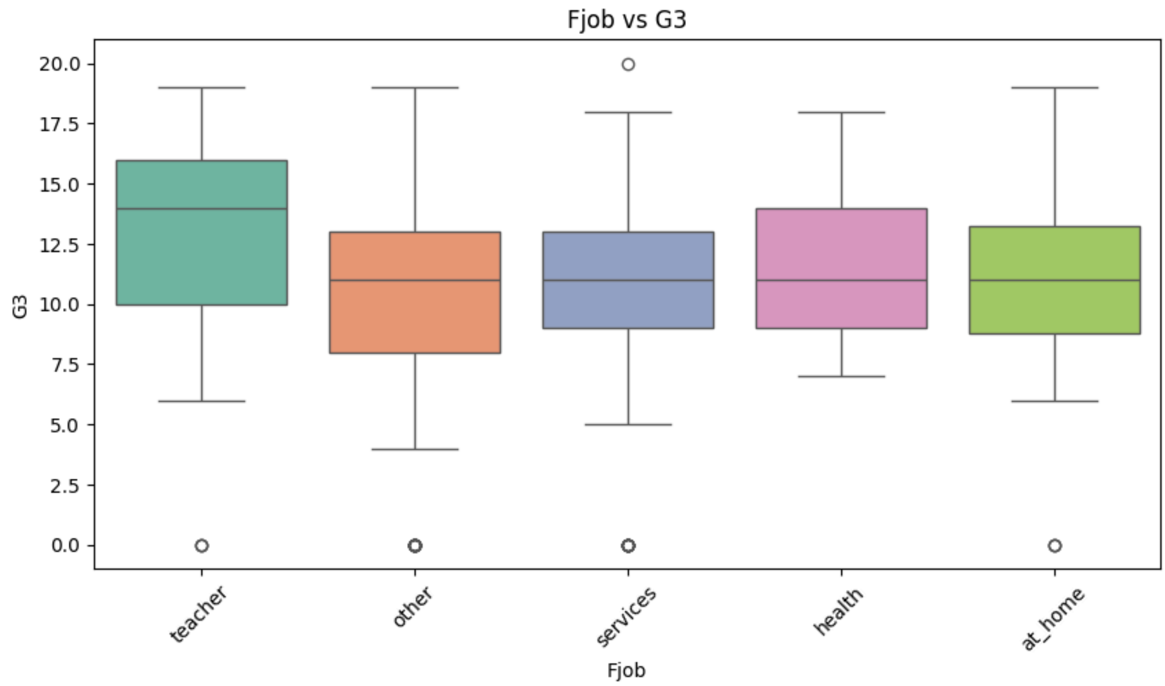


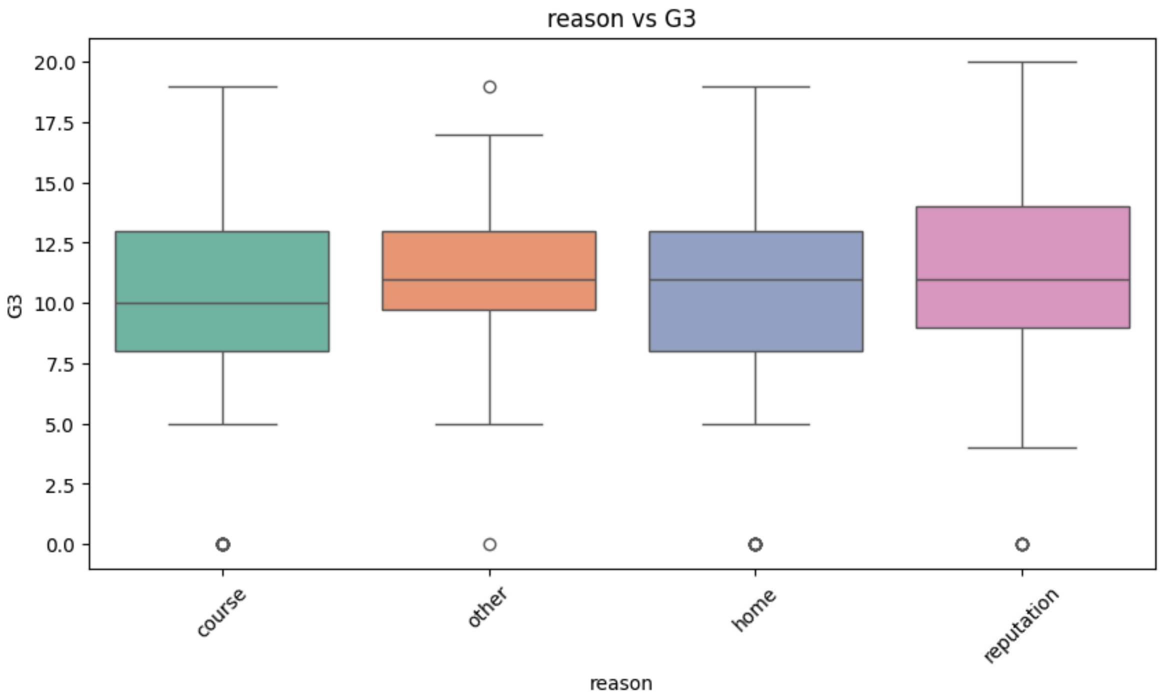

address vs G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
G3
U
R
address

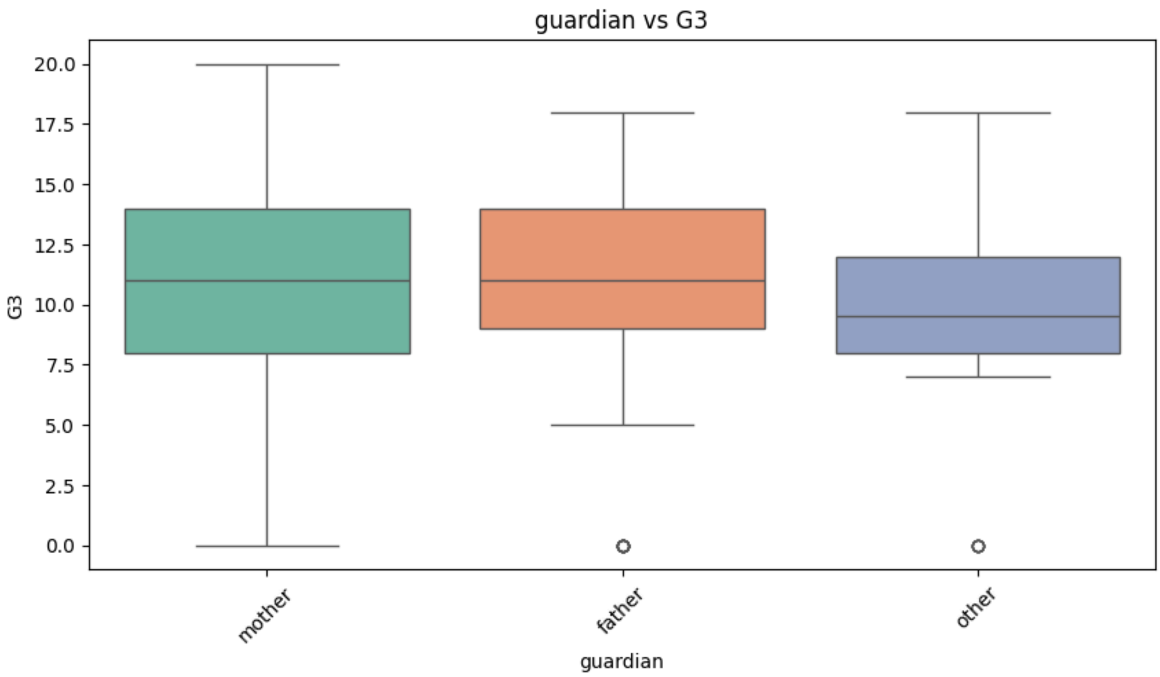

famsize vs G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
G3
GT3
LE3
famsize

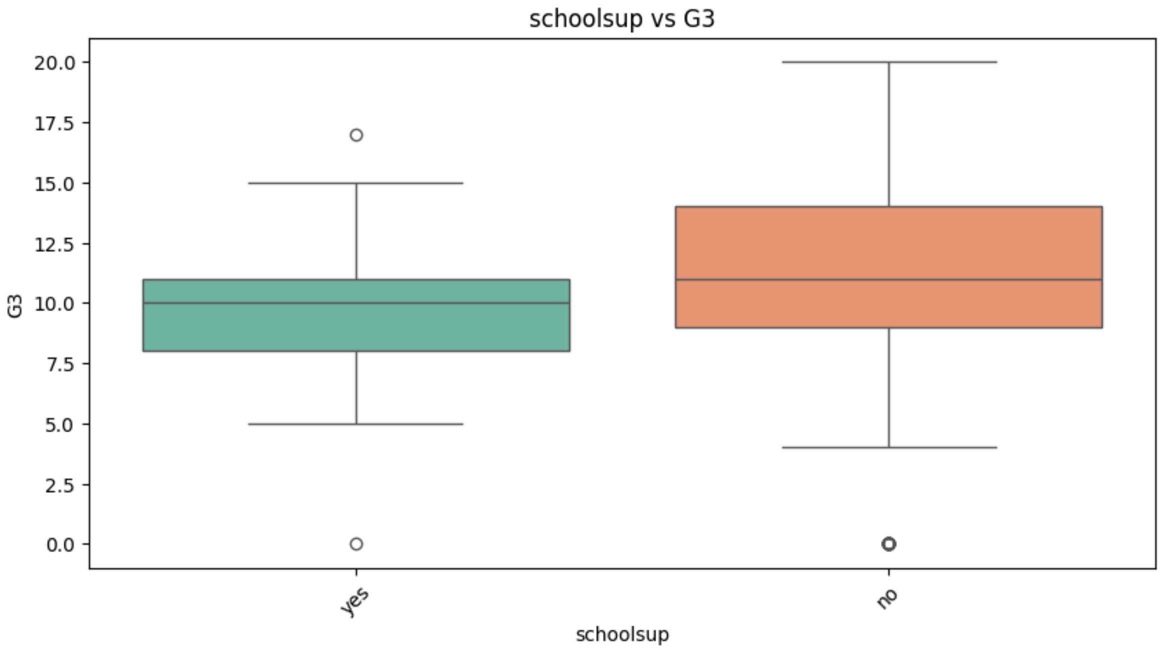

Pstatus vs G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
G3
A
T
Pstatus

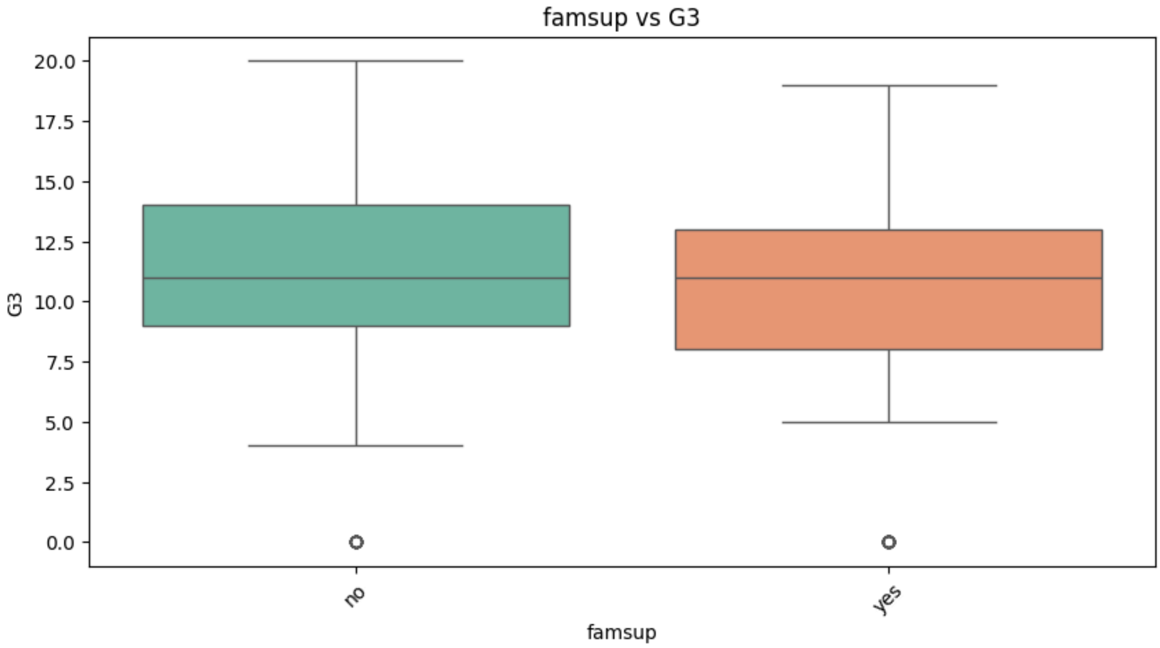

Mjob vs G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
G3
at_home
health
other
services
teacher
Mjob

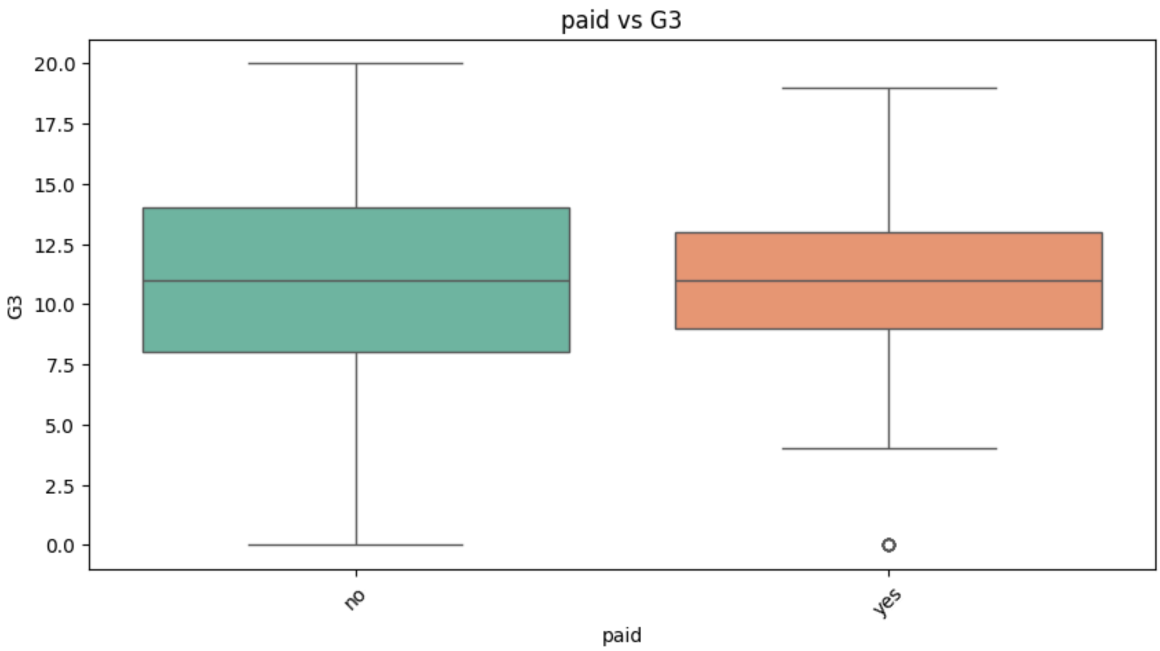

Fjob vs G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
G3
teacher
other
services
health
at_home
Fjob

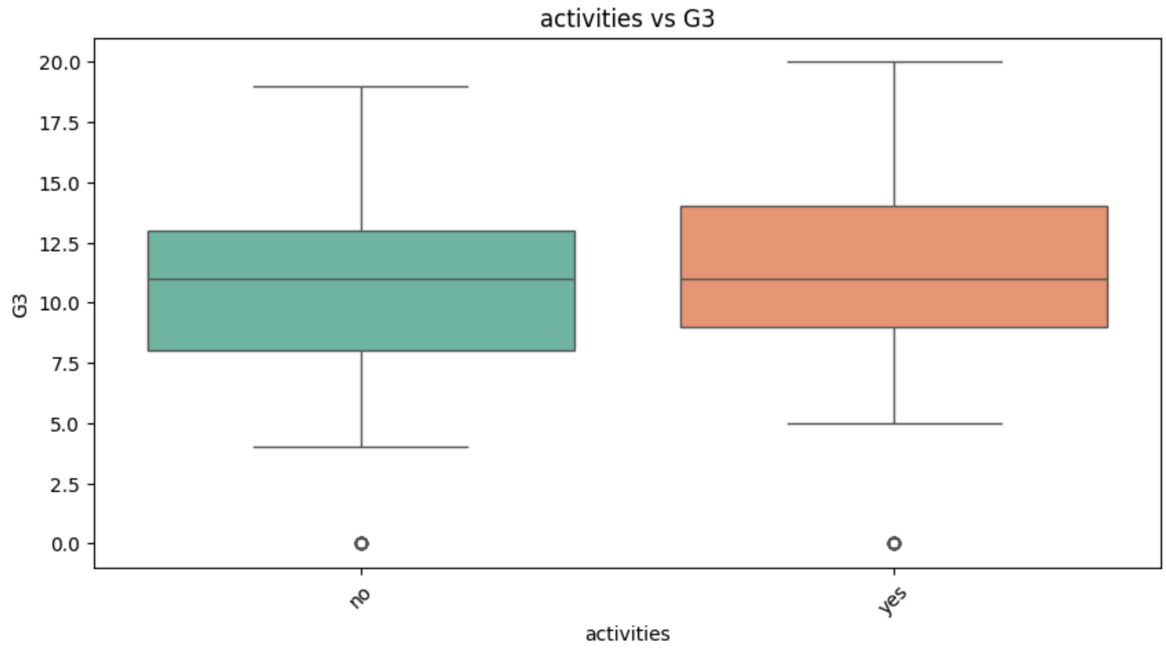

reason vs G3
G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
course
other
home
reputation
reason

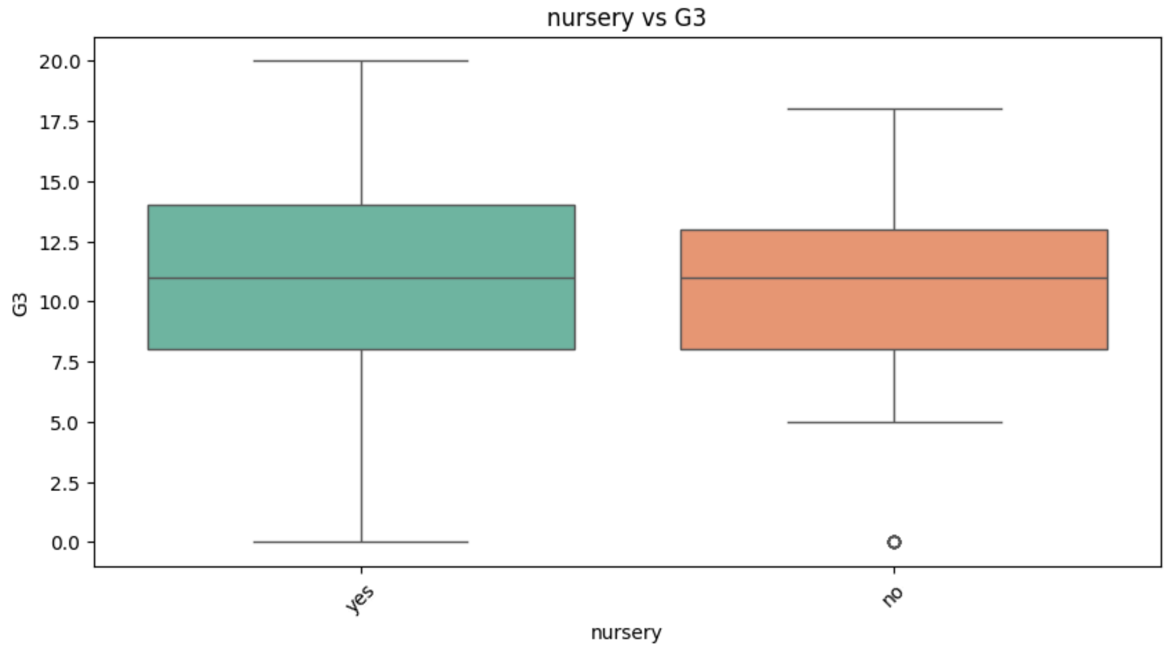

guardian vs G3
G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
mother
father
other
guardian

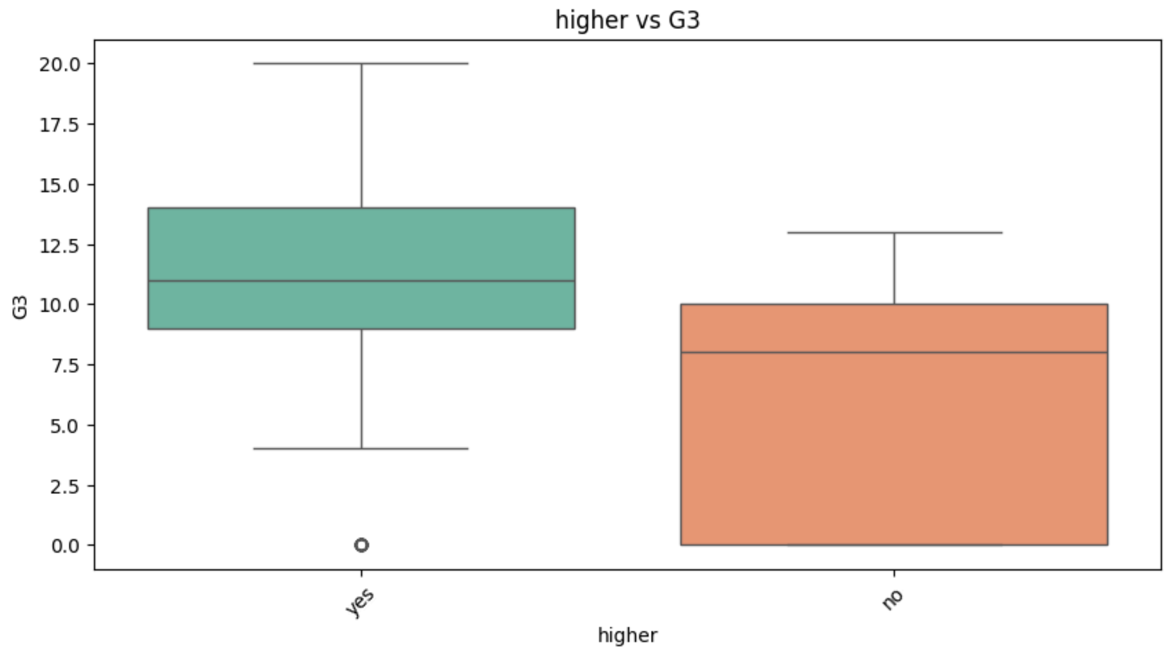

schoolsup vs G3
G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
yes
no
schoolsup

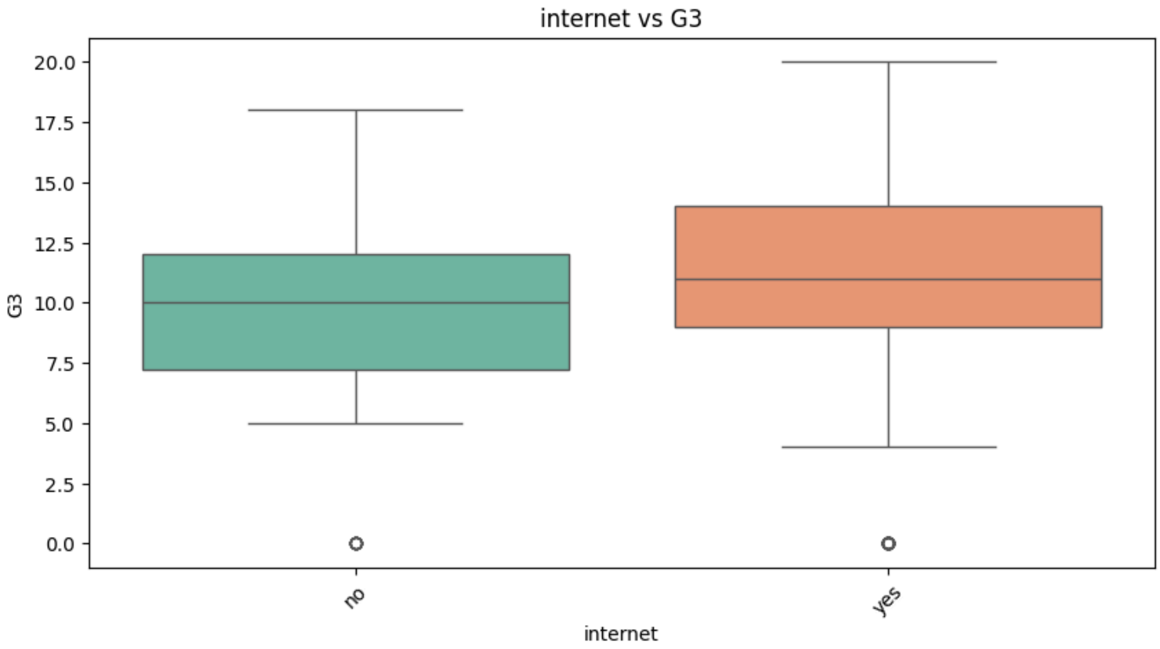

famsup vs G3
G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
no
yes
famsup

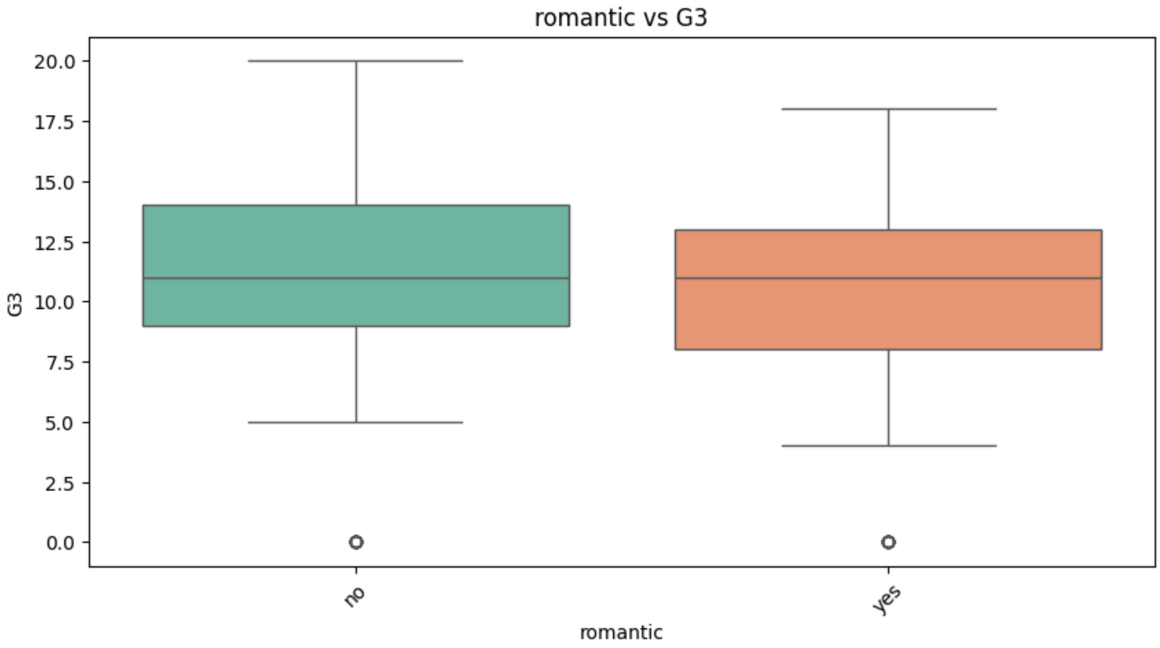

paid vs G3
G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
no
yes
paid


activities vs G3
G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
no
yes
activities


nursery vs G3
G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
yes
no
nursery


higher vs G3
G3
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
yes
no
higher





📝 Bivariate Analysis – Notes

🔹 Correlation Heatmap

G1 & G2 are very strongly correlated with G3 (final grade) → as expected, past performance is the best predictor of future performance.

Other numerical variables (studytime, freetime, health, etc.) show little to no linear correlation with G3.

🔹 Numerical vs Target (G3)

Studytime vs G3 → Higher studytime categories are associated with slightly higher grades, but not a strong trend.

Freetime vs G3 → No clear relationship; grades are spread across all freetime levels.

Goout, Dalc (workday alcohol), Walc (weekend alcohol) → Higher values generally linked with slightly lower grades, but effect size is small.

Absences vs G3 → Students with very high absences have lower grades, though many with low absences also perform poorly → not a clean predictor.

🔹 Categorical vs Target (G3)

School (GP vs MS) → Students from GP school slightly outperform those from MS on average.

Sex → Small difference; females tend to have slightly higher grades, but overlap is large.

Address (Urban vs Rural) → Urban students show a slight advantage, but not dramatic.

Guardian → Little difference across categories (mother, father, other).

Parental education & jobs → Some positive trend (higher education = slightly higher grades), but not very strong.

Activities, Internet, Nursery → No significant difference in G3 observed.

✅ Key Takeaways (Bivariate)

Strong Predictors: G1, G2 (past grades).

Moderate Predictors: Absences (negative effect), studytime (slightly positive), alcohol consumption (slightly negative).

Weak/No Clear Relationship: freetime, health, guardian, activities, internet access.

In [ ]:

```

```