Statistical Analysis of Student Performance

Andy Malinsky, Maha Jayapal, and Scott Hogan

Shiley-Marcos School of Engineering, University of San Diego

AAI-500: Probability and Statistics for Artificial Intelligence

Leonid Shpaner, M.S.

June 24, 2024

Statistical Analysis of Student Performance

The goal of this project was to investigate which factors had the greatest impact on final grades. These factors were derived from the selected data set titled “Student Performance,” taken from the UC Irvine Machine Learning Repository (Cortez, 2008). The data involves survey and questionnaire data among students in two Portuguese schools in two distinct subjects: Mathematics (Math) and Portuguese language (Port). The data is split into two tables, with one table per subject. There are a total of 395 students in the Math table and 649 students in the Port table. Both tables have the same 30 independent features and 3 dependent features. For our analysis, the independent features were divided into three categories: Social, Demographic, and Academic. The dependent features represent first period (G1), second period (G2), and final grades (G3). Code for this project is provided in the Appendices section (Malinsky, Jayapal, & Hogan, 2024). We performed statistical analysis in each of these independent feature categories to determine the impacts of those features on final grade outcomes.

Data Cleaning and Preparation

For each independent feature category, we conducted data cleaning and preparation. This involved selecting specific features relevant to each category, making sure there were no null or missing values, and mapping any binary or categorical variables if needed. Tables of each feature selected are provided.

Social Factors Data Preparation

The data set has 33 features including three features for the first term grade, second term grade, and the final grade. The final grade was used as the target variable to study the impact of social life. Among the 30 features related to demographics, social, and school, domain knowledge was used to make feature selection. Features associated with social life are listed in Table 1. The target variable G3, representing final grades, is an integer ranging from 0 to 20. The yes and no of the binary variables internet and romantic were changed to 1 and 0 respectively to be able to use it in the model. The variables did not have missing values.

Table 1

Features Selected for Social Factors Analysis

Name	Data Type	Description
internet	binary: yes or no	Access to internet at home
romantic	binary: yes or no	Romantic relationship
famrel	numeric: from 1 - very bad to 5 - excellent	quality of family relationships
freetime	numeric: from 1 - very low to 5 - very high	free time after school
goout	numeric: from 1 - very low to 5 - very high	going out with friends
absences	numeric: from 0 to 93	number of school absences

Academic Factors Data Preparation

Independent features considered “academic factors” are listed in Table 2. For this analysis, we filtered out the common students that are part of both the Math and Port data set, merging on the following identifying factors: school, sex, age, address, famsize, Pstatus, Medu, Fedu, Mjob, Fjob, reason, nursery, and internet. The full data set of 33 features was filtered down to 8 selected independent features, as well as three dependent features G1, G2, and G3. Both filtered data sets contain information on the same 382 students. We also applied mapping on Boolean values (yes = 1, no = 0) and binary values such as school (GP = 1, MS = 0). All features are shared in the Math and Port data sets. There were no null or missing values.

Table 2

Features Selected for Academic Factors Analysis

Name	**Data Type**	**Description**
school	binary: "GP" - Gabriel Pereira or "MS" - Mousinho da Silveira	student's school
studytime	numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours	weekly study time
failures	numeric: n if 1 < = n < 3, else 4	number of past class failures
schoolsup	binary: yes or no	extra educational support
famsup	binary: yes or no	family educational support
higher	binary: yes or no	wants to take higher education

Table 2 (continued)

Features Selected for Academic Factors Analysis

Name	Data Type	Description
paid	binary: yes or no	extra paid classes within the course subject (Math or Portuguese)

Demographic Factors Data Preparation

Among 33 available features within our data set, 10 were “demographic” in nature and are listed in Table 3. Features selected include student’s sex, student’s age, family size, parent’s cohabitation status, father’s education, mother’s education, mother’s job, father’s job, student’s guardian, and current health status. The response variable selected was the final grade G3. The features that are categorical were transformed into binary data. The first step was to change the variables for “Mother’s Education” (Medu), “Father’s Education” (Fedu), and “health” into strings. Every demographic feature was either already in binary format, or considered a string, but Medu, Fedu, and health were considered integers because of their ordinal nature. We decided to convert these three features into strings so that they may be encoded with “dummy variables” instead. Had we not done this, the regression model would have treated those features differently. Once these three features were converted into strings, all 11 chosen features were converted into “dummy variables” of ones and zeros.

Table 3

Features Selected for Demographic Factors Analysis

Name	Data Type	Description
sex	binary: "F" - female or "M" – male	student's sex
age	numeric: from 15 to 22	student's age
famsize	binary: "T" - living together or "A" - apart	parent's cohabitation status
Pstatus	binary: yes or no	extra educational support
Fedu	numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education	father's education
Medu	numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education	mother's education
guardian	nominal: "mother", "father" or "other"	student's guardian

Table 3 (continued)

Features Selected for Demographic Factors Analysis

Name	Data Type	Description
Mjob	nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other"	mother's job
Fjob	nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other"	father's job
health	numeric: from 1 - very bad to 5 - very good	current health status

Exploratory Data Analysis

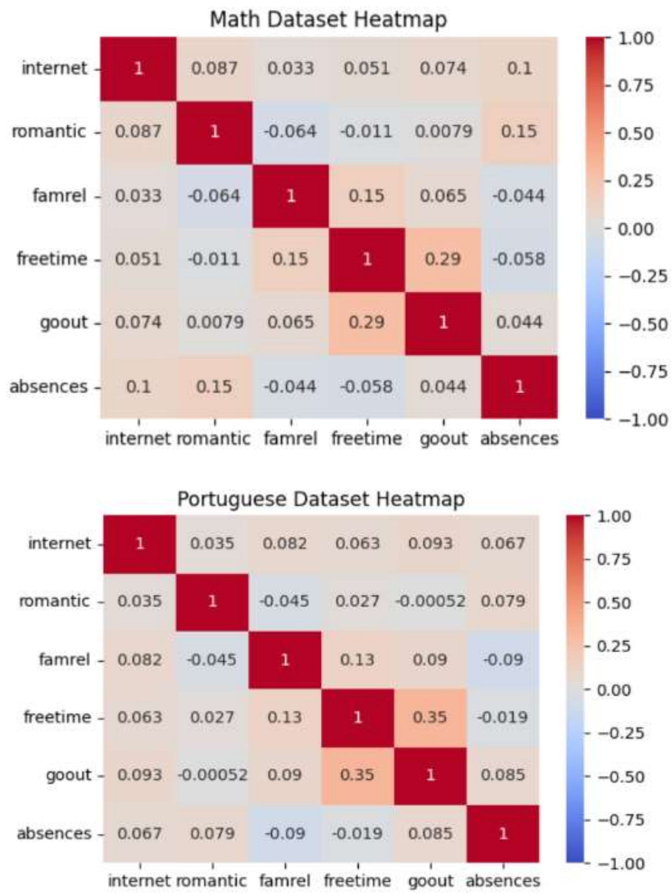
Next, we performed exploratory data analysis within each feature category: social, demographic, and academic. Frequency distributions were plotted and analyzed for the features in each category. Some additional significance tests were performed on the academic features.

Social Factors Exploratory Analysis

The correlation matrix of the selected features for both the Math and Port data sets did not show any multicollinearity, as depicted in Figure 1.

Figure 1

Heatmap of Correlation Matrix for the Selected Social Life Features for the Math and Port Data



The bar plots for the selected features revealed a few interesting insights. For the Port data distributions shown in Figure 2, the internet bar plot showed more than two thirds of the students had access to internet at home. The romantic bar plot revealed that less students were in a romantic relationship. When it comes to family relationships, the data is left skewed with fewer observations to the left. With higher values for family relationships, more students reported a positive family relationship. The distributions for the variables freetime and goout showed a roughly normal distribution. Absences were heavily skewed to the right with more low than high values. The distributions of the Math data were similar to the Port data distributions, as seen in Figure 3.

Figure 2

Bar Plots of the Selected Social Life Features for the Portuguese Data Set

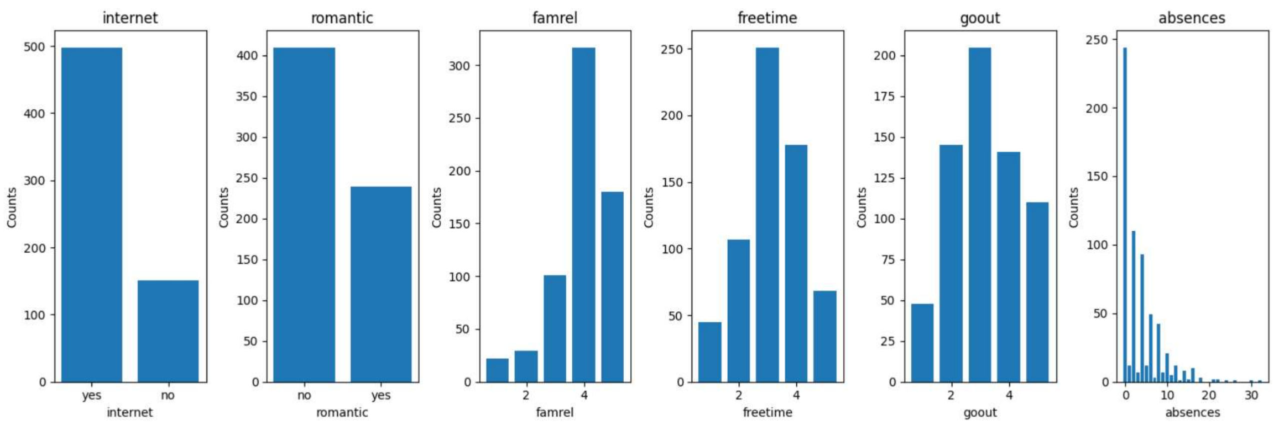
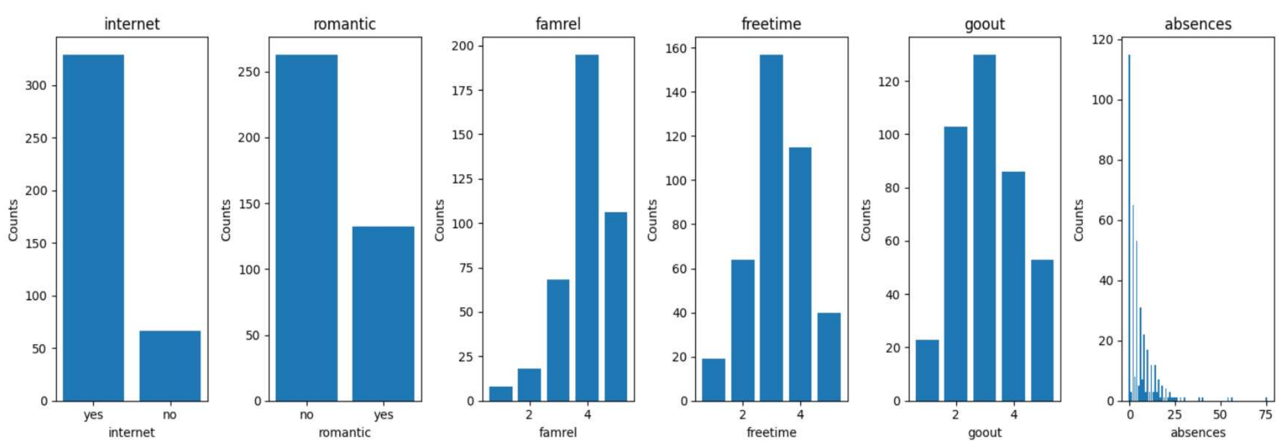


Figure 3

Bar Plots of the Selected Social Life Features for the Math Data Set



The target variable G3 is an integer. The independent variables such as famrel, freetime, goout, and absences were numeric, whereas the variables internet and romantic were a binary of either yes or no. To be used in the model, the yes and no of the variables internet and romantic were changed to 1 and 0, respectively. The variables did not have any missing values.

Demographic Factors Exploratory Analysis

The selected independent demographic features displayed similar frequency distributions in both the Math and Port data sets. Figure 4 displays the frequency distributions of the demographic features for the Math data set. Figure 5 displays the frequency distributions of the demographic features for the Port data set.

Figure 4

Bar Plots of the Selected Demographic Features for the Math Data Set

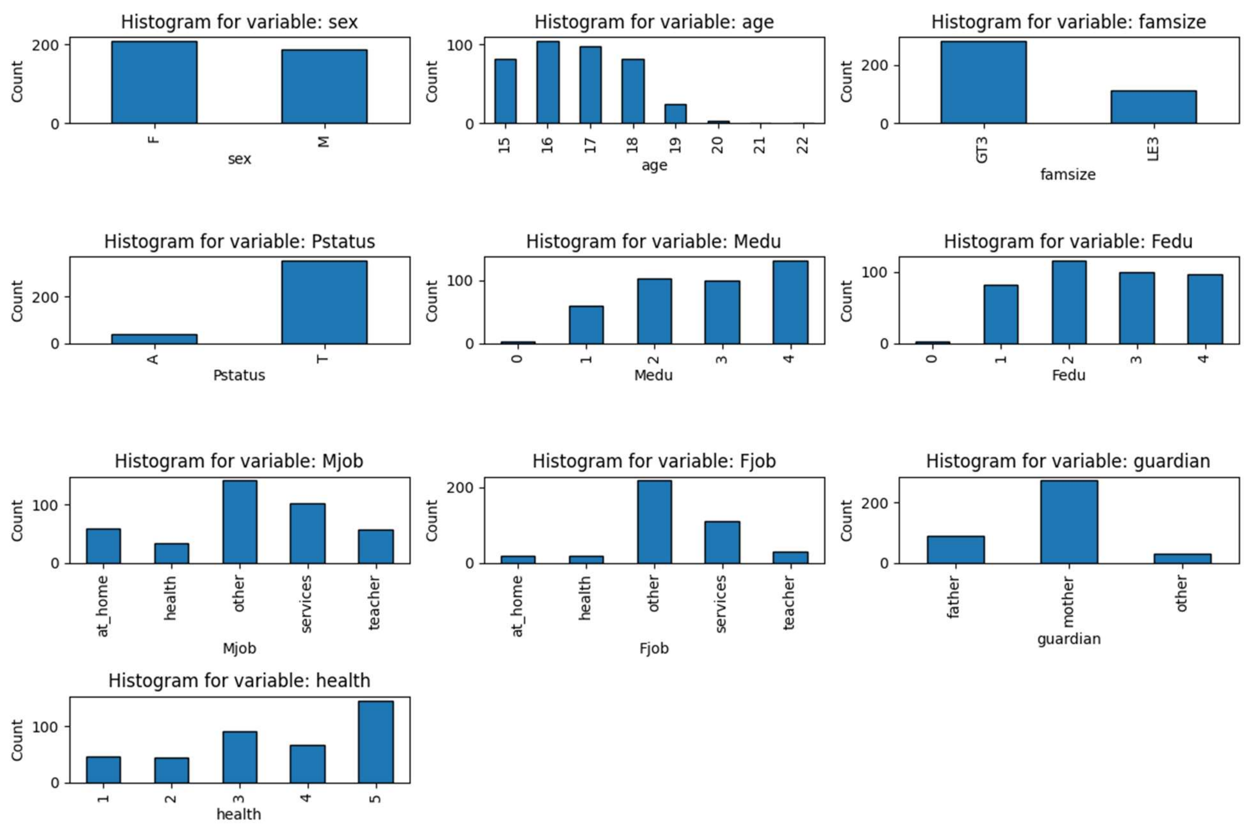
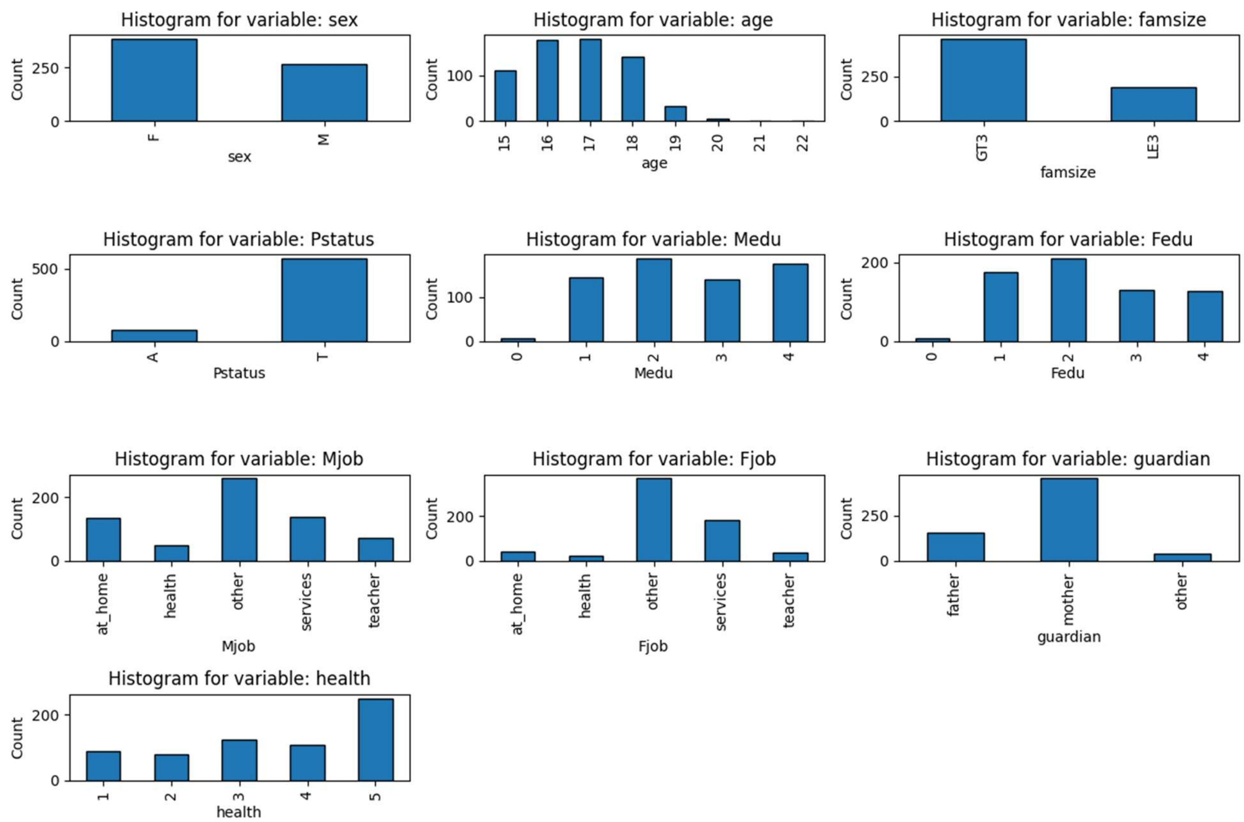


Figure 5

Bar Plots of the Selected Demographic Features for the Port Data Set



Academic Factors Exploratory Analysis

The bar charts of the frequency distributions for the independent features in the Math data are displayed in Figure 6, and the frequency distributions for the Port data are displayed in Figure 7. For both data sets, most students reported a study time of 2–5 hours. Failures are heavily right skewed as most students had zero failures. More students reported not having extra educational support. More students reporting having family educational support. Vastly more students reported wanting to take higher education than not. The main difference between the two data sets is the responses to extra paid classes, with a greater number of students with extra paid classes in math (177) than Portuguese (26). The frequency distribution of the school variable, as shown in Figure 8, depicts that greatly more students attend school 1 than school 0, with 342 students over 40 students, respectively.

Figure 6

Frequency Distributions for each Independent Variable in Math Data Set

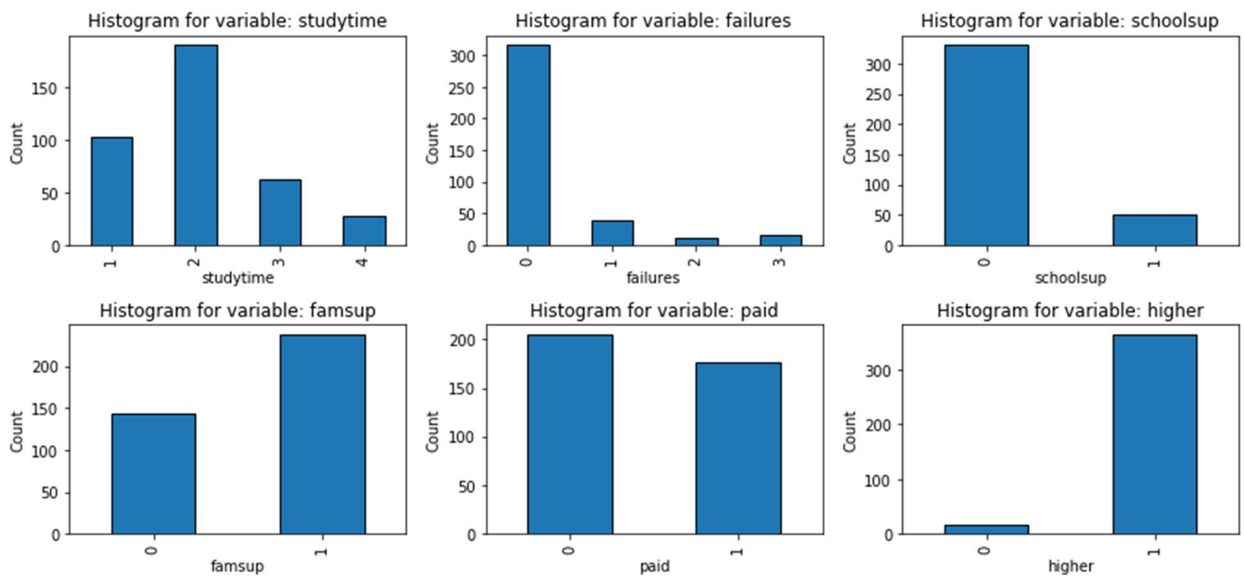


Figure 7

Frequency Distributions for each Independent Variable in the Portuguese Data Set

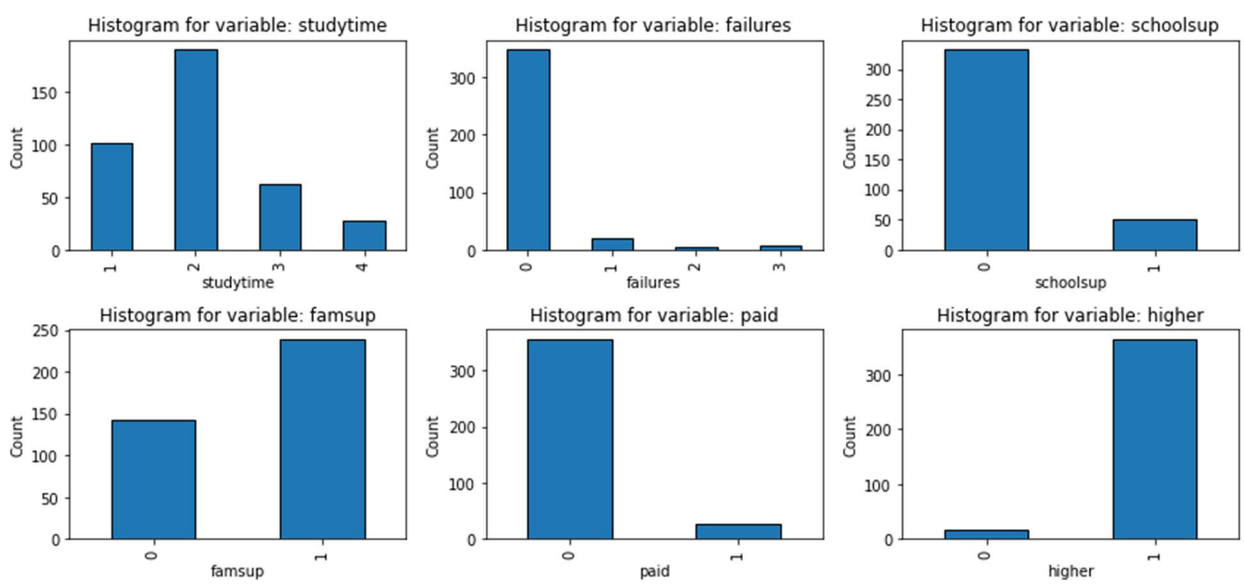
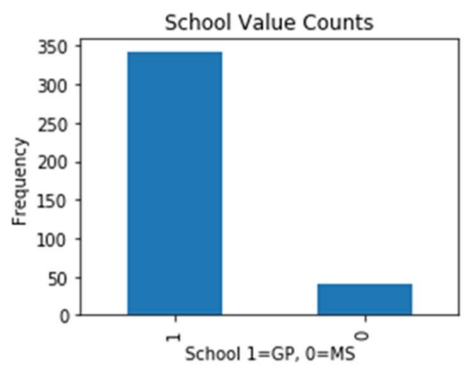


Figure 8

Frequency Distribution for the School Variable in the Merged Data Set

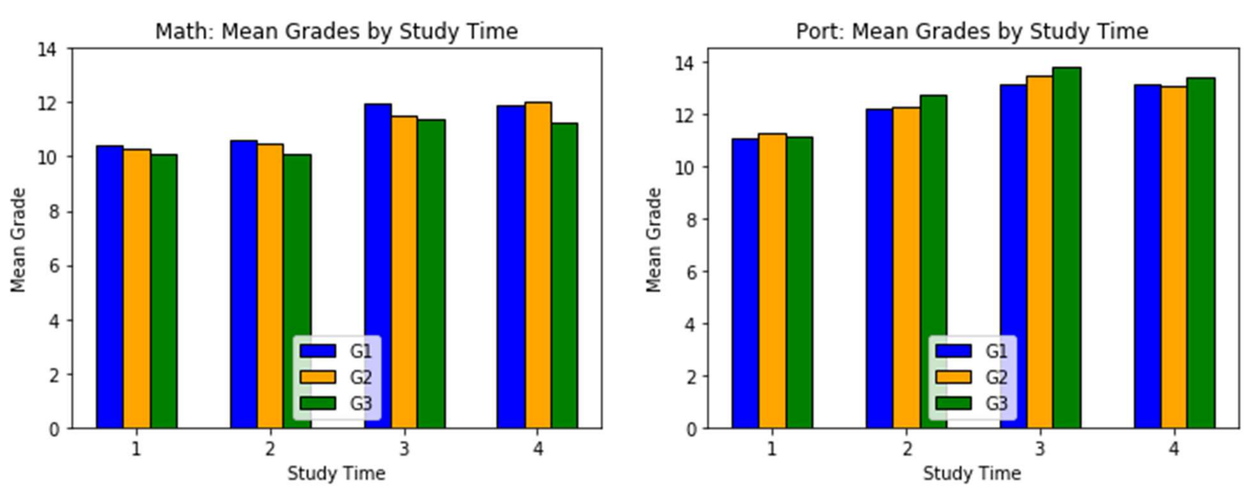


Impact of Study Time and School on Grades

Further exploring the impact of academic factors on final grades, we conducted tests involving a closer look at the variables of study time and school. Figure 9 compares mean G1, G2, and G3 grades in each study time category. We observed that as study time increases, the overall mean grades tend to slightly increase as well. To test the significance of this claim, we conducted a significance test to analyze the impact of study time on mean grades.

Figure 9

Grouped Bar Charts of Mean Grades Per Study Time Group in the Math and Port Data Set



To start, we assumed the independent features were not correlated with each other. The correlation matrices, for both data sets, were found to not have multicollinearity. The null hypothesis was that there is no correlation between study time and G1, G2, or G3 grades. We implemented a chi-square test to determine if there is a relationship between study time and mean grades. Results of the chi-square test, shown in Table 4, show no significant p-values for study time in the Math class, at the 0.05 level. This meant there was insufficient evidence to reject the null hypothesis for the Math class results, and that no correlation between study time and grades was found for the Math class. The Port class results showed significant p-values for G1, G2, and G3, so there was sufficient evidence to reject the null hypothesis and conclude that there is a correlation between study time and grades for the Port class. This suggested that for the Port class, increasing study time is more likely to increase your grades.

Table 4

Results of Chi-Squared Test on Study Times vs Mean Grades

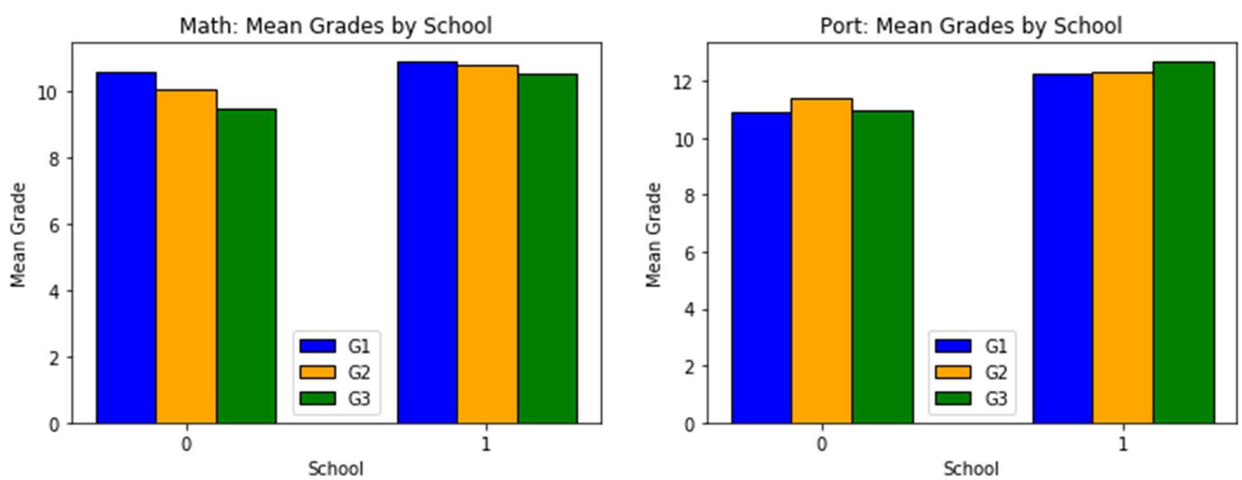
Subject	**Grade**	**p-value**
Math	G1	0.08
Math	G2	0.30
Math	G3	0.20
Port	G1	0.00*
Port	G2	0.00*
Port	G3	0.01*

*p < .05

Next, we investigated whether there is any correlation in the student's school and mean grades per subject. Viewing mean grades by school and subject in Figure 10, we observed that overall school 0 tends to have lower mean grades in both subjects. The difference in mean grades between schools seemed to be larger for the subject Port than for the subject Math. To test the significance of these differences we conducted an independent t-test comparing mean grades across school per subject. Due to unequal sample sizes, we did not assume equal variance.

Figure 10

Grouped Bar Charts for Mean Grades per School for Both the Math and Port Data Sets



Results of the independent t-test are depicted in Table 5. At a significance level of 0.05, there was no significant difference in mean grades across schools for the subject Math. Since no significant p-values were reported for the Math class, there is insufficient evidence to reject the null hypothesis that there is no correlation between school and mean grades in the subject Math. For the subject Port, there was a significant difference between schools in mean G1 grades, no significant difference in mean G2 grades, and a significant difference in mean G3 grades. This would suggest that school 1 has a significantly higher student performance than school 0 in the subject Port. For final grades G3 in the Port class, there is sufficient evidence to reject the null hypothesis and conclude that given which school a student attends can likely inform if their final grade is higher or lower.

Table 5

Results of Independent T-Test on School vs Mean Grades

Subject	**School 0 Mean Grade**	**School 1 Mean Grade**	**Grade**	**p-value**
Math	10.90	10.55	G1	0.55
Math	10.79	10.05	G2	0.22
Math	10.49	9.47	G3	0.20
Port	10.88	12.26	G1	0.02*
Port	11.40	12.34	G2	0.08
Port	10.95	12.70	G3	0.03*

*p < .05

Model Selection

Social Factors Model Selection

To analyze the effects of the selected social features on G3 for subjects Math and Port, we implemented the Ordinary Least Squares (OLS) regression model from the statsmodels Python library. OLS is a type of linear regression that estimates the relationship between a dependent variable and one or more independent variables by minimizing the sum of the squared differences between the observed and predicted values (Agresti & Kateri, 2021).

Demographic Factors Model Selection

A linear regression using OLS was chosen to model the relationship between demographic features and G3. The number of available features, especially after encoding the available features, became a concern. Our goal was to avoid extra “noise” or added complication to our model. To reduce the number of features used in our linear regression model, a technique called Recursive Feature Elimination (RFE) was implemented. RFE evaluates the linear regression model multiple times but removes the least significant feature in every iteration. The RFE algorithm will continue to remove features until a few features you specified is reached. Once the most significant features were chosen, the linear regression model was trained to only use those features to predict a student’s final grade. To determine the number of features to train the model on, various trials of the linear regression model were conducted. For each iteration, the resulting f-statistic guided our decision in determining the number of features to use. Two features were used in predicting the final grade for Math students, and three features were used in determining the final grade for Port students.

Academic Factors Model Selection

To analyze the effects of the selected academic features on the final grade G3 for subjects Math and Port, we performed a multiple regression analysis on both data sets by fitting an OLS regression model using the scikit-learn Python library (Buitinck et al., 2013). We split each data set into a training and test set, with a test size of 20%. We also use the statsmodels

OLS model to find the p-values for each independent variable, given G3 as the dependent variable.

Model Analysis

In the following sections, we report results of model performance from OLS regression models applied to each independent feature category: social, demographic, and academic. These models are applied to both the Math and Port data sets, using G3 as the target variable. We then report R-squared values, coefficients, and p-values to determine which independent features were found to be significant toward predicting final grades.

Social Factors Model Analysis

For the Math Data Set

Model performance on both data sets is listed in Table 6. An R-squared value of 0.056 indicated approximately 5.6% of the variance in the dependent variable G3 can be explained by the independent variables in our model. A low R-squared value like 0.056 suggested the model had weak explanatory power. The independent variables did not explain a large proportion of the variation in G3. The adjusted R-squared (0.041) was even lower than the R-squared, indicating the model might be suffering from overfitting. The inclusion of additional independent variables might not be statistically significant and might be inflating the R-squared value without a true improvement in explanatory power. The p-value associated with the F-statistic was highly significant (less than 0.05). It indicated the observed F-statistic value is unlikely to have occurred by chance, and we can reject the null hypothesis. Based on the coefficients, except for romantic and gout, which was inversely related to the grades, all other features selected had a positive relationship with G3.

Table 6

Model Performance of Social Factors on Fitting Final Grade G3

Performance Metrics	Math Data	Port Data
R-squared	0.056	0.064
Adjusted R-squared	0.041	0.056
F-statistic	3.809	7.352
p-value	0.001*	0.000*

*p < .05

For the Port Data Set

An R-squared value of 0.064 indicated approximately 6.4% of the variance in the dependent variable can be explained by independent variables in our model. A low R-squared value like 0.064 suggested the model has a weak explanatory power. The adjusted R-squared (0.056) was even lower than the R-squared, indicating that the model might be suffering from overfitting. The p-value associated with the F-statistic was highly significant (less than 0.05). It indicated the observed F-statistic value is unlikely to have occurred by chance, and we can reject the null hypothesis. While the F-statistic for both the Math and Port student performance data set suggested a statistically significant relationship between at least one independent variable and the dependent variable, the low R-squared and adjusted R-squared values indicated a weak model fit. The model explained a small portion of the variance, and there might be overfitting due to the inclusion of unnecessary variables. Feature engineering was used to drop less significant features depending on the p-values of the features from the model which did not improve performance. Even though the F-statistic kept increasing after feature selection, the low R-squared and adjusted R-squared values indicated a weak model fit. With no multicollinearity between features selected, the model performance did not improve much with feature selection with respect to R-squared or adjusted R-squared. Table 7 shows the significance of the features selected on the independent variable.

Table 7

Model Coefficients and P-values of Social Factors on Fitting Final Grade G3

	Math Data		**Port Data**	
	Coefficients	p-value	Coefficients	p-value
internet	1.389	0.024*	1.277	0.000*
romantic	-1.396	0.004*	-0.559	0.030*
famrel	0.227	0.377	0.199	0.131
freetime	0.209	0.382	-0.368	0.004*
gout	-0.652	0.002*	-0.164	0.147
absences	0.032	0.263	-0.061	0.024

$^{*}p < .05$

The p-values for the Math data set suggested that only internet, romantic, and gout had statistical significance. The variables romantic and gout were shown to be inversely proportional, and internet was directly correlated. The p-values for the Port dataset showed that internet, romantic, freetime, and absences had statistical significance. It was interesting to find that only internet was directly proportional and the rest of the significant features like romantic, freetime, and absences were inversely proportional. Internet access at home positively impacted students' final grades, while being in a romantic relationship had a negative effect. Based on these findings, schools and the education system in general may want to consider providing home internet services and raising awareness about the impact of romantic relationships.

Demographic Factors Model Analysis

For the Math Data Set

Model performance on both data sets is listed in Table 8. Table 9 shows the significance of the features selected on the independent variable. Using Recursive Feature Elimination (RFE), all the explanatory features besides “Fedu_1” and “Fedu_2” were eliminated. Through an iterative process, reducing the data set’s dimensionality to two features produced the largest F-statistic, suggesting that the model is a better predictor of final grades than just the intercept itself. “Fedu_1,” representing a student’s father whose highest education was a Primary education, was found to be statistically significant. From a logical point of view, it is plausible that the father’s education influences a student’s success in school. Perhaps a father who has not completed higher education values work more than continuing an education, thus not being as supportive of the student’s education. A father’s education may also influence the final grades of a student because the student does not have as easy access to educational support. However, this model is not a strong predictor of a student’s final grade since only 2.5% (R-Squared) of the variance is explained by the chosen features.

For the Port Data Set

Using Recursive Feature Elimination (RFE), all the explanatory features besides “sex_M”, “Medu_4, and “Fedu_1” were eliminated. The large F-Statistic suggests the model is highly significant compared to just using the intercept, but like the regression model representing Math students, this model is not a strong predictor of a student's final grade. However, even though very little of the variance is described by these three explanatory features, education can be heavily affected by a student's support group, such as their parents, their parent's education level, and even their sex because of cultural norms.

Table 8

Model Performance of Demographic Factors on Fitting Final Grade G3

Performance Metrics	Math Data	Port Data
R-squared	0.025	0.087
Adjusted R-squared	0.020	0.083
F-statistic	5.005	20.55
p-value	0.007*	0.000*

*p < .05

Table 9

Model Coefficients and P-values of Demographic Factors on Fitting Final Grade G3

	Math Data		Port Data	
	Coefficients	p-value	Coefficients	p-value
Fedu_1	-1.867	0.002*	-0.978	0.001*
Fedu_2	-0.764	0.151		
sex_M			-1.040	0.000*
Medu_4			1.410	0.000*

*p < .05

Academic Factors Model Analysis

The result of the multiple linear regression is depicted in Table 10. The model trained on the Math data set had an R-squared value of 0.20, which means that approximately 20% of the variance in the model was explained by the selected independent features. Only two independent variables, failures and higher, were statistically significant at the 0.05 level for the Math data set. The failures variable had a significant negative relationship between final grades while the variable higher had a significant positive relationship with final grades. For these two variables, this was also the case in the Port data set, with the same respective significant relationships.

The model trained on the Port data set had an R-squared value of 0.23, which means that approximately 23% of the variance in the model was explained by the selected independent features. This R-squared value was slightly higher than the Math model R-squared model. For the Port data set, all independent variables except famsup and paid showed statistical significance at the 0.05 level. It was interesting to see that the variable schoolsup had a negative relationship with final grades. This sounds counterintuitive as one would assume that more educational support would improve final grades. The extra educational support provided at these schools would maybe need further review of whether they improve student's grades or not. The variable for school also showed significance as a predictor for final grades in the Port class. This meant that school is a more significant predictor when estimating mean final grades for the subject Port than when estimating mean grades for the subject Math. Given that these schools are in Portugal, it was interesting to see more significant findings with the Port data set. Students taking the Portuguese language class may be more likely to get higher final grades compared to the Math class since they live in that Portuguese culture and environment. They may speak the language everyday with family and support systems, so they may be getting more practice from sources outside the classroom which can further impact final grades.

Table 10

Model Coefficients and P-values of Academic Factors on Fitting Final Grade G3

	Math Data (R-squared: 0.20)		**Port Data (R-squared: 0.23)**	
	Coefficients	p-value	Coefficients	p-value
studytime	0.03	0.87	0.80	0.00*
failures	-2.00	0.00*	-1.04	0.00*
schoolsup	-1.01	0.13	-1.39	0.00*
famsup	-0.89	0.09	0.31	0.47
paid	0.51	0.51	-0.95	0.05
higher	2.82	0.03*	2.47	0.00*
school	0.69	0.07	1.39	0.00*

*p < .05

Conclusion and Recommendations

We conducted statistical analysis on student performance from social, demographic, and academic factors, in the two subjects Math and Portuguese language. Factors that had a significant positive correlation with final grades in both subjects included having internet at home, father’s education, and want for higher education. Factors that had a significant negative correlation with final grades in both subjects included romantic relationships, and number of past failures. Some recommendations to improve model performance include to collect more data, select additional features, use interaction terms to capture more complex relationships, and to investigate alternative models or techniques like decision trees or random forests that do not require linear assumptions.

References

Agresti, A., & Kateri, M. (2021). *Foundations of statistics for data scientists: With R and Python* (1st ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9781003159834

Malinsky, A, Jayapal, M, & Hogan, S. (2024). *Student Performance Analysis Notebooks* [Computer software]. https://github.com/apmalinsky/AAI-500-Final-Project

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). *API design for machine learning software: Experiences from the scikit-learn project*. https://doi.org/10.48550/ARXIV.1309.0238

Cortez, P. (2008). *Student performance* [Data set]. UCI Machine Learning Repository. https://doi.org/10.24432/C5TG7T

Appendix A

Impact of Social Factors on Final Grades

Name: Maha Jayapal

Date: 6/24/2024

In [1]:
```
#Importing libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.formula.api as smf
from sklearn import linear_model
from scipy.stats import pearsonr
from sklearn.linear_model import LinearRegression
```

In [2]:
```
file_path = '../dataset/student-por.csv'
student_por = pd.read_csv(file_path,  delimiter=';')

# Display the first few rows of the dataset
student_por.head()
```

Out[2]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freet
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	

5 rows × 33 columns

In [3]:
```python
file_path = '../dataset/student-mat.csv'
student_math = pd.read_csv(file_path,  delimiter=';')

# Display the first few rows of the dataset
student_math.head()
```

Out[3]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freet
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	

5 rows × 33 columns

In [4]:
```python
print(student_math.shape)
student_por.shape
```

```
(395, 33)
```

Out[4]:
```
(649, 33)
```

There are 395 observations in the Math dataset, and 649 observations in the portuguese dataset.

Features used to analyze the grade impact based on social life:

internet: Binary, Internet access at home (binary: yes or no)

romantic: Binary, with a romantic relationship (binary: yes or no)

famrel: Integer, quality of family relationships (numeric: from 1 - very bad to 5 - excellent)

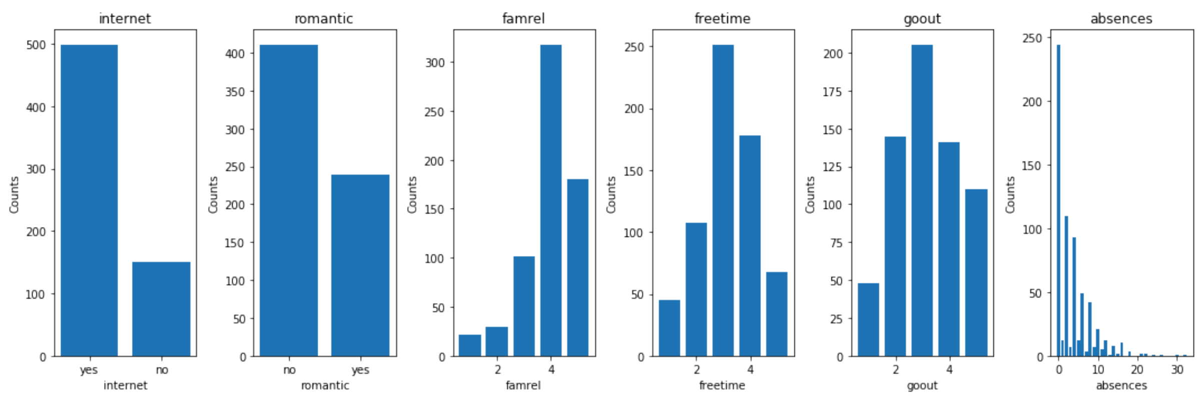
freetime: Integer, free time after school (numeric: from 1 - very low to 5 - very high)

goout: Integer, going out with friends (numeric: from 1 - very low to 5 - very high)

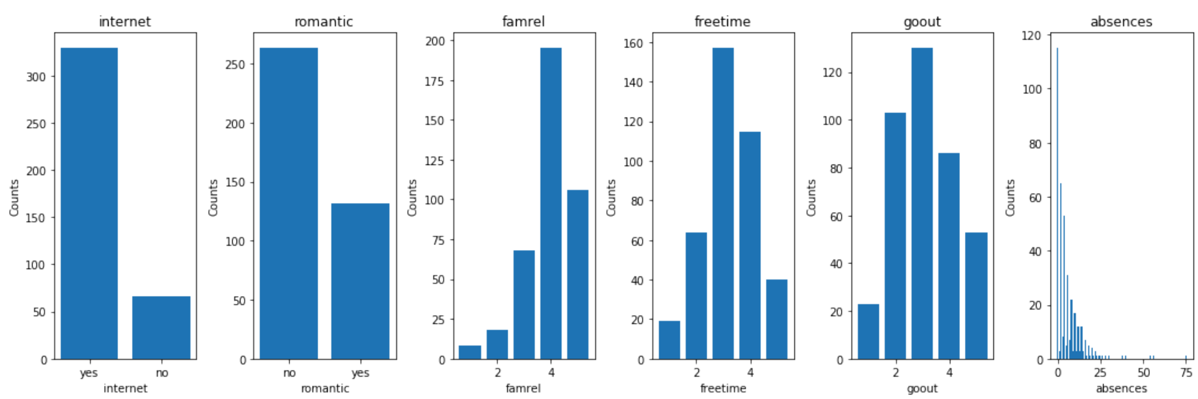
absences: Integer, number of school absences (numeric: from 0 to 93)

In [5]:
```python
selected_columns = ['internet' ,'romantic', 'famrel' , 'freetime' , 'goout'
,'absences']

subset_stu_por = student_por[selected_columns]
subset_stu_math = student_math[selected_columns]
```

In [6]:
```python
def draw_barplots(df, num_columns):
    # Get the specified number of columns
    columns = df.columns[:num_columns]

    # Create subplots with the specified number of columns
    fig, axes = plt.subplots(1, num_columns, figsize=(15, 5))

    # Iterate through the columns and draw bar plots
    for i, col in enumerate(columns):
        ax = axes[i] if num_columns > 1 else axes
        value_counts = df[col].value_counts()
        ax.bar(value_counts.index, value_counts.values)
        ax.set_title(col)
        ax.set_xlabel(col)
        ax.set_ylabel('Counts')

    plt.tight_layout()
    plt.show()


draw_barplots(subset_stu_por, 6)
```



In [7]:
```python
draw_barplots(subset_stu_math, 6)
```



The bar plots for the selected features revealed several insights. For both the datasets, more number of students have internet access at home, fewer students were in romantic relationships, and family relationships were left-skewed, indicating mostly positive family dynamics. Free time and going out showed roughly normal distributions, while absences were heavily right-skewed with mostly low values and few high values.

In [8]:
```
print(student_math[selected_columns].dtypes)
print(student_por[selected_columns].dtypes)
```

```
internet    object
romantic    object
famrel       int64
freetime     int64
goout        int64
absences     int64
dtype: object
internet    object
romantic    object
famrel       int64
freetime     int64
goout        int64
absences     int64
dtype: object
```

The feature

In [9]:
```
X_math = student_math[selected_columns]
X_math.head(5)
```

Out[9]:

	internet	romantic	famrel	freetime	goout	absences
0	no	no	4	3	4	6
1	yes	no	5	3	3	4
2	yes	no	4	3	2	10
3	yes	yes	3	2	2	2
4	no	no	4	3	2	4

Among the six independent features selected, internet and romantic are binary, famrel which is the quality of family relationships, freetime which is the free time after school, goout is going out with friends which are all ordinal numeric variables ranging from 1 which is very bad or low to 5 which is excellent or very high. The variable absences is a numeric variable ranging from 0 to 93 which is the number of school absences.

In [10]:
```
# Create a dictionary mapping 'yes' to 1 and 'no' to 0
mapping = {'yes': 1, 'no': 0}

# Replace values in the 'answer' column using the dictionary
student_math['internet'] = student_math['internet'].replace(mapping)
student_math['romantic'] = student_math['romantic'].replace(mapping)

student_por['internet'] = student_por['internet'].replace(mapping)
student_por['romantic'] = student_por['romantic'].replace(mapping)
```

The target variable G3 is an integer. Independent variables famrel, freetime, goout, and absences were numeric, while internet and romantic were categorical with yes and no. These were converted to 1 and 0 for modeling.

In [11]:
```
X_math = student_math[selected_columns]
X_math.head(5)
```

Out[11]:

	internet	romantic	famrel	freetime	goout	absences
0	0	0	4	3	4	6
1	1	0	5	3	3	4
2	1	0	4	3	2	10
3	1	1	3	2	2	2
4	0	0	4	3	2	4

In [12]:
```
X_math.dtypes
```

Out[12]:
```
internet     int64
romantic     int64
famrel       int64
freetime     int64
goout        int64
absences     int64
dtype: object
```

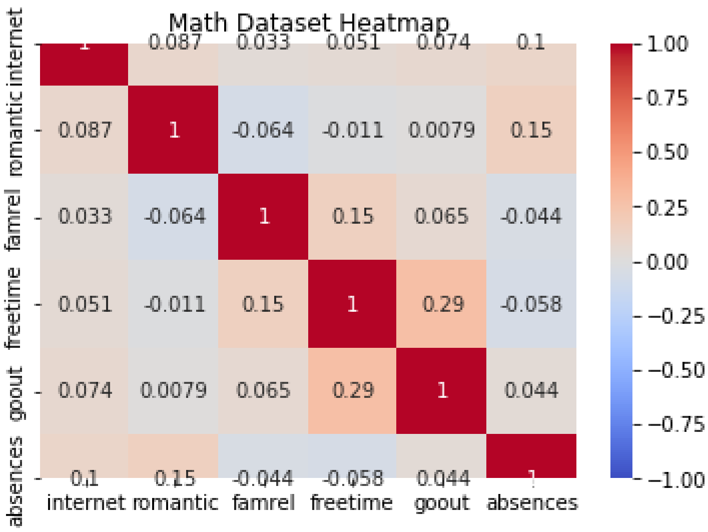
In [13]:
```
correlation_matrix = X_math.corr()
correlation_matrix
```

Out[13]:

	internet	romantic	famrel	freetime	goout	absences
internet	1.000000	0.087122	0.032768	0.051286	0.074370	0.101701
romantic	0.087122	1.000000	-0.063816	-0.011182	0.007870	0.153384
famrel	0.032768	-0.063816	1.000000	0.150701	0.064568	-0.044354
freetime	0.051286	-0.011182	0.150701	1.000000	0.285019	-0.058078
goout	0.074370	0.007870	0.064568	0.285019	1.000000	0.044302
absences	0.101701	0.153384	-0.044354	-0.058078	0.044302	1.000000

The correlation matrix shows that the features selected does not have multicollinearity.

In [14]:
```
# Create the heat map
plt.figure(figsize=(6, 4))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)

# Add a title
plt.title('Math Dataset Heatmap')

# Display the heat map
plt.show()
```



In [15]:
```
X_por = student_por[selected_columns]
X_por.head(5)
```

Out[15]:

	internet	romantic	famrel	freetime	goout	absences
0	0	0	4	3	4	4
1	1	0	5	3	3	2
2	1	0	4	3	2	6
3	1	1	3	2	2	0
4	0	0	4	3	2	0

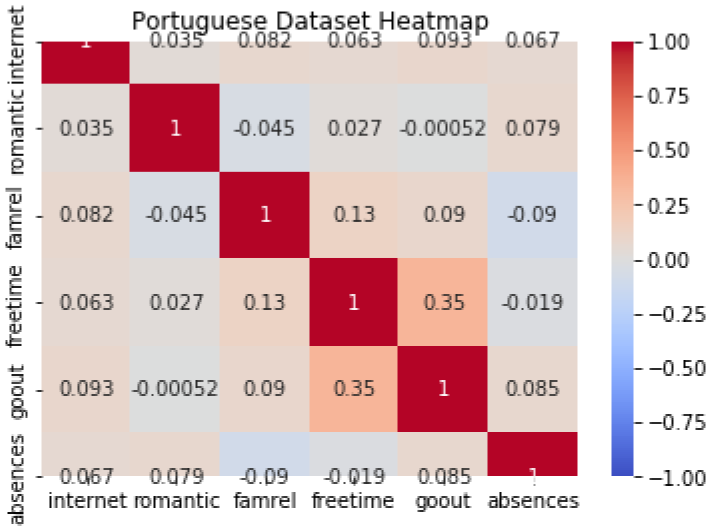
In [16]:
```
correlation_matrix = X_por.corr()
correlation_matrix
```

Out[16]:

	internet	romantic	famrel	freetime	goout	absences
internet	1.000000	0.034832	0.082214	0.063268	0.092869	0.067301
romantic	0.034832	1.000000	-0.044920	0.027112	-0.000520	0.079489
famrel	0.082214	-0.044920	1.000000	0.129216	0.089707	-0.089534
freetime	0.063268	0.027112	0.129216	1.000000	0.346352	-0.018716
goout	0.092869	-0.000520	0.089707	0.346352	1.000000	0.085374
absences	0.067301	0.079489	-0.089534	-0.018716	0.085374	1.000000

The correlation matrix shows that the features selected does not have multicollinearity.

In [17]:
```python
# Create the heat map
plt.figure(figsize=(6, 4))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)

# Add a title
plt.title('Portuguese Dataset Heatmap')

# Display the heat map
plt.show()
```



In [18]:
```
X_math = student_math[selected_columns]
y_math = student_math["G3"]

X_math = sm.add_constant(X_math)

model = sm.OLS(y_math, X_math)
results = model.fit()

params = results.params
params
results.summary()
```

Out[18]:

OLS Regression Results

Dep. Variable:	G3	**R-squared:**	0.056
Model:	OLS	**Adj. R-squared:**	0.041
Method:	Least Squares	**F-statistic:**	3.809
Date:	Mon, 24 Jun 2024	**Prob (F-statistic):**	0.00106
Time:	20:09:09	**Log-Likelihood:**	-1149.9
No. Observations:	395	**AIC:**	2314.
Df Residuals:	388	**BIC:**	2342.
Df Model:	6		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	9.9966	1.339	7.466	0.000	7.364	12.629
internet	1.3893	0.612	2.269	0.024	0.186	2.593
romantic	-1.3960	0.486	-2.870	0.004	-2.352	-0.439
famrel	0.2265	0.256	0.885	0.377	-0.276	0.729
freetime	0.2094	0.239	0.876	0.382	-0.261	0.679
goout	-0.6521	0.213	-3.067	0.002	-1.070	-0.234
absences	0.0323	0.029	1.121	0.263	-0.024	0.089

Omnibus:	26.878	**Durbin-Watson:**	2.050
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	30.518
Skew:	-0.671	**Prob(JB):**	2.36e-07
Kurtosis:	3.236	**Cond. No.**	64.4

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

The R squared value of 0.056 indicates that approximately 5.6% of the variance in the dependent variable can be explained by the independent variables included in your model. A low R-squared value like 0.056 suggests that the model has a weak explanatory power. The independent variables don't explain a large proportion of the variation in the dependent variable.

The adjusted R-squared (0.041) is even lower than the R-squared, indicating that the model might be suffering from overfitting. The inclusion of additional independent variables might not be statistically significant and might be inflating the R-squared value without a true improvement in explanatory power.

The p-value associated with the F-statistic is highly significant (less than 0.05). It indicates that the observed F-statistic value is unlikely to have occurred by chance, and we can reject the null hypothesis.

Based on the coefficients except for the romantic and goout which is inversely related to the grades, all the other features selected has a positive relationship with the dependent variable.

In [19]:
```
X_por = student_por[selected_columns]
y_por = student_por["G3"]

X_por = sm.add_constant(X_por)

model = sm.OLS(y_por, X_por)
results = model.fit()

results.summary()
```

Out[19]:

OLS Regression Results

Dep. Variable:	G3	**R-squared:**	0.064
Model:	OLS	**Adj. R-squared:**	0.056
Method:	Least Squares	**F-statistic:**	7.352
Date:	Mon, 24 Jun 2024	**Prob (F-statistic):**	1.28e-07
Time:	20:09:09	**Log-Likelihood:**	-1659.9
No. Observations:	649	**AIC:**	3334.
Df Residuals:	642	**BIC:**	3365.
Df Model:	6		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	12.2653	0.673	18.227	0.000	10.944	13.587
internet	1.2768	0.295	4.332	0.000	0.698	1.855
romantic	-0.5593	0.257	-2.178	0.030	-1.064	-0.055
famrel	0.1988	0.131	1.513	0.131	-0.059	0.457
freetime	-0.3677	0.126	-2.918	0.004	-0.615	-0.120
goout	-0.1636	0.113	-1.450	0.147	-0.385	0.058
absences	-0.0612	0.027	-2.269	0.024	-0.114	-0.008

Omnibus:	123.806	**Durbin-Watson:**	1.638
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	336.959
Skew:	-0.951	**Prob(JB):**	6.76e-74
Kurtosis:	5.973	**Cond. No.**	42.8

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

The R squared value of 0.064 indicates that approximately 6.4% of the variance in the dependent variable can be explained by the independent variables included in your model. A low R-squared value like 0.064 suggests that the model has a weak explanatory power.

The adjusted R-squared (0.056) is even lower than the R-squared, indicating that the model might be suffering from overfitting.

The p-value associated with the F-statistic is highly significant (less than 0.05). It indicates that the observed F-statistic value is unlikely to have occurred by chance, and we can reject the null hypothesis.

While the F-statistic for both the math and portuguese student performance dataset suggests a statistically significant relationship between at least one independent variable and the dependent variable, the low R-squared and adjusted R-squared values indicate a weak model fit. The model explains a small portion of the variance, and there might be overfitting due to the inclusion of unnecessary variables.

Feature selection to improve model performance:

Dropping the features famrel in both the datasets which has a higher p value.

In [20]:
```
selected_columns = ['internet' ,'romantic' , 'freetime' , 'goout' ,'absences']
```

In [21]:

```
X_math = student_math[selected_columns]
y_math = student_math["G3"]

X_math = sm.add_constant(X_math)

model = sm.OLS(y_math, X_math)
results = model.fit()

results.summary()
```

Out[21]:

OLS Regression Results

Dep. Variable:	G3	**R-squared:**	0.054
Model:	OLS	**Adj. R-squared:**	0.042
Method:	Least Squares	**F-statistic:**	4.417
Date:	Mon, 24 Jun 2024	**Prob (F-statistic):**	0.000634
Time:	20:09:09	**Log-Likelihood:**	-1150.3
No. Observations:	395	**AIC:**	2313.
Df Residuals:	389	**BIC:**	2336.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	10.7829	1.002	10.763	0.000	8.813	12.753
internet	1.4069	0.612	2.300	0.022	0.204	2.610
romantic	-1.4219	0.485	-2.929	0.004	-2.376	-0.467
freetime	0.2378	0.237	1.004	0.316	-0.228	0.703
goout	-0.6477	0.213	-3.048	0.002	-1.066	-0.230
absences	0.0315	0.029	1.095	0.274	-0.025	0.088

Omnibus:	27.551	**Durbin-Watson:**	2.049
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	31.401
Skew:	-0.679	**Prob(JB):**	1.52e-07
Kurtosis:	3.251	**Cond. No.**	48.0

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [22]:
```
X_por = student_por[selected_columns]
y_por = student_por["G3"]

X_por = sm.add_constant(X_por)

model = sm.OLS(y_por, X_por)
results = model.fit()

results.summary()
```

Out[22]:

OLS Regression Results

Dep. Variable:	G3	**R-squared:**	0.061
Model:	OLS	**Adj. R-squared:**	0.054
Method:	Least Squares	**F-statistic:**	8.348
Date:	Mon, 24 Jun 2024	**Prob (F-statistic):**	1.16e-07
Time:	20:09:09	**Log-Likelihood:**	-1661.1
No. Observations:	649	**AIC:**	3334.
Df Residuals:	643	**BIC:**	3361.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	12.9517	0.497	26.037	0.000	11.975	13.929
internet	1.3120	0.294	4.461	0.000	0.734	1.889
romantic	-0.5762	0.257	-2.243	0.025	-1.081	-0.072
freetime	-0.3487	0.126	-2.778	0.006	-0.595	-0.102
goout	-0.1549	0.113	-1.373	0.170	-0.376	0.067
absences	-0.0650	0.027	-2.420	0.016	-0.118	-0.012

Omnibus:	123.434	**Durbin-Watson:**	1.635
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	335.248
Skew:	-0.949	**Prob(JB):**	1.59e-73
Kurtosis:	5.965	**Cond. No.**	29.1

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Freetime and absences have higher p values for the math dataset, goout has higher p value for the portuguese dataset. So dropping those features and checking the model performance.

In [23]:

```
selected_columns = ['internet' ,'romantic' , 'goout']
X_math = student_math[selected_columns]
y_math = student_math["G3"]

X_math = sm.add_constant(X_math)

model = sm.OLS(y_math, X_math)
results = model.fit()

results.summary()
```

Out[23]:

OLS Regression Results

Dep. Variable:	G3	**R-squared:**	0.049
Model:	OLS	**Adj. R-squared:**	0.041
Method:	Least Squares	**F-statistic:**	6.676
Date:	Mon, 24 Jun 2024	**Prob (F-statistic):**	0.000210
Time:	20:09:10	**Log-Likelihood:**	-1151.3
No. Observations:	395	**AIC:**	2311.
Df Residuals:	391	**BIC:**	2327.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	11.4301	0.820	13.938	0.000	9.818	13.042
internet	1.4852	0.609	2.439	0.015	0.288	2.682
romantic	-1.3523	0.480	-2.816	0.005	-2.297	-0.408
goout	-0.5790	0.204	-2.845	0.005	-0.979	-0.179

Omnibus:	32.489	**Durbin-Watson:**	2.040
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	38.096
Skew:	-0.736	**Prob(JB):**	5.34e-09
Kurtosis:	3.387	**Cond. No.**	14.4

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [24]:
```
selected_columns = ['internet' ,'romantic' , 'freetime' , 'absences']
X_por = student_por[selected_columns]
y_por = student_por["G3"]

X_por = sm.add_constant(X_por)

model = sm.OLS(y_por, X_por)
results = model.fit()

results.summary()
```

Out[24]:

OLS Regression Results

Dep. Variable:	G3	**R-squared:**	0.058
Model:	OLS	**Adj. R-squared:**	0.052
Method:	Least Squares	**F-statistic:**	9.950
Date:	Mon, 24 Jun 2024	**Prob (F-statistic):**	8.12e-08
Time:	20:09:10	**Log-Likelihood:**	-1662.0
No. Observations:	649	**AIC:**	3334.
Df Residuals:	644	**BIC:**	3356.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	12.6801	0.457	27.763	0.000	11.783	13.577
internet	1.2837	0.294	4.373	0.000	0.707	1.860
romantic	-0.5690	0.257	-2.214	0.027	-1.074	-0.064
freetime	-0.4084	0.118	-3.465	0.001	-0.640	-0.177
absences	-0.0685	0.027	-2.559	0.011	-0.121	-0.016

Omnibus:	120.536	**Durbin-Watson:**	1.641
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	322.008
Skew:	-0.933	**Prob(JB):**	1.19e-70
Kurtosis:	5.902	**Cond. No.**	25.2

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Even though the F statistic kept increasing after feature selection, the low R-squared and adjusted R-squared values indicate a weak model fit. With no multicollinearity between features selected, the model performance did not improve much with feature selection with respect to R squared or adjusted R squared.

Appendix B

Impact of Demographic Factors on Final Grades

Name: Scott Hogan

Date: 6/24/2024

In [1]:
```
import pandas as pd
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
import matplotlib.pyplot as plt
from matplotlib import gridspec

math = pd.read_csv('../dataset/student-mat.csv', delimiter=';')
language = pd.read_csv('../dataset/student-por.csv', delimiter=';')
```

Tasks:

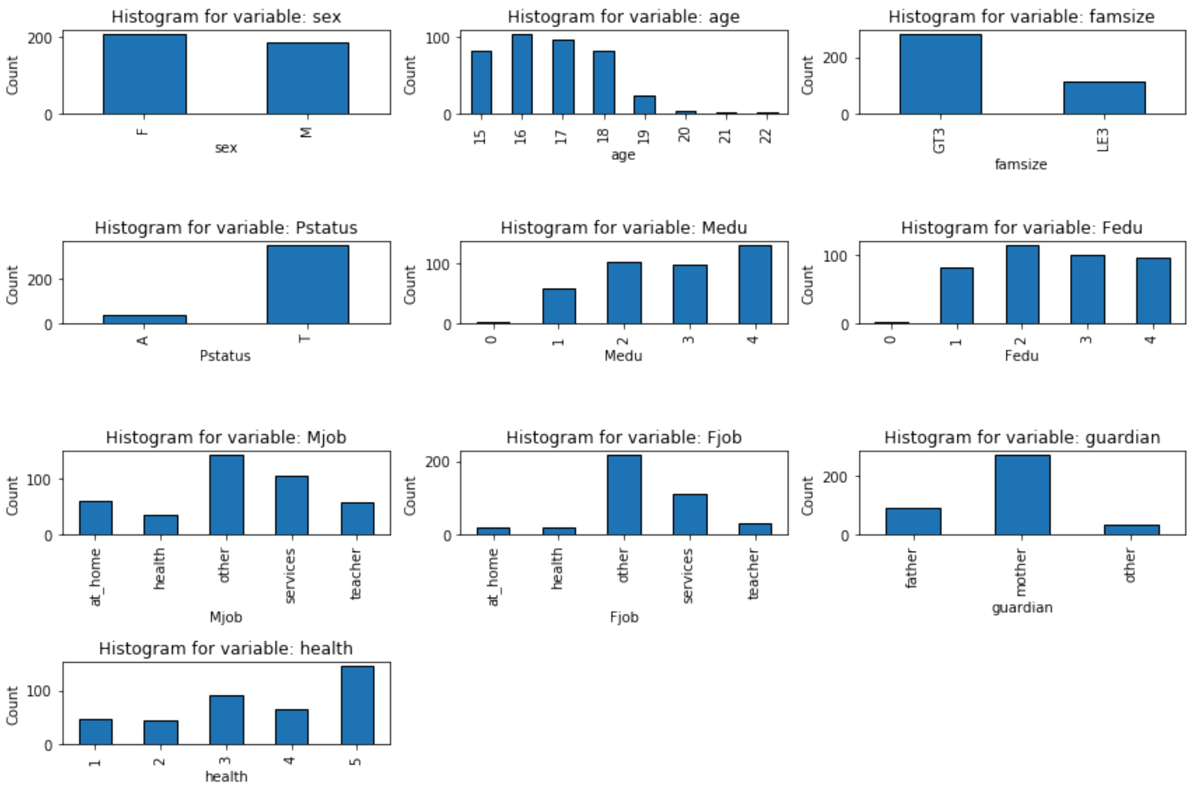
Investigate impact of demographic factors on final grades

Compare across subjects

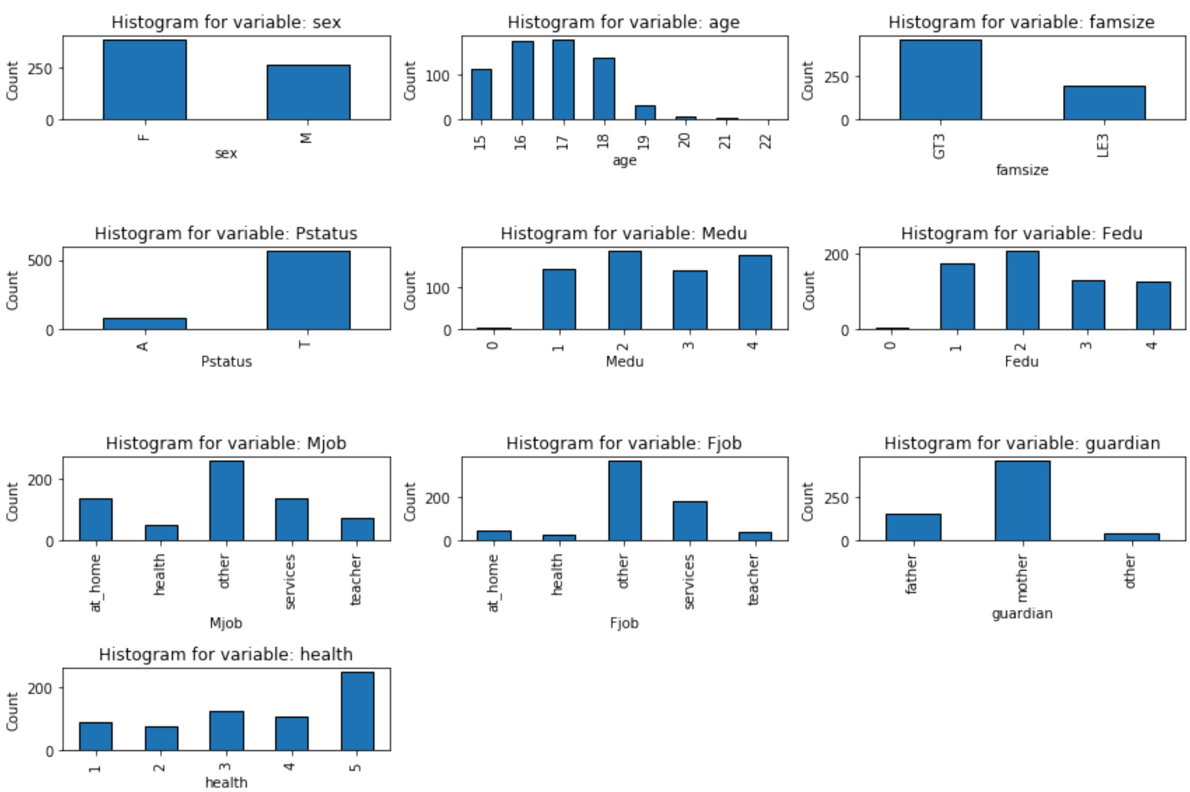
In [2]:
```python
# Drop non-demographic columns
demo_math = math.drop(['school', 'address', 'reason', 'traveltime', 'studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'absences', 'G1', 'G2', 'G3', 'romantic'], axis=1)
demo_language = language.drop(['school', 'address', 'reason', 'traveltime', 'studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'absences', 'G1', 'G2', 'G3', 'romantic'], axis=1)

# Change 'Medu', 'Fedu', and 'health' columns into strings for label encoding
convert_type = {'Medu': str, 'Fedu': str, 'health': str}

demo_math = demo_math.astype(convert_type)
demo_language = demo_language.astype(convert_type)



# Encode labels
demo_math_dummies = pd.get_dummies(demo_math, drop_first=True, dtype=int)
demo_language_dummies = pd.get_dummies(demo_language, drop_first=True)



# G3 Response Variable
G3_math = math['G3']
G3_lang = language['G3']
```

In [3]:
```
columns = demo_math.columns  # independent variables
n_plots = len(columns)
# Plot frequency distributions for each independent variable
def plot_variables(table):
    gs = gridspec.GridSpec(4, 3)
    fig = plt.figure(figsize=(12,8))
    for i in range(n_plots):
        ax = fig.add_subplot(gs[i])
        table[columns[i]].value_counts().sort_index().plot(kind='bar', ax=ax,
edgecolor='black')
        ax.set_xlabel(columns[i])
        ax.set_ylabel('Count')
        ax.set_title('Histogram for variable: ' + columns[i])
    fig.tight_layout()
    plt.show()
print('MAT STUDENTS')
plot_variables(demo_math)
print('POR STUDENTS')
plot_variables(demo_language)
```

MAT STUDENTS



POR STUDENTS



In [4]:
```
# Reduce number of features using Recursive Feature Elimination (RFE) and fit
linear regression for math class at Gabriel Pereira

model = LinearRegression()
rfe = RFE(estimator=model, n_features_to_select=2)
rfe = rfe.fit(demo_math_dummies, G3_math)

# Get the selected features
selected_features = demo_math_dummies.columns[rfe.support_]
print("Selected features:", selected_features)

# Fit the model again with selected features
X_selected = demo_math_dummies[selected_features]
X_selected = sm.add_constant(X_selected)
model_selected = sm.OLS(G3_math, X_selected).fit()
print(model_selected.summary())
```

```
Selected features: Index(['Fedu_1', 'Fedu_2'], dtype='object')
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                     G3   R-squared:                       0.02
5
Model:                            OLS   Adj. R-squared:                  0.02
0
Method:                 Least Squares   F-statistic:                     5.00
5
Date:                Mon, 24 Jun 2024   Prob (F-statistic):            0.0071
4
Time:                        20:08:56   Log-Likelihood:                -1156.
2
No. Observations:                 395   AIC:                             231
8.
Df Residuals:                     392   BIC:                             233
0.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const         11.0253      0.322     34.205      0.000      10.392      11.65
9
Fedu_1        -1.8667      0.596     -3.134      0.002      -3.038      -0.69
6
Fedu_2        -0.7644      0.532     -1.437      0.151      -1.810       0.28
1
==============================================================================
=
Omnibus:                       33.014   Durbin-Watson:                   1.97
6
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               38.83
6
Skew:                          -0.740   Prob(JB):                     3.69e-0
9
Kurtosis:                       3.413   Cond. No.                         3.2
6
==============================================================================
=

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [5]:
```
# Reduce number of features using Recursive Feature Elimination (RFE) and fit
linear regression for langauge class at Gabriel Pereira

model = LinearRegression()
rfe = RFE(estimator=model, n_features_to_select=3)
rfe = rfe.fit(demo_language_dummies, G3_lang)

# Get the selected features
selected_features = demo_language_dummies.columns[rfe.support_]
print("Selected features:", selected_features)

# Fit the model again with selected features
X_selected = demo_language_dummies[selected_features]
X_selected = sm.add_constant(X_selected)
model_selected = sm.OLS(G3_lang, X_selected.astype(int)).fit()
print(model_selected.summary())
```

```
Selected features: Index(['sex_M', 'Medu_4', 'Fedu_1'], dtype='object')
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                     G3   R-squared:                       0.08
7
Model:                            OLS   Adj. R-squared:                  0.08
3
Method:                 Least Squares   F-statistic:                     20.5
5
Date:                Mon, 24 Jun 2024   Prob (F-statistic):           9.98e-1
3
Time:                        20:08:56   Log-Likelihood:               -1651.
8
No. Observations:                 649   AIC:                             331
2.
Df Residuals:                     645   BIC:                             333
0.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const         12.2140      0.200     60.975      0.000      11.821      12.60
7
sex_M         -1.0397      0.248     -4.185      0.000      -1.528      -0.55
2
Medu_4         1.4102      0.288      4.891      0.000       0.844       1.97
6
Fedu_1        -0.9775      0.288     -3.392      0.001      -1.543      -0.41
2
==============================================================================
=
Omnibus:                      136.389   Durbin-Watson:                   1.67
7
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              419.96
9
Skew:                          -1.001   Prob(JB):                     6.38e-9
2
Kurtosis:                       6.394   Cond. No.                         3.4
4
==============================================================================
=

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [6]:
```
print(demo_math.columns)
```

```
Index(['sex', 'age', 'famsize', 'Pstatus', 'Medu', 'Fedu', 'Mjob', 'Fjob',
       'guardian', 'health'],
      dtype='object')
```

Appendix C

Impact of Academic Factors on Final Grades

Name: Andy Malinsky

Date: 6/24/2024

Data Cleaning/Preparation

Independent Variables:

school - student's school (binary: "GP" - Gabriel Pereira or "MS" - Mousinho da Silveira)
studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
failures - number of past class failures (numeric: n if 1<=n<3, else 4)
schoolsup - extra educational support (binary: yes or no)
famsup - family educational support (binary: yes or no)
paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
higher - wants to take higher education (binary: yes or no)

Dependent Variables:

G1 - first period grade (numeric: from 0 to 20)
G2 - second period grade (numeric: from 0 to 20)
G3 - final grade (numeric: from 0 to 20, output target)

In [1]:
```
# Import libraries
import pandas as pd
import math
from matplotlib import gridspec
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy.stats import chi2_contingency
from scipy import stats
import statsmodels.formula.api as smf
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [2]:
```
# Read in the datasets
students_mat = pd.read_csv('../dataset/student-mat.csv', delimiter=';')
students_por = pd.read_csv('../dataset/student-por.csv', delimiter=';')

# Merge common students
mat_students_merged = students_mat.merge(students_por[["school","sex","age","a
ddress","famsize","Pstatus","Medu","Fedu","Mjob","Fjob","reason","nursery","in
ternet"]])
por_students_merged = students_por.merge(students_mat[["school","sex","age","a
ddress","famsize","Pstatus","Medu","Fedu","Mjob","Fjob","reason","nursery","in
ternet"]])

# Filter out for academic features
mat_students_premap = mat_students_merged[['school', 'studytime', 'failures',
'schoolsup', 'famsup', 'paid', 'higher', 'G1', 'G2', 'G3']]
por_students_premap = por_students_merged[['school', 'studytime', 'failures',
'schoolsup', 'famsup', 'paid', 'higher', 'G1', 'G2', 'G3']]

mat_students = mat_students_premap.copy() # create copies of original datafram
e to avoid mapping warnings
por_students = por_students_premap.copy()
```

In [3]:
```
print("mat shape:", mat_students.shape[0]) # shape of datasets
print("por shape:", por_students.shape[0])
```

```
mat shape: 382
por shape: 382
```

In [4]:
```
mat_students.head(3) # View the math dataset
```

Out[4]:

	school	studytime	failures	schoolsup	famsup	paid	higher	G1	G2	G3
0	GP	2	0	yes	no	no	yes	5	6	6
1	GP	2	0	no	yes	no	yes	5	5	6
2	GP	2	3	yes	no	yes	yes	7	8	10

In [5]:
```
por_students.head(3) # View the por dataset
```

Out[5]:

	school	studytime	failures	schoolsup	famsup	paid	higher	G1	G2	G3
0	GP	2	0	yes	no	no	yes	0	11	11
1	GP	2	0	no	yes	no	yes	9	11	11
2	GP	2	0	yes	no	no	yes	12	13	12

In [6]:
```python
# Create a dictionary mapping 'yes' to 1 and 'no' to 0
mapping = {'yes': 1, 'no': 0, 'GP': 1, 'MS': 0}

mat_students['school'] = mat_students_premap['school'].replace(mapping)
por_students['school'] = por_students['school'].replace(mapping)

mat_students['schoolsup'] = mat_students_premap['schoolsup'].replace(mapping)
mat_students['famsup'] = mat_students_premap['famsup'].replace(mapping)
mat_students['paid'] = mat_students_premap['paid'].replace(mapping)
mat_students['higher'] = mat_students_premap['higher'].replace(mapping)
por_students['schoolsup'] = por_students_premap['schoolsup'].replace(mapping)
por_students['famsup'] = por_students_premap['famsup'].replace(mapping)
por_students['paid'] = por_students_premap['paid'].replace(mapping)
por_students['higher'] = por_students_premap['higher'].replace(mapping)
```

In [7]:
```python
mat_students.head(3) # View the por dataset
```

Out[7]:

	school	studytime	failures	schoolsup	famsup	paid	higher	G1	G2	G3
0	1	2	0	1	0	0	1	5	6	6
1	1	2	0	0	1	0	1	5	5	6
2	1	2	3	1	0	1	1	7	8	10

In [8]:
```python
por_students.head(3) # View the por dataset
```

Out[8]:

	school	studytime	failures	schoolsup	famsup	paid	higher	G1	G2	G3
0	1	2	0	1	0	0	1	0	11	11
1	1	2	0	0	1	0	1	9	11	11
2	1	2	0	1	0	0	1	12	13	12

In [9]:
```python
mat_students.describe().applymap('{:,.2f}'.format).T # descriptive statistics for mat class
```

Out[9]:

	count	mean	std	min	25%	50%	75%	max
school	382.00	0.90	0.31	0.00	1.00	1.00	1.00	1.00
studytime	382.00	2.03	0.85	1.00	1.00	2.00	2.00	4.00
failures	382.00	0.29	0.73	0.00	0.00	0.00	0.00	3.00
schoolsup	382.00	0.13	0.34	0.00	0.00	0.00	0.00	1.00
famsup	382.00	0.62	0.49	0.00	0.00	1.00	1.00	1.00
paid	382.00	0.46	0.50	0.00	0.00	0.00	1.00	1.00
higher	382.00	0.95	0.21	0.00	1.00	1.00	1.00	1.00
G1	382.00	10.86	3.35	3.00	8.00	10.50	13.00	19.00
G2	382.00	10.71	3.83	0.00	8.25	11.00	13.00	19.00
G3	382.00	10.39	4.69	0.00	8.00	11.00	14.00	20.00

In [10]:
```
por_students.describe().applymap('{:,.2f}'.format).T # descriptive statistics
for por class
```

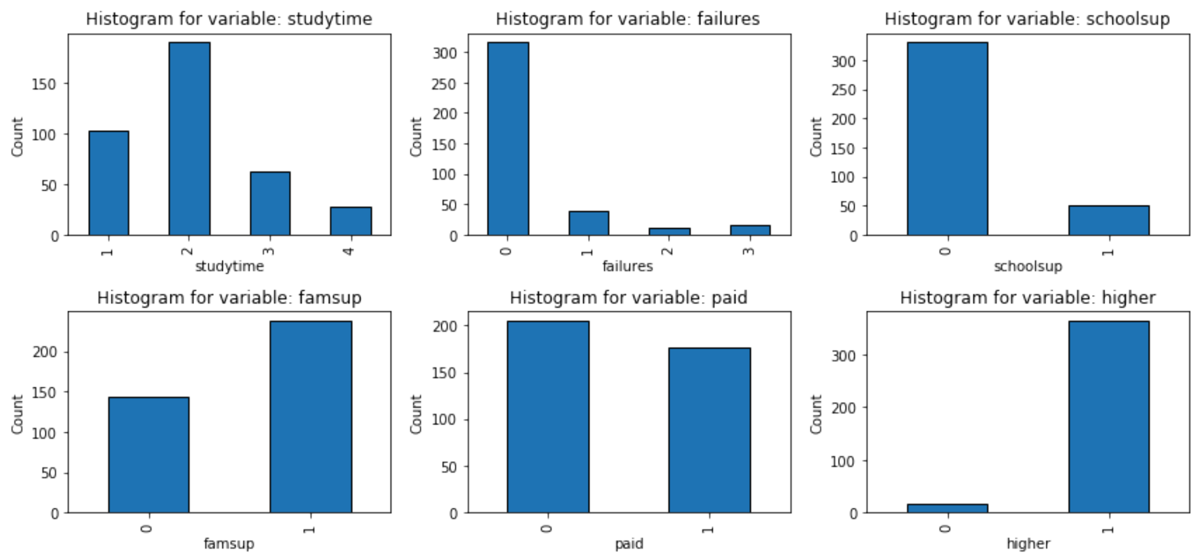
Out[10]:

	count	mean	std	min	25%	50%	75%	max
school	382.00	0.90	0.31	0.00	1.00	1.00	1.00	1.00
studytime	382.00	2.04	0.85	1.00	1.00	2.00	2.00	4.00
failures	382.00	0.14	0.51	0.00	0.00	0.00	0.00	3.00
schoolsup	382.00	0.13	0.34	0.00	0.00	0.00	0.00	1.00
famsup	382.00	0.63	0.48	0.00	0.00	1.00	1.00	1.00
paid	382.00	0.07	0.25	0.00	0.00	0.00	0.00	1.00
higher	382.00	0.95	0.21	0.00	1.00	1.00	1.00	1.00
G1	382.00	12.11	2.56	0.00	10.00	12.00	14.00	19.00
G2	382.00	12.24	2.47	5.00	11.00	12.00	14.00	19.00
G3	382.00	12.52	2.95	0.00	11.00	13.00	14.00	19.00

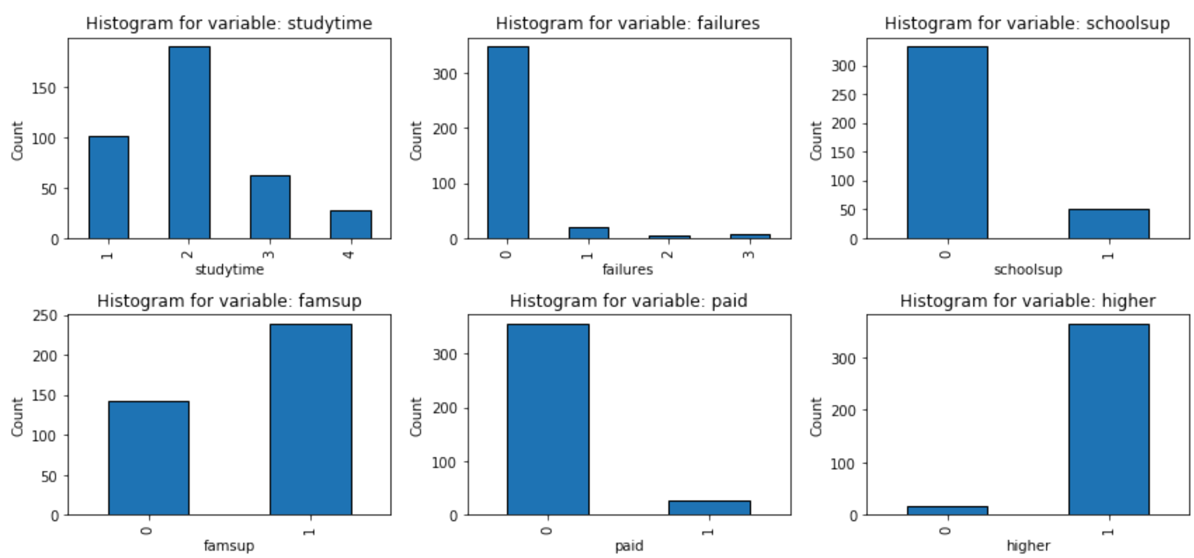
In [11]:
```
columns = ['studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'higher']
# independent variables
n_plots = len(columns)

# Plot frequency distributions for each independent variable
def plot_variables(table):
    gs = gridspec.GridSpec(3, 3)
    fig = plt.figure(figsize=(12,8))
    for i in range(n_plots):
        ax = fig.add_subplot(gs[i])
        table[columns[i]].value_counts().sort_index().plot(kind='bar', ax=ax,
edgecolor='black')
        ax.set_xlabel(columns[i])
        ax.set_ylabel('Count')
        ax.set_title('Histogram for variable: ' + columns[i])
    fig.tight_layout()
    plt.show()
```

```
In [12]: print('MAT STUDENTS')
         plot_variables(mat_students)
         print('POR STUDENTS')
         plot_variables(por_students)
```

MAT STUDENTS



POR STUDENTS



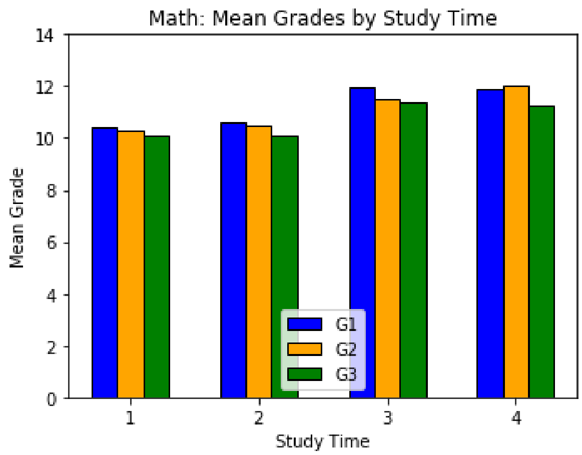

```
In [13]: mat_students['paid'].value_counts() # value counts for paid in mat class

Out[13]: 0    205
         1    177
         Name: paid, dtype: int64
```

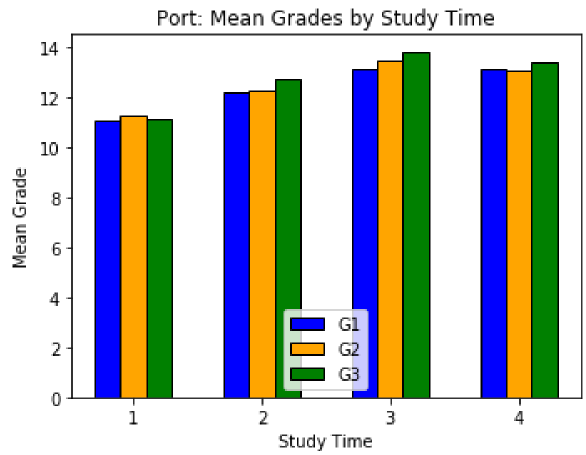
```
In [14]: por_students['paid'].value_counts() # value counts for paid in por class

Out[14]: 0    356
         1     26
         Name: paid, dtype: int64
```

Impact of Study Time on Grades

In [15]:

```
# Plot grouped bar charts for study time vs mean grades
gs = gridspec.GridSpec(1, 2)
fig = plt.figure(figsize=(12,4))
width = 0.2
studytime = [1, 2, 3, 4]
mean_grade_ticks = [0,2,4,6,8,10,12,14]

# Math class
ax = fig.add_subplot(gs[0])
for x in studytime:
    y1 = mat_students.loc[mat_students['studytime'] == x]['G1'].mean()
    y2 = mat_students.loc[mat_students['studytime'] == x]['G2'].mean()
    y3 = mat_students.loc[mat_students['studytime'] == x]['G3'].mean()
    plt.bar(x-width, y1, width, color='blue', edgecolor='black')
    plt.bar(x, y2, width, color='orange', edgecolor='black')
    plt.bar(x+width, y3, width, color='green', edgecolor='black')
plt.legend(["G1", "G2", "G3"], loc='lower center')
plt.xticks(studytime)
plt.yticks(mean_grade_ticks)
ax.set_xlabel('Study Time')
ax.set_ylabel('Mean Grade')
ax.set_title("Math: Mean Grades by Study Time")

# Port class
ax = fig.add_subplot(gs[1])
for x in studytime:
    y1 = por_students.loc[por_students['studytime'] == x]['G1'].mean()
    y2 = por_students.loc[por_students['studytime'] == x]['G2'].mean()
    y3 = por_students.loc[por_students['studytime'] == x]['G3'].mean()
    plt.bar(x-width, y1, width, color='blue', edgecolor='black')
    plt.bar(x, y2, width, color='orange', edgecolor='black')
    plt.bar(x+width, y3, width, color='green', edgecolor='black')
plt.legend(["G1", "G2", "G3"], loc='lower center')
plt.xticks(studytime)
plt.yticks(mean_grade_ticks)
ax.set_xlabel('Study Time')
ax.set_ylabel('Mean Grade')
ax.set_title("Port: Mean Grades by Study Time")

plt.show()
```





Significance Testing Using Chi-Squared Test

In [16]:
```
columns.append('school') # include school in correlation matrix
```

In [17]:
```
mat_students[columns].corr()
```

Out[17]:

	studytime	failures	schoolsup	famsup	paid	higher	school
studytime	1.000000	-0.198990	0.029744	0.159236	0.161443	0.184467	0.084631
failures	-0.198990	1.000000	0.023038	-0.023408	-0.197673	-0.369164	-0.004424
schoolsup	0.029744	0.023038	1.000000	0.082983	-0.025172	0.014643	0.134242
famsup	0.159236	-0.023408	0.082983	1.000000	0.267807	0.081949	0.157394
paid	0.161443	-0.197673	-0.025172	0.267807	1.000000	0.181856	0.009156
higher	0.184467	-0.369164	0.014643	0.081949	0.181856	1.000000	0.004648
school	0.084631	-0.004424	0.134242	0.157394	0.009156	0.004648	1.000000

In [18]:
```
por_students[columns].corr()
```

Out[18]:

	studytime	failures	schoolsup	famsup	paid	higher	school
studytime	1.000000	-0.200304	0.027910	0.151267	-0.024875	0.185895	0.086774
failures	-0.200304	1.000000	0.044395	-0.039946	0.128251	-0.324302	-0.072483
schoolsup	0.027910	0.044395	1.000000	0.091692	0.049211	0.013041	0.132719
famsup	0.151267	-0.039946	0.091692	1.000000	0.101653	0.083265	0.159462
paid	-0.024875	0.128251	0.049211	0.101653	1.000000	0.011043	-0.043368
higher	0.185895	-0.324302	0.013041	0.083265	0.011043	1.000000	0.004648
school	0.086774	-0.072483	0.132719	0.159462	-0.043368	0.004648	1.000000

In [19]:
```
# Define chi-square test function
def create_chisquare_test_table(alpha= 0.05):
    data = []
    cross_tabs = [pd.crosstab(mat_students['studytime'], mat_students['G1']),
                  pd.crosstab(mat_students['studytime'], mat_students['G2']),
                  pd.crosstab(mat_students['studytime'], mat_students['G3']),
                  pd.crosstab(por_students['studytime'], por_students['G1']),
                  pd.crosstab(por_students['studytime'], por_students['G2']),
                  pd.crosstab(por_students['studytime'], por_students['G3'])]
    gradeCounter = 1
    for i in range(0,len(cross_tabs)):
        if gradeCounter == 4:
            gradeCounter = 1
        stat, p_value, dof, expected = chi2_contingency(cross_tabs[i], correct
ion=False)
        data.append(['Math' if i < 3 else 'Port', 'G'+str(gradeCounter), round
(p_value, 2), '*' if p_value < 0.05 else ''])
        gradeCounter+=1

    return pd.DataFrame(data, columns=['Class', 'grade', 'p-value', '|< 0.05
|'])
```

In [20]:
```
create_chisquare_test_table()
```

Out[20]:

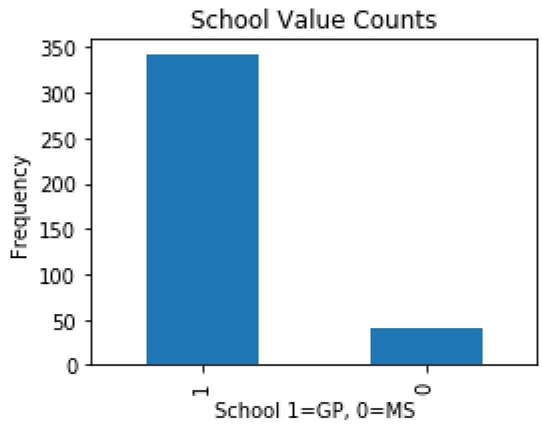
	Class	grade	p-value	|< 0.05|
0	Math	G1	0.08	
1	Math	G2	0.30	
2	Math	G3	0.20	
3	Port	G1	0.00	*
4	Port	G2	0.00	*
5	Port	G3	0.01	*

Impact of Academic Factors on Final Grades (Comparing Accross Schools)

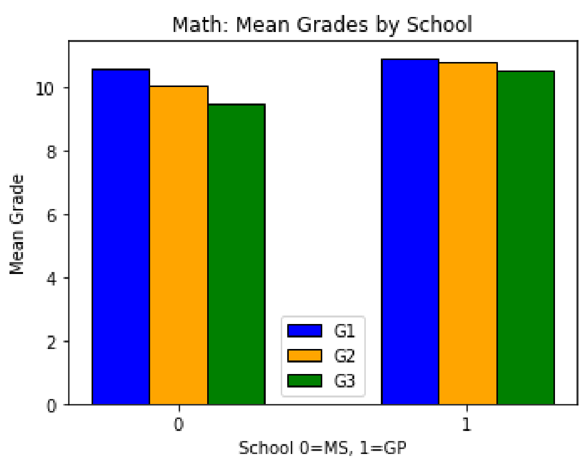
In [21]:
```
mat_students['school'].value_counts()
```

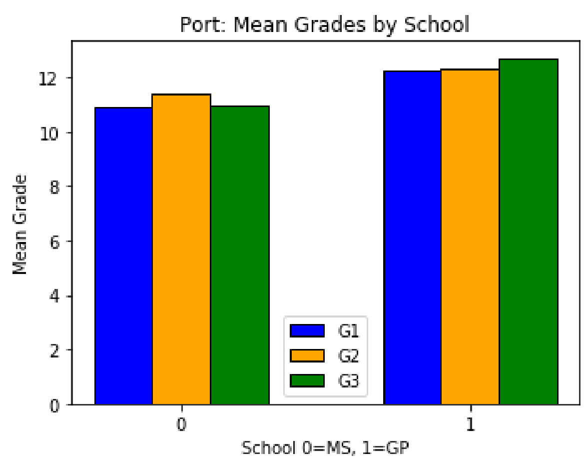
Out[21]:
```
1    342
0     40
Name: school, dtype: int64
```

In [22]:
```
# Visualize distribution school value counts
fig, ax = plt.subplots(1, sharey=True, figsize=(4, 3))
counts = mat_students['school'].value_counts()
mat_students['school'].value_counts().plot(ax=ax, kind='bar', xlabel='School 1
=GP, 0=MS', ylabel='Frequency', title='School Value Counts')
plt.show()
```



In [23]:
```python
# Plot grouped bar charts for school vs mean grades
gs = gridspec.GridSpec(1, 2)
fig = plt.figure(figsize=(12,4))
width = 0.2
school = [1, 0]

# Math class
ax = fig.add_subplot(gs[0])
for x in school:
    y1 = mat_students.loc[mat_students['school'] == x]['G1'].mean()
    y2 = mat_students.loc[mat_students['school'] == x]['G2'].mean()
    y3 = mat_students.loc[mat_students['school'] == x]['G3'].mean()
    plt.bar(x-width, y1, width, color='blue', edgecolor='black')
    plt.bar(x, y2, width, color='orange', edgecolor='black')
    plt.bar(x+width, y3, width, color='green', edgecolor='black')
plt.legend(["G1", "G2", "G3"], loc='lower center')
plt.xticks(school)
ax.set_xlabel('School 0=MS, 1=GP')
ax.set_ylabel('Mean Grade')
ax.set_title("Math: Mean Grades by School")

# Port class
ax = fig.add_subplot(gs[1])
for x in school:
    y1 = por_students.loc[por_students['school'] == x]['G1'].mean()
    y2 = por_students.loc[por_students['school'] == x]['G2'].mean()
    y3 = por_students.loc[por_students['school'] == x]['G3'].mean()
    plt.bar(x-width, y1, width, color='blue', edgecolor='black')
    plt.bar(x, y2, width, color='orange', edgecolor='black')
    plt.bar(x+width, y3, width, color='green', edgecolor='black')
plt.legend(["G1", "G2", "G3"], loc='lower center')
plt.xticks(school)
ax.set_xlabel('School 0=MS, 1=GP')
ax.set_ylabel('Mean Grade')
ax.set_title("Port: Mean Grades by School")

plt.show()
```





Significance Testing Using Independent T-Test Comparing Mean Grades Across Schools

In [24]:
```
grades = ['G1', 'G2', 'G3']

# Independent T-test Comparing Means Grades Across Schools
data = []
for grade in grades:
    x1 = mat_students.loc[mat_students['school'] == 1][grade]
    x2 = mat_students.loc[mat_students['school'] == 0][grade]
    ttests = [stats.ttest_ind(x1,x2, equal_var=False)[1]]
    data.append(['Math',round(x1.mean(),2), round(x2.mean(),2), grade, round(t
tests[0], 2), '*' if ttests[0] < 0.05 else ''])
for grade in grades:
    x1 = por_students.loc[por_students['school'] == 1][grade]
    x2 = por_students.loc[por_students['school'] == 0][grade]
    ttests = [stats.ttest_ind(x1,x2, equal_var=False)[1]]
    data.append(['Port',round(x2.mean(),2), round(x1.mean(),2), grade, round(t
tests[0], 2), '*' if ttests[0] < 0.05 else ''])

pd.DataFrame(data, columns=['Subject', 'School 0 Mean Grade', 'School 1 Mean G
rade', 'Grade', 'p-value', '|< 0.05|'])
```

Out[24]:

	Subject	School 0 Mean Grade	School 1 Mean Grade	Grade	p-value	|< 0.05|
0	Math	10.90	10.55	G1	0.55	
1	Math	10.79	10.05	G2	0.22	
2	Math	10.49	9.47	G3	0.20	
3	Port	10.88	12.26	G1	0.02	*
4	Port	11.40	12.34	G2	0.08	
5	Port	10.95	12.70	G3	0.03	*

Multiple Regression of Academic Features vs G3

In [25]:

```
# function to train and predict using linear regression model
def predict(table, columns):
    x = table[columns]
    y = table['G3']
    # split data into training and test sets
    X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20,
                                                        random_state=42)
    # fit model on training sets
    sk_model = LinearRegression().fit(X_train, y_train)
    # print intercept value
    print('Intercept:', round(sk_model.intercept_, 2))

    cols = pd.DataFrame(x.columns, columns=['Features'])
    coef = pd.DataFrame(sk_model.coef_.T, columns=['Coefficients'])

    # Calculate p-values for independent variables
    model_p = smf.ols(formula = 'G3 ~ '+' + '.join(columns), data=table).fit()
    p_vals = round(model_p.summary2().tables[1]['P>|t|'][1:], 2)
    vals = []
    for i in range(0,len(p_vals)):
        vals.append(p_vals[i])
    p_vals_df = pd.DataFrame(vals, columns=['p-value'])
    vals_strings = []
    for x in vals:
        vals_strings.append('*' if x < 0.05 else '')
    p_val_strings_df = pd.DataFrame(vals_strings, columns=['|< 0.05|'])

    # calculate R-squared value
    print('R-squared:', round(sk_model.score(X_test, y_test), 2))

    # return Dataframe output
    return pd.concat([cols['Features'], round(coef['Coefficients'], 2), p_vals
_df, p_val_strings_df], axis=1)
```

In [26]:

```
# independent variables
columns =['studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'higher', 's
chool']
```

In [27]:
```
predict(mat_students, columns) # predict for Math
```

```
Intercept: 8.02
R-squared: 0.2
```

Out[27]:

	Features	Coefficients	p-value	|< 0.05|
0	studytime	0.03	0.87	
1	failures	-2.00	0.00	*
2	schoolsup	-1.01	0.13	
3	famsup	-0.89	0.09	
4	paid	0.51	0.51	
5	higher	2.82	0.03	*
6	school	0.69	0.07	

In [28]:
```
predict(por_students, columns) # predict for Port
```

```
Intercept: 7.53
R-squared: 0.23
```

Out[28]:

	Features	Coefficients	p-value	|< 0.05|
0	studytime	0.80	0.00	*
1	failures	-1.04	0.00	*
2	schoolsup	-1.39	0.00	*
3	famsup	0.31	0.47	
4	paid	-0.95	0.05	
5	higher	2.47	0.00	*
6	school	1.39	0.00	*