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

In [3]:
```
#load data
mat = pd.read_csv('C:/Users/nneam/OneDrive/Documents/540Assignments/student-mat.csv')
por = pd.read_csv('C:/Users/nneam/OneDrive/Documents/540Assignments/student-por.csv')
```

In [4]:
```
#Concat tables
x1 = pd.concat([mat, por])
df = pd.DataFrame(data=x1, columns = x1.columns)
```

In [5]:
```
#Show all columns and show df
pd.set_option('display.max_columns', None)
df.head()
```

Out[5]:

	school	subject	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	studytime	failures	scho
0	GP	m	F	18	U	GT3	A	4	4	at_home	teacher	course	mother	2	2	0	
1	GP	m	F	17	U	GT3	T	1	1	at_home	other	course	father	1	2	0	
2	GP	m	F	15	U	LE3	T	1	1	at_home	other	other	mother	1	2	3	
3	GP	m	F	15	U	GT3	T	4	2	health	services	home	mother	1	3	0	
4	GP	m	F	16	U	GT3	T	3	3	other	other	home	father	1	2	0	

Data Cleaning

In [6]:

```
#Df info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1044 entries, 0 to 648
Data columns (total 34 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   school      1044 non-null   object
 1   subject     1044 non-null   object
 2   sex         1044 non-null   object
 3   age         1044 non-null   int64
 4   address     1044 non-null   object
 5   famsize     1044 non-null   object
 6   Pstatus     1044 non-null   object
 7   Medu        1044 non-null   int64
 8   Fedu        1044 non-null   int64
 9   Mjob        1044 non-null   object
 10  Fjob        1044 non-null   object
 11  reason      1044 non-null   object
 12  guardian    1044 non-null   object
 13  traveltime  1044 non-null   int64
 14  studytime   1044 non-null   int64
 15  failures    1044 non-null   int64
 16  schoolsup   1044 non-null   object
 17  famsup      1044 non-null   object
 18  paid        1044 non-null   object
 19  activities  1044 non-null   object
 20  nursery     1044 non-null   object
 21  higher      1044 non-null   object
 22  internet    1044 non-null   object
 23  romantic    1044 non-null   object
 24  famrel      1044 non-null   int64
 25  freetime    1044 non-null   int64
 26  goout       1044 non-null   int64
 27  Dalc        1044 non-null   int64
 28  Walc        1044 non-null   int64
 29  health      1044 non-null   int64
 30  absences    1044 non-null   int64
 31  G1          1044 non-null   int64
 32  G2          1044 non-null   int64
 33  G3          1044 non-null   int64
dtypes: int64(16), object(18)
memory usage: 285.5+ KB
```

In [7]:
```
df['school'].value_counts(ascending=False)
```

Out[7]:
```
GP    772
MS    272
Name: school, dtype: int64
```

In [8]:
```
df['sex'].value_counts(ascending=False)
```

Out[8]:
```
F    591
M    453
Name: sex, dtype: int64
```

In [9]:
```
df['address'].value_counts(ascending=False)
```

Out[9]:
```
U    759
R    285
Name: address, dtype: int64
```

In [10]:
```
df['famsize'].value_counts(ascending=False)
```

Out[10]:
```
GT3    738
LE3    306
Name: famsize, dtype: int64
```

In [11]:
```
df['Mjob'].value_counts(ascending=False)
```

Out[11]:
```
other       399
services    239
at_home     194
teacher     130
health       82
Name: Mjob, dtype: int64
```

In [12]:
```
df['Fjob'].value_counts(ascending=False)
```

Out[12]:
```
other       584
services    292
teacher      65
at_home      62
health       41
Name: Fjob, dtype: int64
```

In [13]:
```
df['reason'].value_counts(ascending=False)
```

Out[13]:
```
course        430
home          258
reputation    248
other         108
Name: reason, dtype: int64
```

In [14]:
```
df['guardian'].value_counts(ascending=False)
```

Out[14]:
```
mother    728
father    243
other      73
Name: guardian, dtype: int64
```

In [15]:
```
df['Pstatus'].value_counts(ascending=False)
```

Out[15]:
```
T    923
A    121
Name: Pstatus, dtype: int64
```

In [16]:

```
#Replcing text column value for model and combining data
df['school'] = df['school'].replace(['GP','MS'],['1','0'])
df['subject'] = df['subject'].replace(['m','p'],['1','0'])
df['sex'] = df['sex'].replace(['F','M'],['1','0'])
df['address'] = df['address'].replace(['U','R'],['1','0'])
df['famsize'] = df['famsize'].replace(['GT3','LE3'],['1','0'])
df['Mjob'] = df['Mjob'].replace(['other','services','at_home','teacher','health'],['0','1','2','3','4'])
df['Fjob'] = df['Fjob'].replace(['other','services','at_home','teacher','health'],['0','1','2','3','4'])
df['reason'] = df['reason'].replace(['course','home','reputation','other'],['0','1','2','3'])
df['guardian'] = df['guardian'].replace(['mother','father','other'],['0','1','2'])
df['Pstatus'] = df['Pstatus'].replace(['T','A'],['1','0'])
df['schoolsup'] = df['schoolsup'].replace(['yes','no'],['1','0'])
df['famsup'] = df['famsup'].replace(['yes','no'],['1','0'])
df['paid'] = df['paid'].replace(['yes','no'],['1','0'])
df['activities'] = df['activities'].replace(['yes','no'],['1','0'])
df['nursery'] = df['nursery'].replace(['yes','no'],['1','0'])
df['higher'] = df['higher'].replace(['yes','no'],['1','0'])
df['internet'] = df['internet'].replace(['yes','no'],['1','0'])
df['romantic'] = df['romantic'].replace(['yes','no'],['1','0'])
#Data Preview
df.head()
```

Out[16]:

	school	subject	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	studytime	failures	schoolsup
0	1	1	1	18	1	1	0	4	4	2	3	0	0	2	2	0	1
1	1	1	1	17	1	1	1	1	1	2	0	0	1	1	2	0	0
2	1	1	1	15	1	0	1	1	1	2	0	3	0	1	2	3	1
3	1	1	1	15	1	1	1	4	2	4	1	1	0	1	3	0	0
4	1	1	1	16	1	1	1	3	3	0	0	1	1	1	2	0	0

In [17]:

```
#Changing value type
df['school'] = df.school.astype(int)
df['subject'] = df.subject.astype(int)
df['sex'] = df.sex.astype(int)
df['address'] = df.address.astype(int)
df['famsize'] = df.famsize.astype(int)
df['Mjob'] = df.Mjob.astype(int)
df['Fjob'] = df.Fjob.astype(int)
df['reason'] = df.reason.astype(int)
df['guardian'] = df.guardian.astype(int)
df['Pstatus'] = df.Pstatus.astype(int)
df['schoolsup'] = df.schoolsup.astype(int)
df['famsup'] = df.famsup.astype(int)
df['paid'] = df.paid.astype(int)
df['activities'] = df.activities.astype(int)
df['higher'] = df.higher.astype(int)
df['internet'] = df.internet.astype(int)
df['romantic'] = df.romantic.astype(int)
df['nursery'] = df.Pstatus.astype(int)
```

In [18]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1044 entries, 0 to 648
Data columns (total 34 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   school      1044 non-null   int32
 1   subject     1044 non-null   int32
 2   sex         1044 non-null   int32
 3   age         1044 non-null   int64
 4   address     1044 non-null   int32
 5   famsize     1044 non-null   int32
 6   Pstatus     1044 non-null   int32
 7   Medu        1044 non-null   int64
 8   Fedu        1044 non-null   int64
 9   Mjob        1044 non-null   int32
 10  Fjob        1044 non-null   int32
 11  reason      1044 non-null   int32
 12  guardian    1044 non-null   int32
 13  traveltime  1044 non-null   int64
 14  studytime   1044 non-null   int64
 15  failures    1044 non-null   int64
 16  schoolsup   1044 non-null   int32
 17  famsup      1044 non-null   int32
 18  paid        1044 non-null   int32
 19  activities  1044 non-null   int32
 20  nursery     1044 non-null   int32
 21  higher      1044 non-null   int32
 22  internet    1044 non-null   int32
 23  romantic    1044 non-null   int32
 24  famrel      1044 non-null   int64
 25  freetime    1044 non-null   int64
 26  goout       1044 non-null   int64
 27  Dalc        1044 non-null   int64
 28  Walc        1044 non-null   int64
 29  health      1044 non-null   int64
 30  absences    1044 non-null   int64
 31  G1          1044 non-null   int64
 32  G2          1044 non-null   int64
 33  G3          1044 non-null   int64
dtypes: int32(18), int64(16)
memory usage: 212.1 KB
```

In [19]:
```
# Getting mean of weekend and weekday consumption
df['avg_consumption'] = df[['Dalc', 'Walc']].mean(axis=1)
df.avg_consumption = df.avg_consumption.round()
```

In [20]:
```
df.head()
```

Out[20]:

	school	subject	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	studytime	failures	schoolsup
0	1	1	1	18	1	1	0	4	4	2	3	0	0	2	2	0	1
1	1	1	1	17	1	1	1	1	1	2	0	0	1	1	2	0	0
2	1	1	1	15	1	0	1	1	1	2	0	3	0	1	2	3	1
3	1	1	1	15	1	1	1	4	2	4	1	1	0	1	3	0	0
4	1	1	1	16	1	1	1	3	3	0	0	1	1	1	2	0	0

In [21]:
```
df['avg_consumption'].value_counts(ascending=False)
```

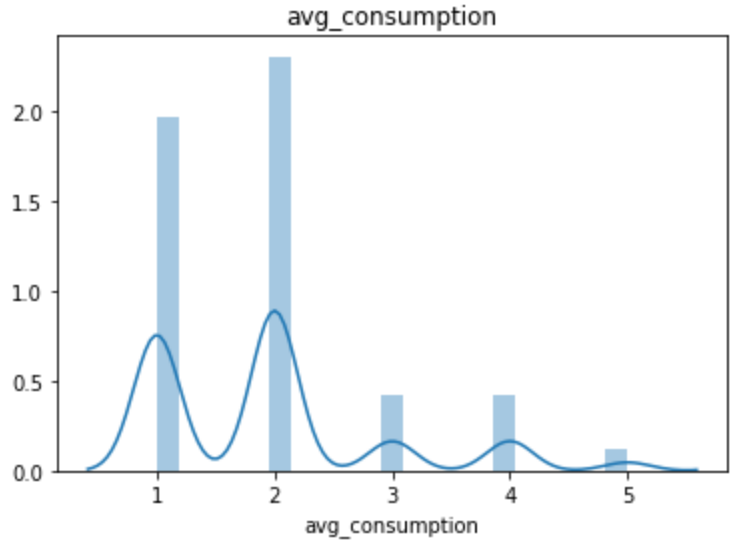
Out[21]:
```
2.0    459
1.0    391
4.0     85
3.0     85
5.0     24
Name: avg_consumption, dtype: int64
```

In [22]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1044 entries, 0 to 648
Data columns (total 35 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   school           1044 non-null   int32
 1   subject          1044 non-null   int32
 2   sex              1044 non-null   int32
 3   age              1044 non-null   int64
 4   address          1044 non-null   int32
 5   famsize          1044 non-null   int32
 6   Pstatus          1044 non-null   int32
 7   Medu             1044 non-null   int64
 8   Fedu             1044 non-null   int64
 9   Mjob             1044 non-null   int32
 10  Fjob             1044 non-null   int32
 11  reason           1044 non-null   int32
 12  guardian         1044 non-null   int32
 13  traveltime       1044 non-null   int64
 14  studytime        1044 non-null   int64
 15  failures         1044 non-null   int64
 16  schoolsup        1044 non-null   int32
 17  famsup           1044 non-null   int32
 18  paid             1044 non-null   int32
 19  activities       1044 non-null   int32
 20  nursery          1044 non-null   int32
 21  higher           1044 non-null   int32
 22  internet         1044 non-null   int32
 23  romantic         1044 non-null   int32
 24  famrel           1044 non-null   int64
 25  freetime         1044 non-null   int64
 26  goout            1044 non-null   int64
 27  Dalc             1044 non-null   int64
 28  Walc             1044 non-null   int64
 29  health           1044 non-null   int64
 30  absences         1044 non-null   int64
 31  G1               1044 non-null   int64
 32  G2               1044 non-null   int64
 33  G3               1044 non-null   int64
 34  avg_consumption  1044 non-null   float64
dtypes: float64(1), int32(18), int64(16)
memory usage: 220.2 KB
```

In [23]:
```
sns.distplot(a=df.avg_consumption).set_title('avg_consumption')
```

Out[23]:
```
Text(0.5, 1.0, 'avg_consumption')
```



In [24]:
```
#grouping to low(1-2) and high(3-5) risk
df['avg_consumption'] = df['avg_consumption'].replace([1,2,3,4,5],[0,0,1,1,1])
```

In [25]:
```
#turning to int
df['avg_consumption'] = df.avg_consumption.astype(int)
```

In [26]:
```
df['avg_consumption'].value_counts(ascending=False)
```

Out[26]:
```
0    850
1    194
Name: avg_consumption, dtype: int64
```

In [27]:
```
df.head()
```

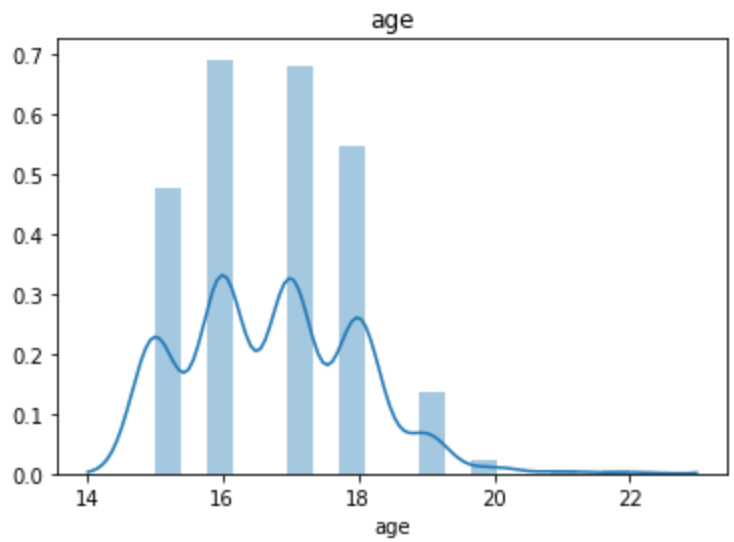
Out[27]:

	school	subject	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	studytime	failures	schoolsup
0	1	1	1	18	1	1	0	4	4	2	3	0	0	2	2	0	1
1	1	1	1	17	1	1	1	1	1	2	0	0	1	1	2	0	0
2	1	1	1	15	1	0	1	1	1	2	0	3	0	1	2	3	1
3	1	1	1	15	1	1	1	4	2	4	1	1	0	1	3	0	0
4	1	1	1	16	1	1	1	3	3	0	0	1	1	1	2	0	0

Data Exploration

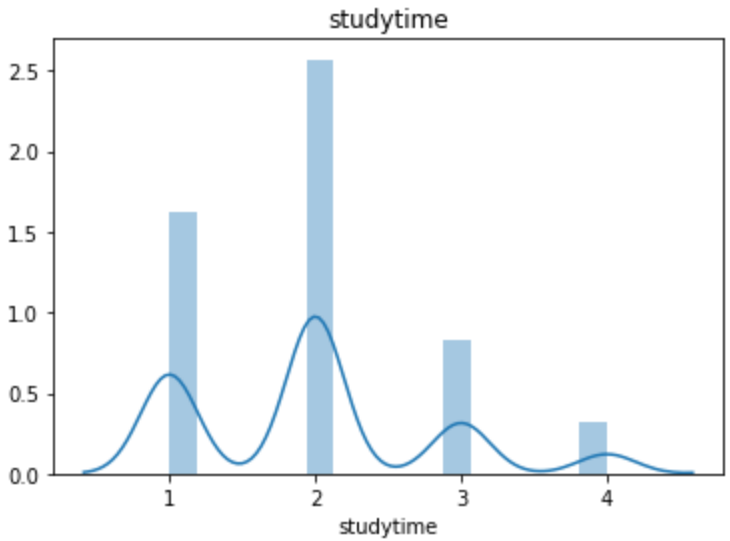
In [28]:
```
# Desnity and Histogram for Age
sns.distplot(a=df.age).set_title('age')
```

Out[28]: Text(0.5, 1.0, 'age')



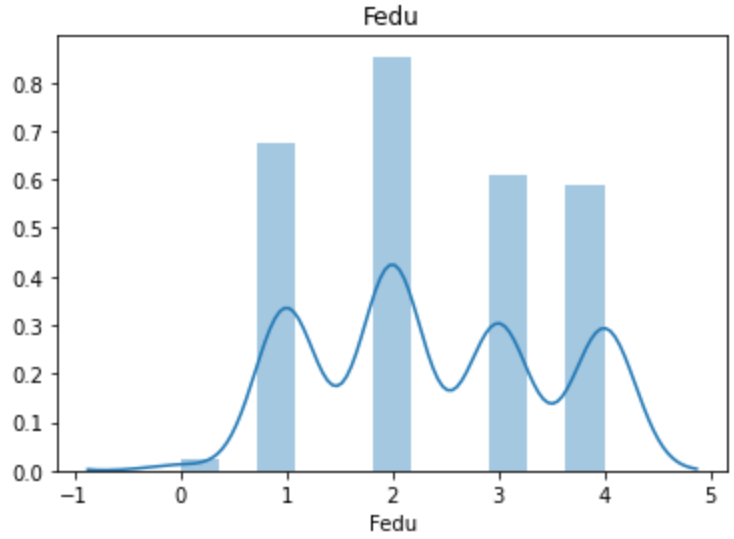
In [33]:
```
# Desnity and Histogram for failures
sns.distplot(a=df.studytime).set_title('studytime')
```

Out[33]:
```
Text(0.5, 1.0, 'studytime')
```



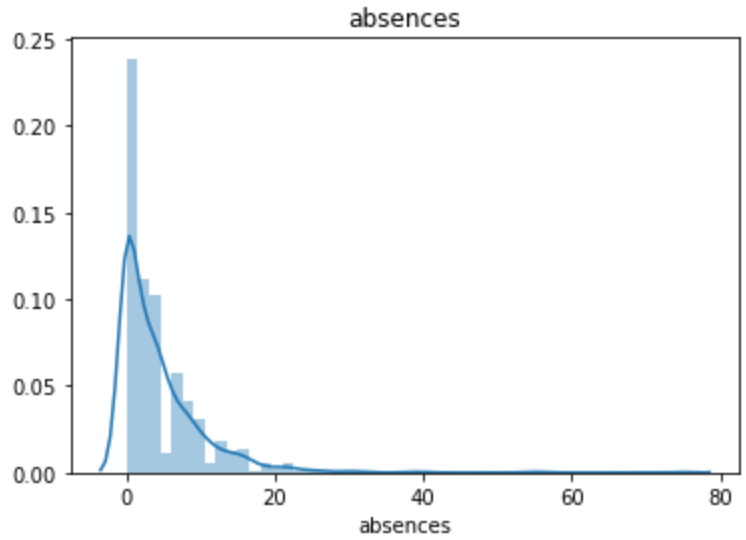
In [30]:
```
sns.distplot(a=df.Fedu).set_title('Fedu')
```

Out[30]: Text(0.5, 1.0, 'Fedu')



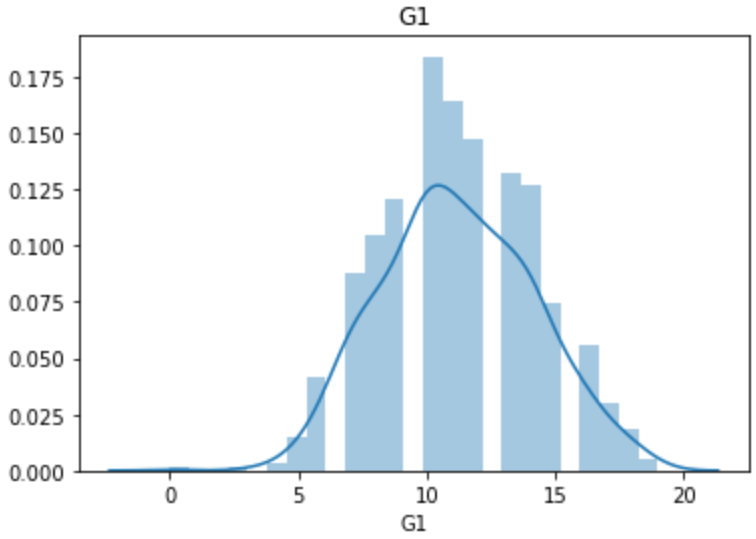
In [216]:
```
sns.distplot(a=df.absences).set_title('absences')
```

Out[216]: Text(0.5, 1.0, 'absences')



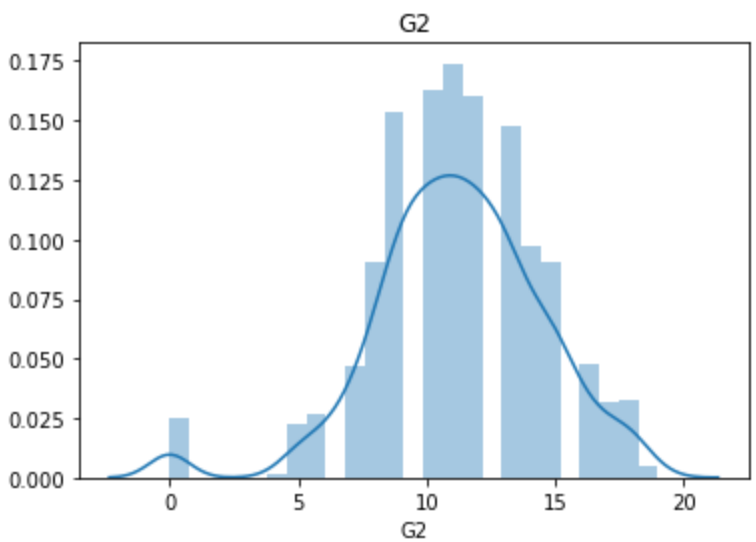
In [217]:
```
sns.distplot(a=df.G1).set_title('G1')
```

Out[217]: Text(0.5, 1.0, 'G1')



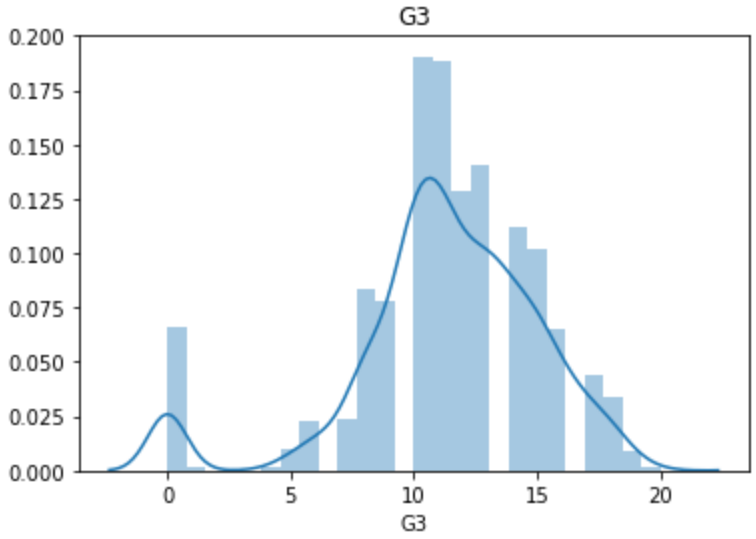
In [218]:
```
sns.distplot(a=df.G2).set_title('G2')
```

Out[218]: Text(0.5, 1.0, 'G2')



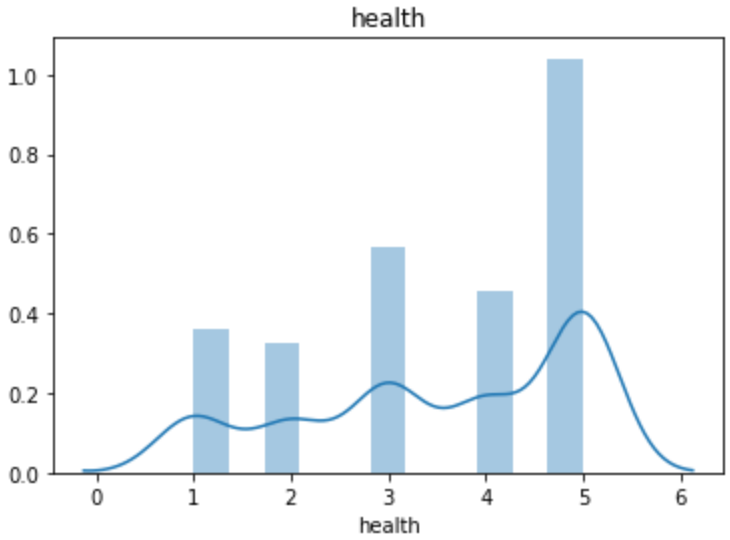
In [219]:
```
sns.distplot(a=df.G3).set_title('G3')
```

Out[219]: Text(0.5, 1.0, 'G3')



In [220]:
```
sns.distplot(a=df.health).set_title('health')
```

Out[220]: Text(0.5, 1.0, 'health')



In [221]:

```
#Assign pearson correlation
pearson = df.corr(method = 'pearson')
pearson
```

Out[221]:

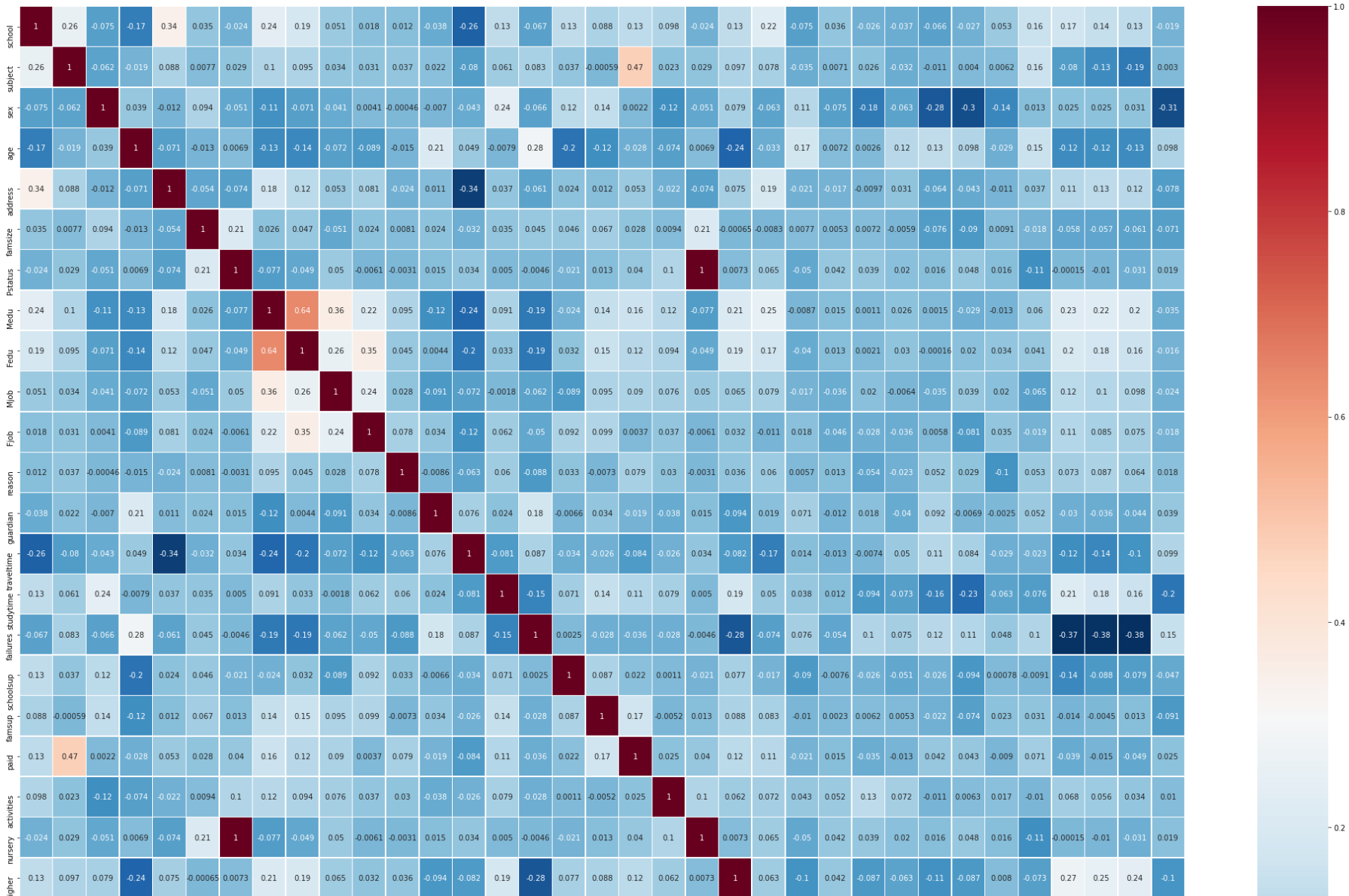
	school	subject	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reaso
school	1.000000	0.256088	-0.074955	-0.169938	0.341657	0.034882	-0.024030	0.235114	0.187611	0.051069	0.018105	0.01240
subject	0.256088	1.000000	-0.062192	-0.018790	0.087916	0.007705	0.029497	0.101246	0.094795	0.033601	0.030833	0.03667
sex	-0.074955	-0.062192	1.000000	0.038832	-0.011556	0.094361	-0.051337	-0.109387	-0.070786	-0.040727	0.004098	-0.00046
age	-0.169938	-0.018790	0.038832	1.000000	-0.071257	-0.013290	0.006887	-0.130196	-0.138521	-0.071665	-0.088954	-0.01476
address	0.341657	0.087916	-0.011556	-0.071257	1.000000	-0.054484	-0.074097	0.179720	0.124303	0.053208	0.081162	-0.02444
famsize	0.034882	0.007705	0.094361	-0.013290	-0.054484	1.000000	0.207302	0.025556	0.047290	-0.051438	0.023802	0.00809
Pstatus	-0.024030	0.029497	-0.051337	0.006887	-0.074097	0.207302	1.000000	-0.077133	-0.049156	0.050379	-0.006062	-0.00307
Medu	0.235114	0.101246	-0.109387	-0.130196	0.179720	0.025556	-0.077133	1.000000	0.642063	0.360673	0.219845	0.09463
Fedu	0.187611	0.094795	-0.070786	-0.138521	0.124303	0.047290	-0.049156	0.642063	1.000000	0.257518	0.354572	0.04464
Mjob	0.051069	0.033601	-0.040727	-0.071665	0.053208	-0.051438	0.050379	0.360673	0.257518	1.000000	0.243789	0.02802
Fjob	0.018105	0.030833	0.004098	-0.088954	0.081162	0.023802	-0.006062	0.219845	0.354572	0.243789	1.000000	0.07849
reason	0.012405	0.036672	-0.000463	-0.014762	-0.024447	0.008097	-0.003076	0.094632	0.044644	0.028024	0.078494	1.00000
guardian	-0.038027	0.022040	-0.006987	0.210603	0.011230	0.024158	0.015104	-0.116713	0.004411	-0.090597	0.033875	-0.00861
traveltime	-0.258834	-0.079881	-0.042508	0.049216	-0.343803	-0.031550	0.033883	-0.238181	-0.196328	-0.071953	-0.116160	-0.06326
studytime	0.133255	0.060934	0.239972	-0.007870	0.037480	0.035109	0.005049	0.090616	0.033458	-0.001822	0.062003	0.06019
failures	-0.066856	0.083043	-0.065543	0.282364	-0.061160	0.044589	-0.004615	-0.187769	-0.191390	-0.062440	-0.050129	-0.08784
schoolsup	0.130498	0.037141	0.119411	-0.202824	0.023583	0.045552	-0.020789	-0.023618	0.032450	-0.088825	0.091607	0.03253
famsup	0.088460	-0.000590	0.137696	-0.116904	0.011975	0.067340	0.013370	0.143063	0.153342	0.094521	0.098645	-0.00733
paid	0.130125	0.473453	0.002179	-0.027917	0.053024	0.028290	0.040341	0.161349	0.118897	0.089562	0.003684	0.07938
activities	0.097932	0.022794	-0.116368	-0.073648	-0.022095	0.009434	0.100576	0.116924	0.093800	0.075502	0.037356	0.03010
nursery	-0.024030	0.029497	-0.051337	0.006887	-0.074097	0.207302	1.000000	-0.077133	-0.049156	0.050379	-0.006062	-0.00307
higher	0.131382	0.096707	0.078775	-0.244601	0.074716	-0.000650	0.007339	0.206551	0.191956	0.065068	0.032135	0.03627
internet	0.222993	0.078377	-0.062671	-0.033229	0.194790	-0.008315	0.065260	0.249728	0.170012	0.079108	-0.010792	0.05970
romantic	-0.074506	-0.034534	0.108944	0.173800	-0.021209	0.007656	-0.050021	-0.008685	-0.039906	-0.016880	0.017864	0.00566
famrel	0.036359	0.007091	-0.074725	0.007162	-0.016801	0.005328	0.042448	0.015004	0.013066	-0.036080	-0.046109	0.01313
freetime	-0.026008	0.025949	-0.181603	0.002645	-0.009744	0.007249	0.038714	0.001054	0.002142	0.019621	-0.027570	-0.05394
goout	-0.037000	-0.032011	-0.062530	0.118510	0.030790	-0.005889	0.020498	0.025614	0.030075	-0.006393	-0.035591	-0.02283

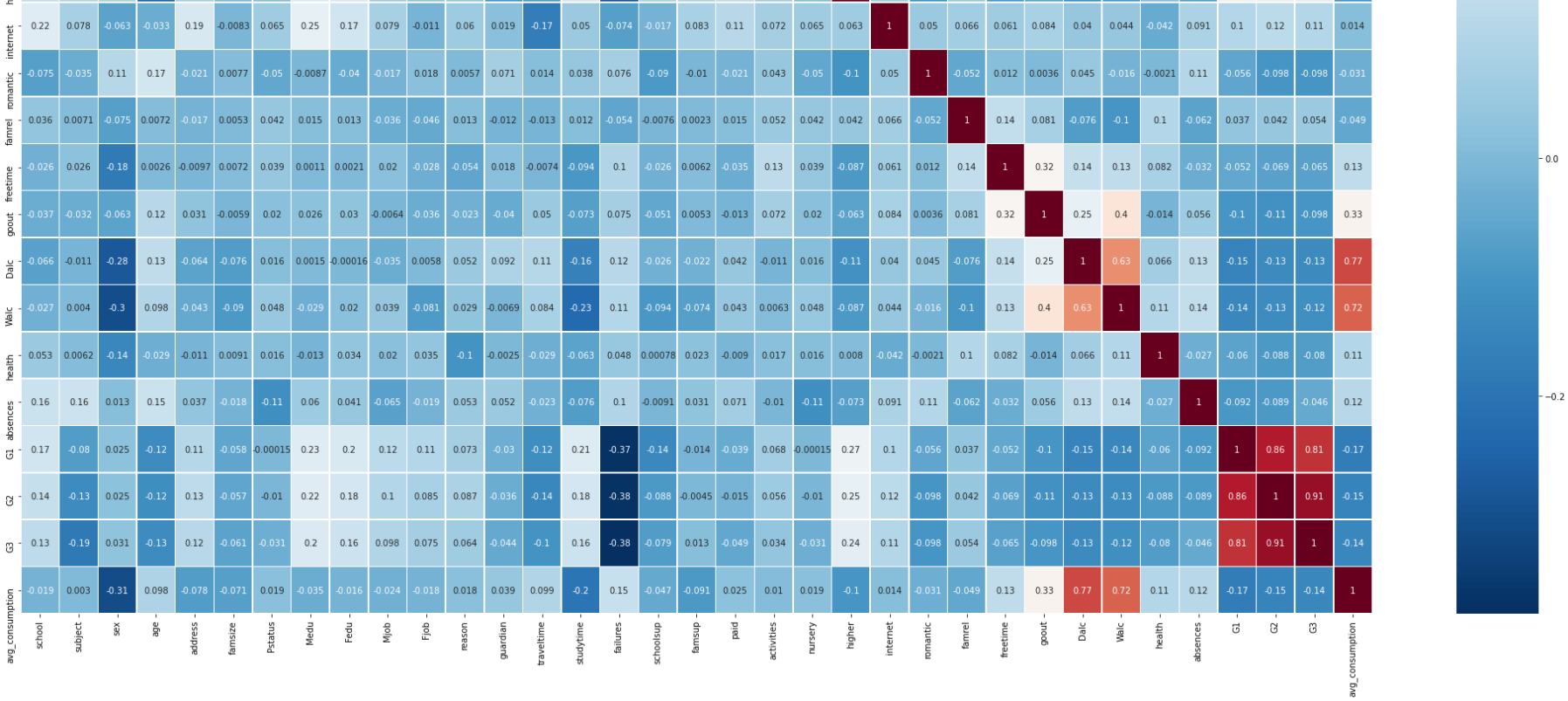
	school	subject	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reaso
Dalc	-0.066006	-0.011335	-0.275928	0.133453	-0.064030	-0.075646	0.015777	0.001515	-0.000165	-0.034583	0.005824	0.05221
Walc	-0.026539	0.004043	-0.302623	0.098291	-0.043391	-0.090019	0.047575	-0.029331	0.019524	0.039221	-0.080822	0.02917
health	0.053214	0.006205	-0.141113	-0.029129	-0.010895	0.009147	0.016213	-0.013254	0.034288	0.020452	0.035111	-0.10277
absences	0.155146	0.160125	0.013076	0.153196	0.036696	-0.018284	-0.113955	0.059708	0.040829	-0.064922	-0.019389	0.05303
G1	0.169123	-0.079727	0.025122	-0.124121	0.113113	-0.057610	-0.000155	0.226101	0.195898	0.119770	0.108041	0.07347
G2	0.144858	-0.126459	0.025024	-0.119475	0.127793	-0.057469	-0.010219	0.224662	0.182634	0.103206	0.084753	0.08707
G3	0.127114	-0.187166	0.031472	-0.125282	0.117696	-0.061209	-0.030693	0.201472	0.159796	0.098248	0.074755	0.06360
avg_consumption	-0.019389	0.003045	-0.312146	0.097649	-0.077611	-0.071077	0.019115	-0.035194	-0.016259	-0.024500	-0.018363	0.01835

In [222]:

```
#Initial Pearson Correlation Heatmap
fig, ax = plt.subplots(figsize=(35,35))
sns.heatmap(pearson,
            xticklabels=pearson.columns,
            yticklabels=pearson.columns,
            cmap='RdBu_r',
            annot=True,
            linewidth= 0.5,
            annot_kws={"size": 10},
            ax=ax)
```

Out[222]: <matplotlib.axes._subplots.AxesSubplot at 0x21507fa7a30>




0.0
−0.2

In [223]:
```
df.var()
```

Out[223]:
```
school              0.192842
subject             0.235427
sex                 0.245867
age                 1.537537
address             0.198656
famsize             0.207392
Pstatus             0.102566
Medu                1.265415
Fedu                1.209864
Mjob                1.691484
Fjob                1.155981
reason              1.063176
guardian            0.373976
traveltime          0.535425
studytime           0.696145
failures            0.430522
schoolsup           0.101089
famsup              0.237452
paid                0.166481
activities          0.250207
nursery             0.102566
higher              0.078056
internet            0.164809
romantic            0.229300
famrel              0.871237
freetime            1.064006
goout               1.328428
Dalc                0.831223
Walc                1.651494
health              2.029780
absences           38.564306
G1                  8.900639
G2                 10.791692
G3                 14.936647
avg_consumption     0.151438
dtype: float64
```

In [224]:
```
#Removing Unneccesary Features
df1 = df.drop(columns=['failures', 'studytime', 'sex', 'traveltime', 'traveltime', 'famsize','Medu', 'Fedu','higher','
```

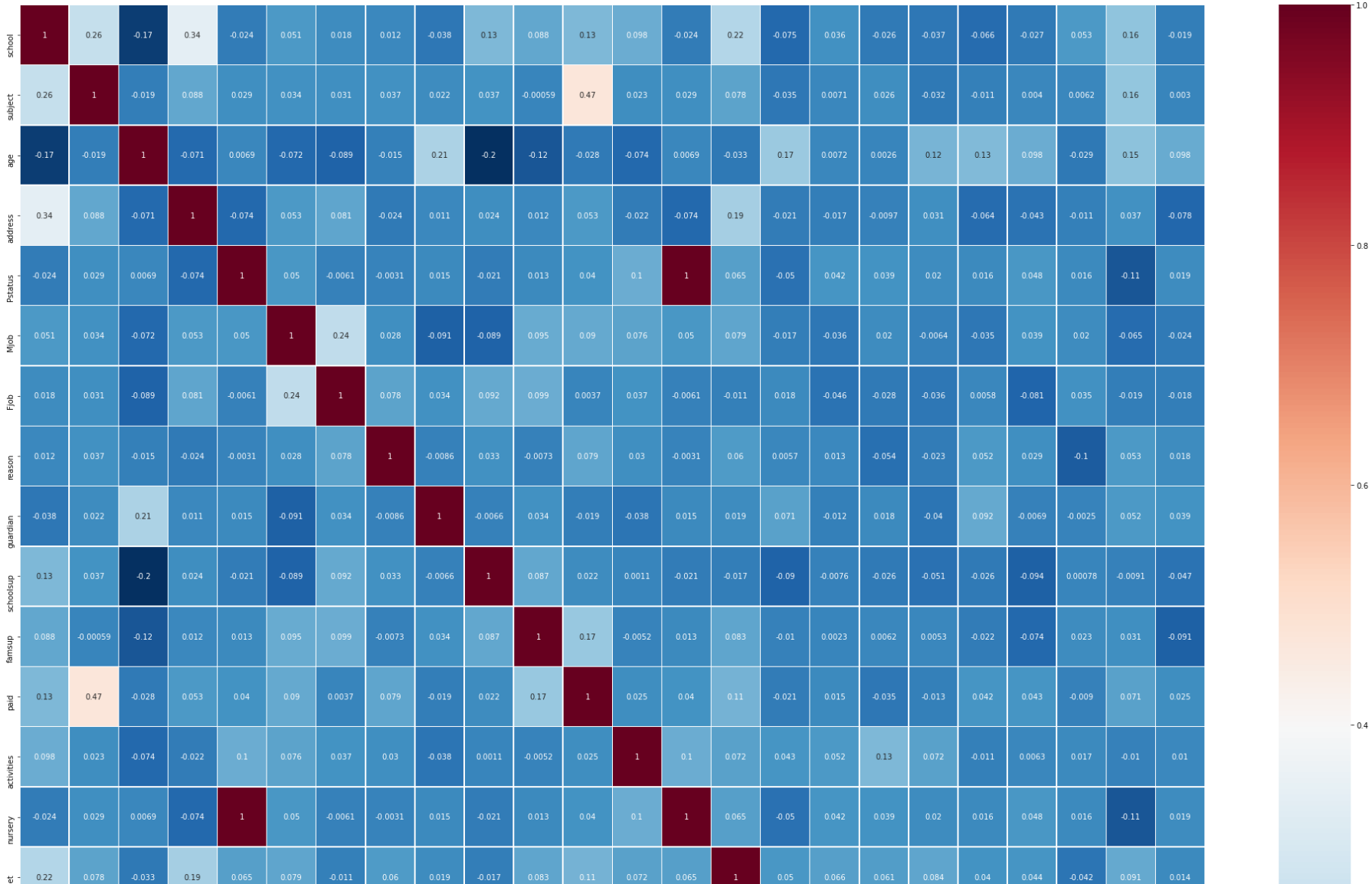
In [245]:
```
df1.var()
```

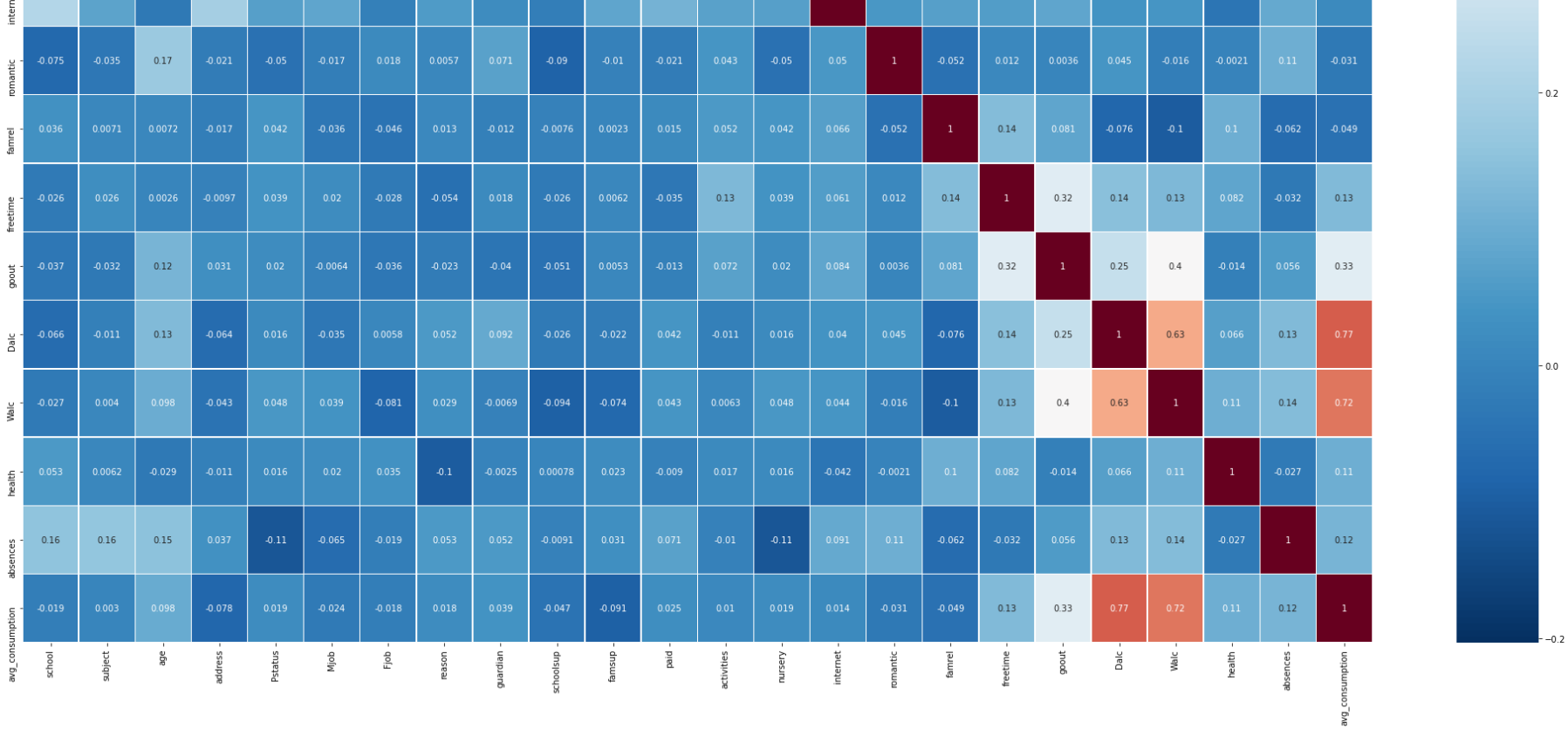
Out[245]:
```
school              0.192842
subject             0.235427
age                 1.537537
address             0.198656
Pstatus             0.102566
Mjob                1.691484
Fjob                1.155981
reason              1.063176
guardian            0.373976
schoolsup           0.101089
famsup              0.237452
paid                0.166481
activities          0.250207
nursery             0.102566
internet            0.164809
romantic            0.229300
famrel              0.871237
freetime            1.064006
goout               1.328428
Dalc                0.831223
Walc                1.651494
health              2.029780
absences           38.564306
avg_consumption     0.151438
dtype: float64
```

In [225]:

```
pearson = df1.corr(method = 'pearson')
#Final Pearson Correlation Heatmap
fig, ax = plt.subplots(figsize=(35,35))
sns.heatmap(pearson,
            xticklabels=pearson.columns,
            yticklabels=pearson.columns,
            cmap='RdBu_r',
            annot=True,
            linewidth= 0.5,
            annot_kws={"size": 10},
            ax=ax)
```

Out[225]: <matplotlib.axes._subplots.AxesSubplot at 0x2150bf91250>





Model Creation

In [226]:

```
#Load Model Libraries
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.inspection import permutation_importance
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,precision_score,recall_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import GridSearchCV
```

In [227]:

```
#Creating Data Variables
X = df1.drop('avg_consumption', axis=1)
y = df1['avg_consumption']
```

In [1]:

```
#Creating the Test, Train, and Split
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.2,random_state=40)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-d1bec3691419> in <module>
      1 #Creating the Test, Train, and Split
----> 2 X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.2,random_state=40)

NameError: name 'train_test_split' is not defined
```

In [229]:

```
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [230]:

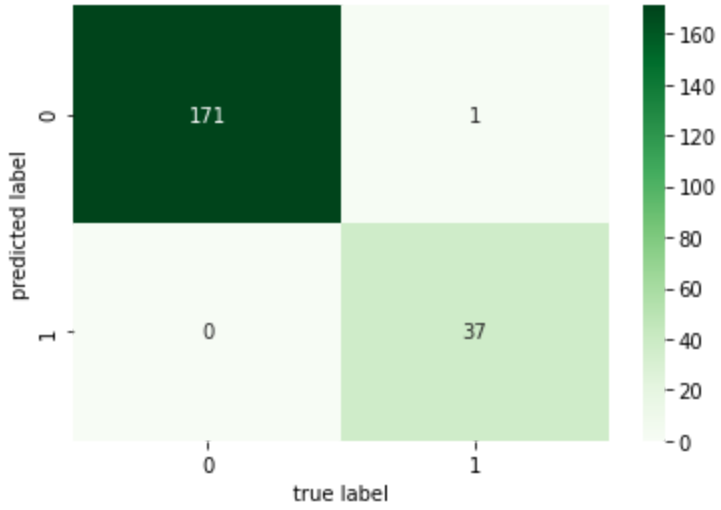
```
#Model
clf = RandomForestClassifier(n_estimators=20, random_state=40)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

Model Analysis

In [231]:

```
#confusion matrix
mat = confusion_matrix(y_test,y_pred)
sns.heatmap(mat.T, annot=True, fmt='d', cmap=plt.cm.Greens, cbar=True)
plt.xlabel('true label')
plt.ylabel('predicted label')
```

Out[231]: Text(33.0, 0.5, 'predicted label')



In [232]:

```
#Classification score
print(classification_report(y_test,y_pred.round()))
print("Total Accuracy:", accuracy_score(y_test, y_pred.round()))
```

```
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       171
           1       1.00      0.97      0.99        38

    accuracy                           1.00       209
   macro avg       1.00      0.99      0.99       209
weighted avg       1.00      1.00      1.00       209

Total Accuracy: 0.9952153110047847
```

In [233]:

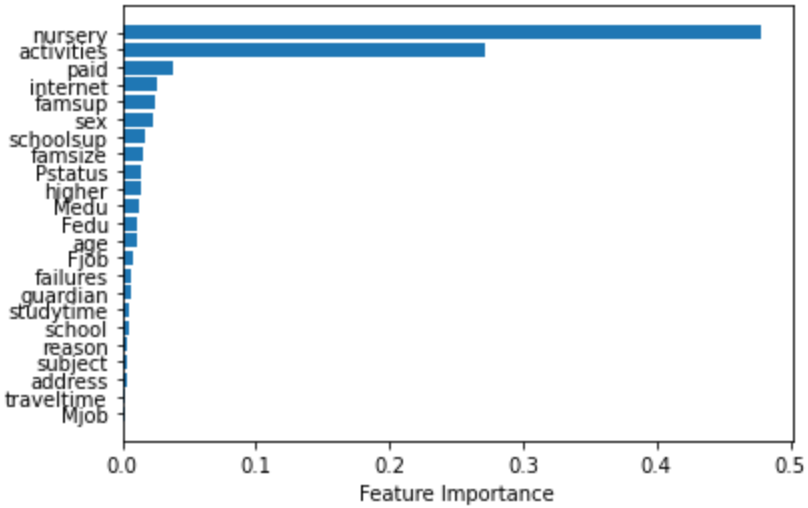
```python
#Compare Dataset Results
print('Training set metrics:')
print('Accuracy:', accuracy_score(y_train, clf.predict(X_train)))
print('Precision:', precision_score(y_train, clf.predict(X_train)))
print('Recall:', recall_score(y_train, clf.predict(X_train)))

print('-----------------------')

print('Test set metrics:')
print('Accuracy:', accuracy_score(y_test, clf.predict(X_test)))
print('Precision:', precision_score(y_test, clf.predict(X_test)))
print('Recall:', recall_score(y_test, clf.predict(X_test)))
```

```
Training set metrics:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
-----------------------
Test set metrics:
Accuracy: 0.9952153110047847
Precision: 1.0
Recall: 0.9736842105263158
```

In [234]:

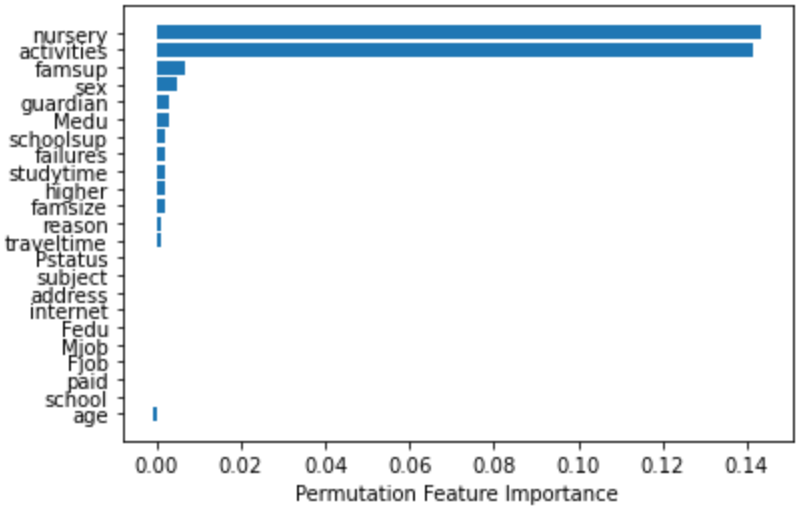
```
# Feature Importance
clf.feature_importances_
sorted_idx = clf.feature_importances_.argsort()
plt.barh(x1.columns[sorted_idx], clf.feature_importances_[sorted_idx])
plt.xlabel("Feature Importance")
```

Out[234]: Text(0.5, 0, 'Feature Importance')



In [235]:
```
# Permutation Feature Importance
perm_importance = permutation_importance(clf, X_test, y_test)
sorted_idx = perm_importance.importances_mean.argsort()
plt.barh(x1.columns[sorted_idx], perm_importance.importances_mean[sorted_idx])
plt.xlabel("Permutation Feature Importance")
```

Out[235]:
```
Text(0.5, 0, 'Permutation Feature Importance')
```



Standard Logistic Regression Model

In [236]:
```
#Load Library
from sklearn.linear_model import LogisticRegression

#Create Model
logmodel = LogisticRegression(solver='liblinear', max_iter=200, penalty='l2')
logmodel.fit(X_train,y_train)
predictions = logmodel.predict(X_test)
```

In [244]:

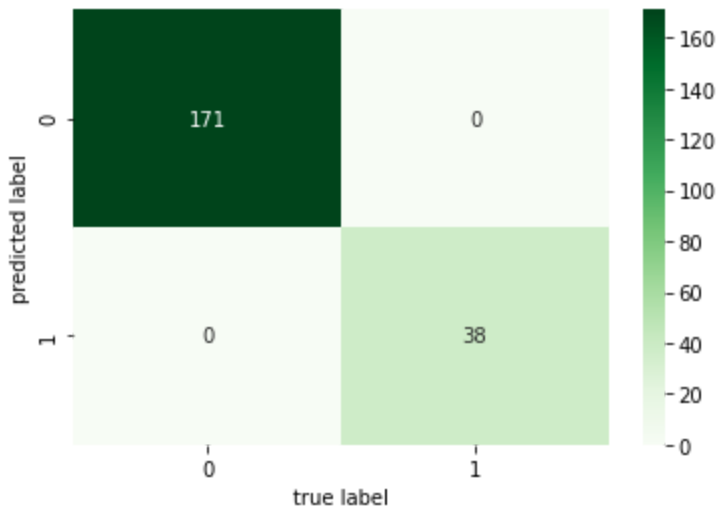
```
#Classification Report
print(classification_report(y_test,predictions))
print("Total Accuracy:", accuracy_score(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       171
           1       1.00      1.00      1.00        38

    accuracy                           1.00       209
   macro avg       1.00      1.00      1.00       209
weighted avg       1.00      1.00      1.00       209

Total Accuracy: 1.0
```

In [246]:

```
#Confusion Matrix
mat = confusion_matrix(y_test,predictions)
sns.heatmap(mat.T, annot=True, fmt='d', cmap=plt.cm.Greens, cbar=True)
plt.xlabel('true label')
plt.ylabel('predicted label')
```

Out[246]: Text(33.0, 0.5, 'predicted label')



In [ ]: