Import Library

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns

from matplotlib import pyplot as plt
```

Loading Dataset

In [2]:
```
df1 = pd.read_csv('student-mat.csv',sep=';')
df2 = pd.read_csv('student-por.csv',sep=';')
```

student-mat.csv dataset on Maths Score

Diplay the 5 head rows

In [3]:
```
df1.head()
```

Out[3]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	free
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

Explore the datatype of each columns

In [4]:

```
df1.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
```

Explore the ranges of values for numeric values and distinct values for categorical values

In [5]:

```
# distinguish between categorical features and numeical features
categorical_features = [ i for i in df1 if df1[i].dtype == 'O' ]
numerical_features   = [ i for i in df1 if df1[i].dtype != 'O' ]

print("No of Numerical feature columns", len(numerical_features))
print("No of Numerical feature columns", len(categorical_features))
```

```
No of Numerical feature columns 16
No of Numerical feature columns 17
```

In [6]:

```
# Ranges of values for Numerical values

df1[numerical_features].describe()
```

Out[6]:

	age	Medu	Fedu	traveltime	studytime	failures	famrel	f
count	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395
mean	16.696203	2.749367	2.521519	1.448101	2.035443	0.334177	3.944304	3
std	1.276043	1.094735	1.088201	0.697505	0.839240	0.743651	0.896659	0
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1
25%	16.000000	2.000000	2.000000	1.000000	1.000000	0.000000	4.000000	3
50%	17.000000	3.000000	2.000000	1.000000	2.000000	0.000000	4.000000	3
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.000000	5.000000	4
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.000000	5.000000	5

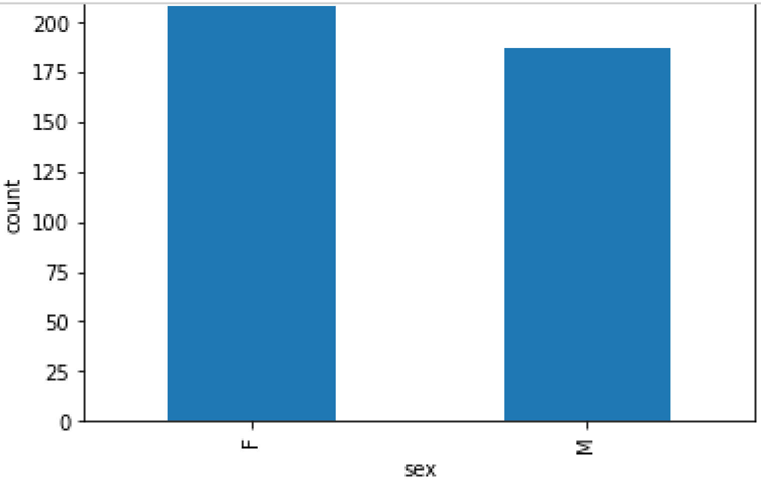
In [7]:
```
# Distinct values of Categorical values

for i in categorical_features:
    print(i,':',df1[i].unique())
```

```
school : ['GP' 'MS']
sex : ['F' 'M']
address : ['U' 'R']
famsize : ['GT3' 'LE3']
Pstatus : ['A' 'T']
Mjob : ['at_home' 'health' 'other' 'services' 'teacher']
Fjob : ['teacher' 'other' 'services' 'health' 'at_home']
reason : ['course' 'other' 'home' 'reputation']
guardian : ['mother' 'father' 'other']
schoolsup : ['yes' 'no']
famsup : ['no' 'yes']
paid : ['no' 'yes']
activities : ['no' 'yes']
nursery : ['yes' 'no']
higher : ['yes' 'no']
internet : ['no' 'yes']
romantic : ['no' 'yes']
```

Explore the distribution of columns

In [8]:
```
def plot_bar(column:str, df):
    df.groupby(column).size().plot(kind='bar')
    plt.ylabel('count')
    plt.title(f'Distribution of {column}')
    plt.show()
```

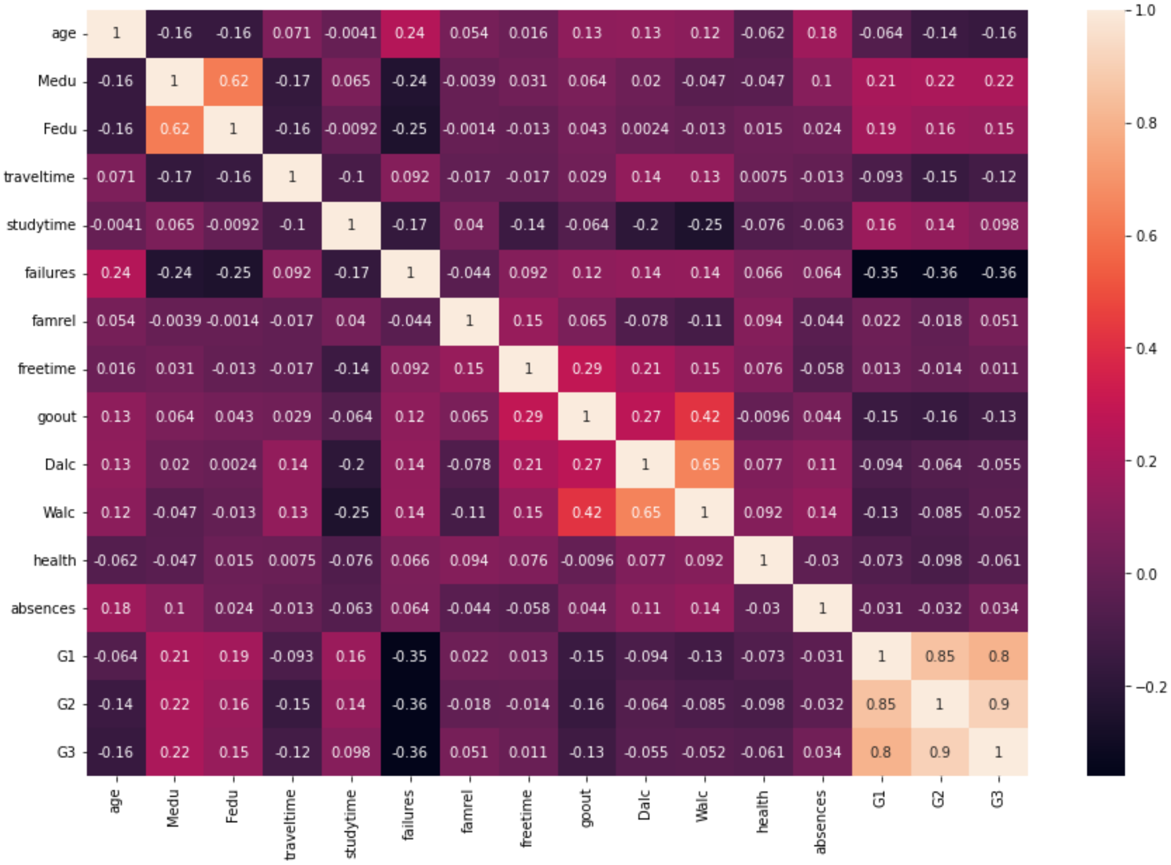
In [9]:
```
for i in df1:
    plot_bar(i,df1)
```



Explore the relationship of columns using confusion matrix

In [10]:
```
corr = df1[numerical_features].corr()
plt.figure(figsize=(15,10))
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, annot=Tr
```

Out[10]:
```
<AxesSubplot:>
```



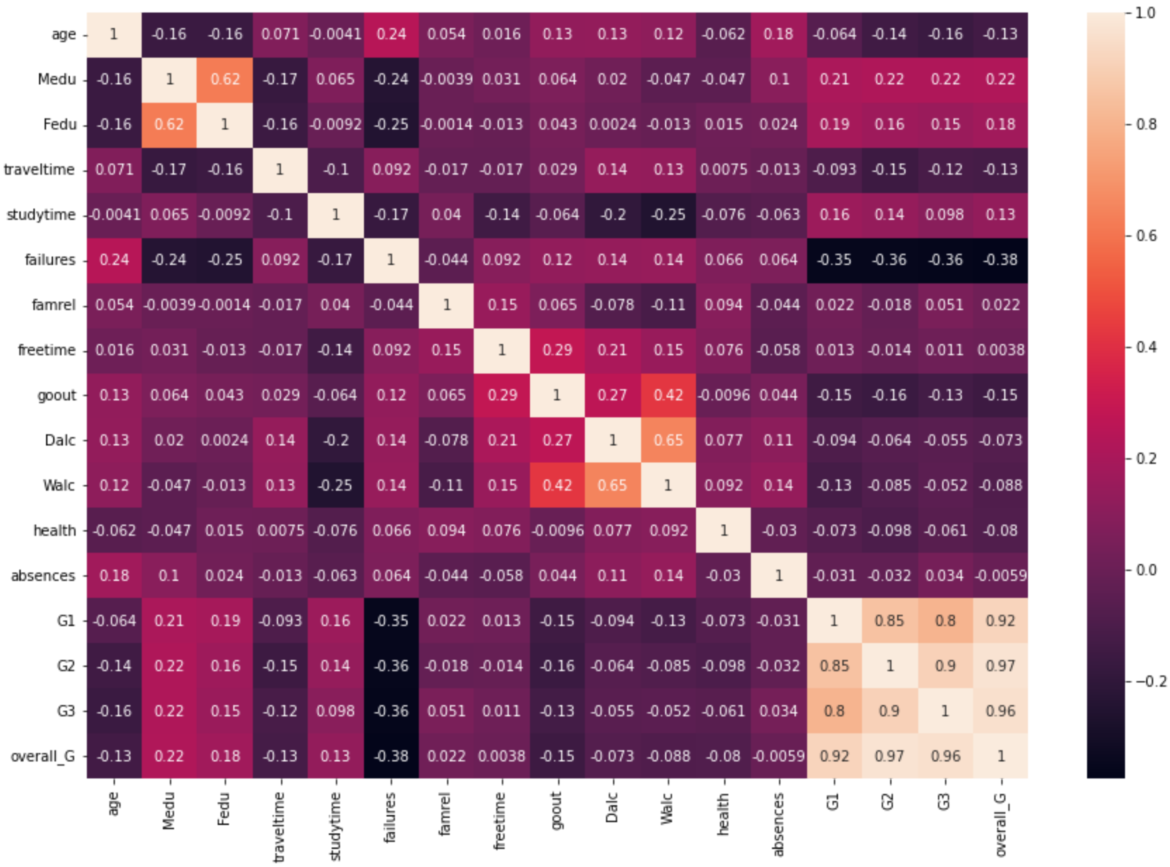
	age	Medu	Fedu	traveltime	studytime	failures	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
age	1	-0.16	-0.16	0.071	-0.0041	0.24	0.054	0.016	0.13	0.13	0.12	-0.062	0.18	-0.064	-0.14	-0.16
Medu	-0.16	1	0.62	-0.17	0.065	-0.24	-0.0039	0.031	0.064	0.02	-0.047	-0.047	0.1	0.21	0.22	0.22
Fedu	-0.16	0.62	1	-0.16	-0.0092	-0.25	-0.0014	-0.013	0.043	0.0024	-0.013	0.015	0.024	0.19	0.16	0.15
traveltime	0.071	-0.17	-0.16	1	-0.1	0.092	-0.017	-0.017	0.029	0.14	0.13	0.0075	-0.013	-0.093	-0.15	-0.12
studytime	-0.0041	0.065	-0.0092	-0.1	1	-0.17	0.04	-0.14	-0.064	-0.2	-0.25	-0.076	-0.063	0.16	0.14	0.098
failures	0.24	-0.24	-0.25	0.092	-0.17	1	-0.044	0.092	0.12	0.14	0.14	0.066	0.064	-0.35	-0.36	-0.36
famrel	0.054	-0.0039	-0.0014	-0.017	0.04	-0.044	1	0.15	0.065	-0.078	-0.11	0.094	-0.044	0.022	-0.018	0.051
freetime	0.016	0.031	-0.013	-0.017	-0.14	0.092	0.15	1	0.29	0.21	0.15	0.076	-0.058	0.013	-0.014	0.011
goout	0.13	0.064	0.043	0.029	-0.064	0.12	0.065	0.29	1	0.27	0.42	-0.0096	0.044	-0.15	-0.16	-0.13
Dalc	0.13	0.02	0.0024	0.14	-0.2	0.14	-0.078	0.21	0.27	1	0.65	0.077	0.11	-0.094	-0.064	-0.055
Walc	0.12	-0.047	-0.013	0.13	-0.25	0.14	-0.11	0.15	0.42	0.65	1	0.092	0.14	-0.13	-0.085	-0.052
health	-0.062	-0.047	0.015	0.0075	-0.076	0.066	0.094	0.076	-0.0096	0.077	0.092	1	-0.03	-0.073	-0.098	-0.061
absences	0.18	0.1	0.024	-0.013	-0.063	0.064	-0.044	-0.058	0.044	0.11	0.14	-0.03	1	-0.031	-0.032	0.034
G1	-0.064	0.21	0.19	-0.093	0.16	-0.35	0.022	0.013	-0.15	-0.094	-0.13	-0.073	-0.031	1	0.85	0.8
G2	-0.14	0.22	0.16	-0.15	0.14	-0.36	-0.018	-0.014	-0.16	-0.064	-0.085	-0.098	-0.032	0.85	1	0.9
G3	-0.16	0.22	0.15	-0.12	0.098	-0.36	0.051	0.011	-0.13	-0.055	-0.052	-0.061	0.034	0.8	0.9	1

Rank all the feature that would determine the Overall grade

In [11]:
```
df1['overall_G'] = df1[['G1','G2','G3']].sum(axis=1)
```

In [12]:
```
corr = df1[numerical_features + ['overall_G']].corr()
plt.figure(figsize=(15,10))
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, annot=Tr
```

Out[12]:
```
<AxesSubplot:>
```



In [13]:
```
corr['overall_G'].sort_values(ascending=False)
```

Out[13]:
```
overall_G     1.000000
G2            0.967999
G3            0.959873
G1            0.919386
Medu          0.224260
Fedu          0.175852
studytime     0.134565
famrel        0.021653
freetime      0.003773
absences     -0.005909
Dalc         -0.072508
health       -0.080380
Walc         -0.088025
traveltime   -0.128197
age          -0.134589
goout        -0.154511
failures     -0.375759
Name: overall_G, dtype: float64
```

student-por.csv dataset on Portuguese Language Course

Diplay the 5 head rows

In [14]:
```
df2.head()
```

Out[14]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	free
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

Explore the datatype of each columns

```
In [15]: df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 649 entries, 0 to 648
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   school      649 non-null    object
 1   sex         649 non-null    object
 2   age         649 non-null    int64 
 3   address     649 non-null    object
 4   famsize     649 non-null    object
 5   Pstatus     649 non-null    object
 6   Medu        649 non-null    int64 
 7   Fedu        649 non-null    int64 
 8   Mjob        649 non-null    object
 9   Fjob        649 non-null    object
 10  reason      649 non-null    object
 11  guardian    649 non-null    object
 12  traveltime  649 non-null    int64 
 13  studytime   649 non-null    int64 
 14  failures    649 non-null    int64 
 15  schoolsup   649 non-null    object
 16  famsup      649 non-null    object
 17  paid        649 non-null    object
 18  activities  649 non-null    object
 19  nursery     649 non-null    object
 20  higher      649 non-null    object
 21  internet    649 non-null    object
 22  romantic    649 non-null    object
 23  famrel      649 non-null    int64 
 24  freetime    649 non-null    int64 
 25  goout       649 non-null    int64 
 26  Dalc        649 non-null    int64 
 27  Walc        649 non-null    int64 
 28  health      649 non-null    int64 
 29  absences    649 non-null    int64 
 30  G1          649 non-null    int64 
 31  G2          649 non-null    int64 
 32  G3          649 non-null    int64 
dtypes: int64(16), object(17)
memory usage: 167.4+ KB
```

Explore the ranges of values for numeric values and distinct values for categorical values

distinguish between categorical features and numeical features

In [16]:
```
categorical_features = [ i for i in df2 if df2[i].dtype == 'O' ]
numerical_features   = [ i for i in df2 if df2[i].dtype != 'O' ]

print("No of Numerical feature columns", len(numerical_features))
print("No of Numerical feature columns", len(categorical_features))
```

```
No of Numerical feature columns 16
No of Numerical feature columns 17
```

Ranges of values for Numerical values

In [17]:
```
df2[numerical_features].describe()
```

Out[17]:

	age	Medu	Fedu	traveltime	studytime	failures	famrel	f
count	649.000000	649.000000	649.000000	649.000000	649.000000	649.000000	649.000000	649
mean	16.744222	2.514638	2.306626	1.568567	1.930663	0.221880	3.930663	3
std	1.218138	1.134552	1.099931	0.748660	0.829510	0.593235	0.955717	1
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1
25%	16.000000	2.000000	1.000000	1.000000	1.000000	0.000000	4.000000	3
50%	17.000000	2.000000	2.000000	1.000000	2.000000	0.000000	4.000000	3
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.000000	5.000000	4
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.000000	5.000000	5

Distinct values of Categorical values

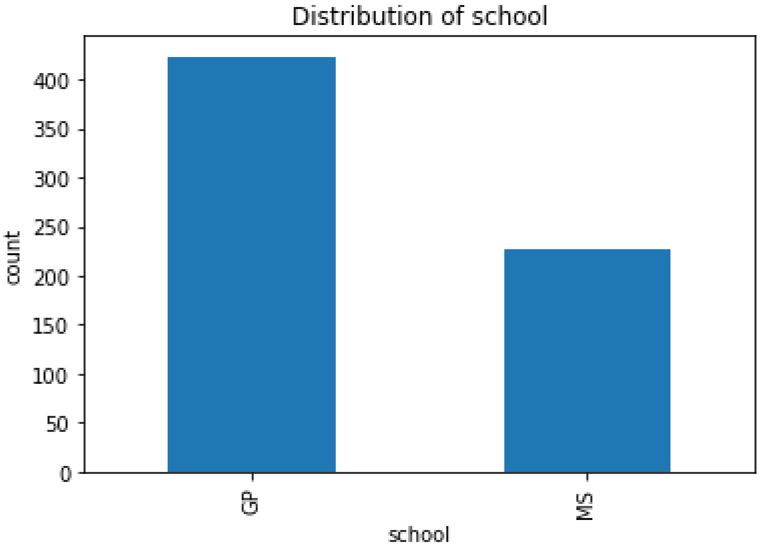
In [18]:
```
for i in categorical_features:
    print(i,':',df2[i].unique())
```

```
school : ['GP' 'MS']
sex : ['F' 'M']
address : ['U' 'R']
famsize : ['GT3' 'LE3']
Pstatus : ['A' 'T']
Mjob : ['at_home' 'health' 'other' 'services' 'teacher']
Fjob : ['teacher' 'other' 'services' 'health' 'at_home']
reason : ['course' 'other' 'home' 'reputation']
guardian : ['mother' 'father' 'other']
schoolsup : ['yes' 'no']
famsup : ['no' 'yes']
paid : ['no' 'yes']
activities : ['no' 'yes']
nursery : ['yes' 'no']
higher : ['yes' 'no']
internet : ['no' 'yes']
romantic : ['no' 'yes']
```

Explore the distribution of columns

In [19]:
```
def plot_bar(column:str, df):
    df.groupby(column).size().plot(kind='bar')
    plt.ylabel('count')
    plt.title(f'Distribution of {column}')
    plt.show()
```

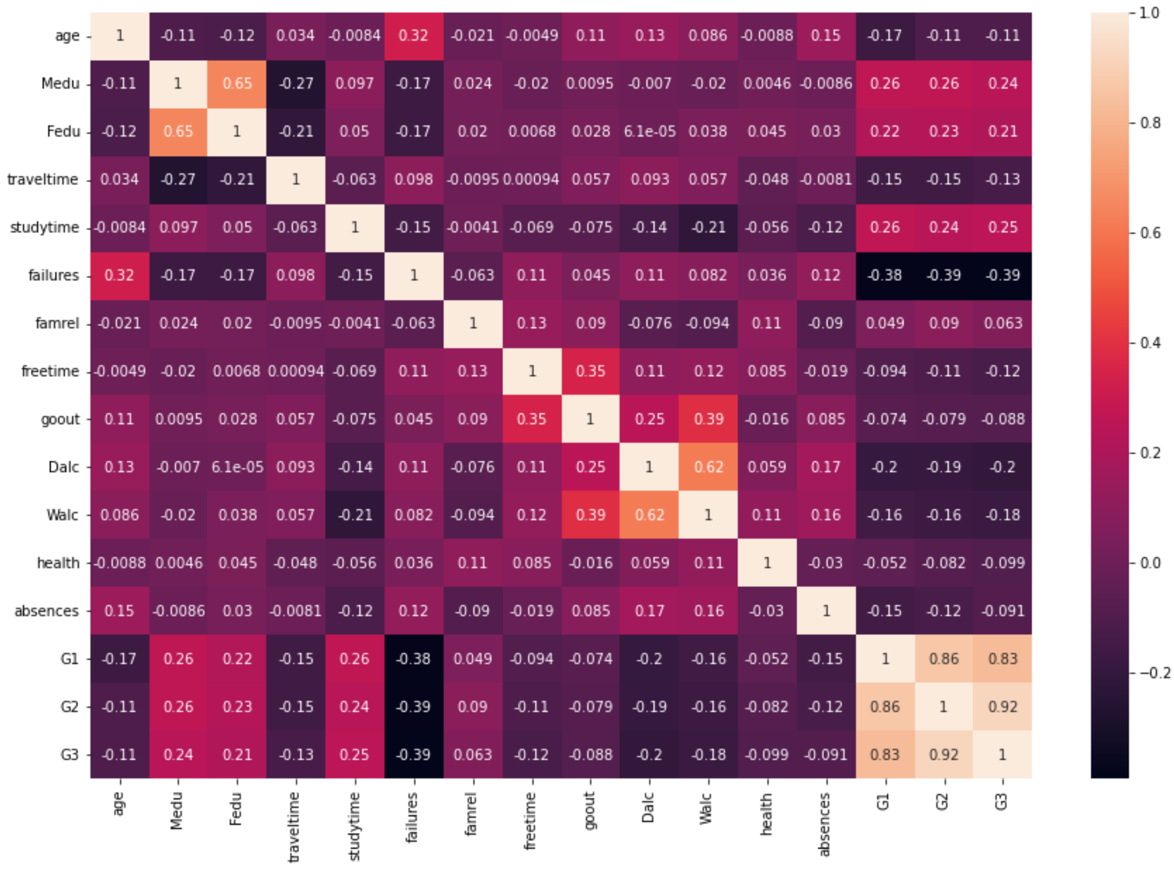
In [20]:
```
for i in df2:
    plot_bar(i,df2)
```



Explore the relationship of columns using confusion matrix

In [21]:
```
corr = df2[numerical_features].corr()
plt.figure(figsize=(15,10))
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, annot=Tr
```

Out[21]:
```
<AxesSubplot:>
```

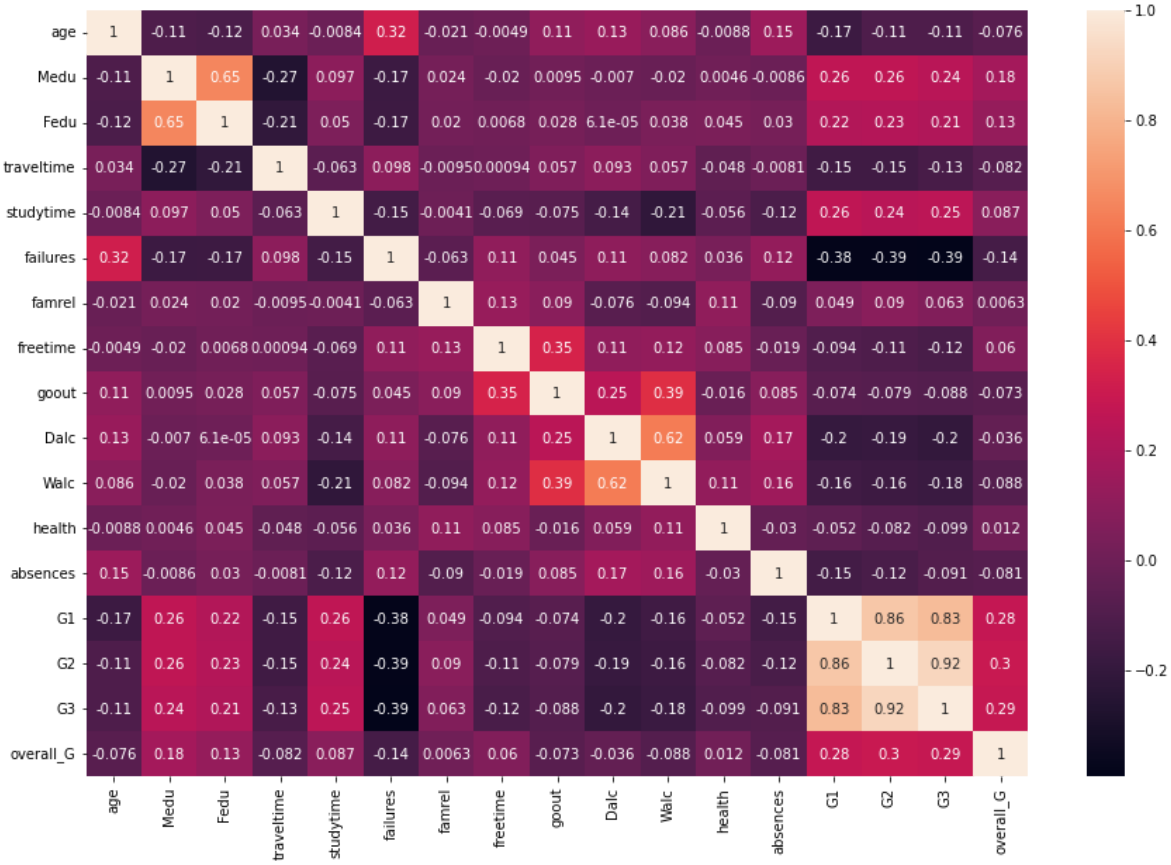


Rank all the feature that would determine the Overall grade

In [22]:
```
df2['overall_G'] = df1[['G1','G2','G3']].sum(axis=1)
```

In [23]:
```
corr = df2[numerical_features + ['overall_G']].corr()
plt.figure(figsize=(15,10))
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, annot=Tr
```

Out[23]:
```
<AxesSubplot:>
```



In [24]:
```
corr['overall_G'].sort_values(ascending=False)
```

Out[24]:
```
overall_G     1.000000
G2            0.296009
G3            0.285873
G1            0.276191
Medu          0.176824
Fedu          0.127257
studytime     0.087463
freetime      0.060016
health        0.011510
famrel        0.006276
Dalc         -0.035689
goout        -0.073104
age          -0.075857
absences     -0.080829
traveltime   -0.081722
Walc         -0.087647
failures     -0.139271
Name: overall_G, dtype: float64
```