In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```python
df = pd.read_csv('st.csv')
```

In [3]:
```python
df.head()
```

Out[3]:

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

5 rows × 31 columns

In [4]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 31 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  health      395 non-null    int64 
 27  absences    395 non-null    int64 
 28  G1          395 non-null    int64 
 29  G2          395 non-null    int64 
 30  G3          395 non-null    int64 
dtypes: int64(14), object(17)
memory usage: 95.8+ KB
```

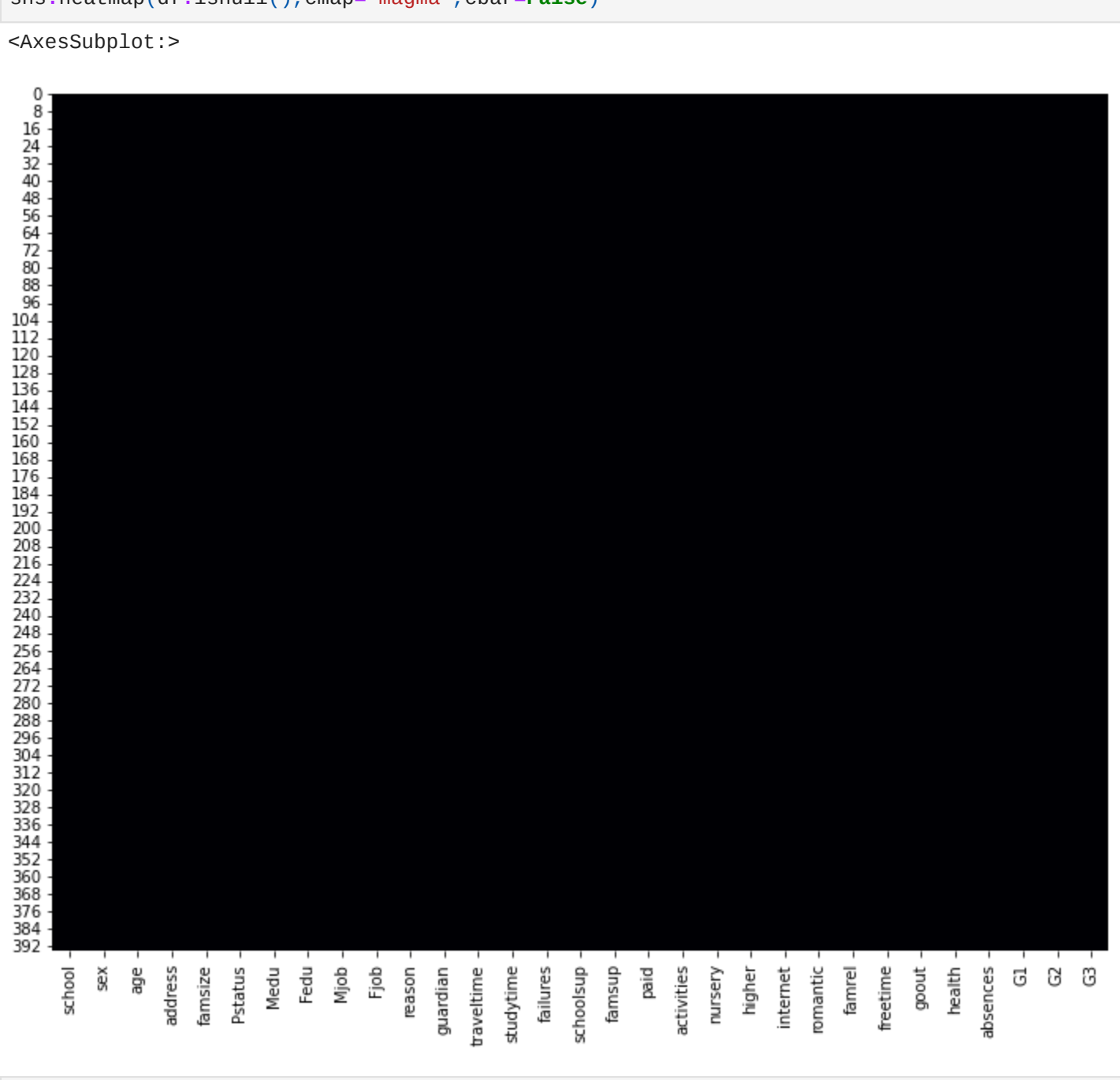
In [5]:
```python
df.describe()
```

Out[5]:

	age	Medu	Fedu	traveltime	studytime	failures	famrel	freetime	goout	health	absences	G1	G2	G3
count	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000
mean	16.696203	2.749367	2.521519	1.448101	2.035443	0.334177	3.944304	3.235443	3.108861	3.554430	5.708861	10.908861	10.713924	10.415190
std	1.276043	1.094735	1.088201	0.697505	0.839240	0.743651	0.896659	0.998862	1.113278	1.390303	8.003096	3.319195	3.761505	4.581443
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	0.000000	3.000000	0.000000	0.000000
25%	16.000000	2.000000	2.000000	1.000000	1.000000	0.000000	4.000000	3.000000	2.000000	3.000000	0.000000	8.000000	9.000000	8.000000
50%	17.000000	3.000000	2.000000	1.000000	2.000000	0.000000	4.000000	3.000000	3.000000	4.000000	4.000000	11.000000	11.000000	11.000000
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.000000	5.000000	4.000000	4.000000	5.000000	8.000000	13.000000	13.000000	14.000000
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.000000	5.000000	5.000000	5.000000	5.000000	75.000000	19.000000	19.000000	20.000000

In [6]:
```python
plt.figure(figsize= (12,10))
sns.heatmap(df.isnull(),cmap='magma',cbar=False)
```

Out[6]: <AxesSubplot:>



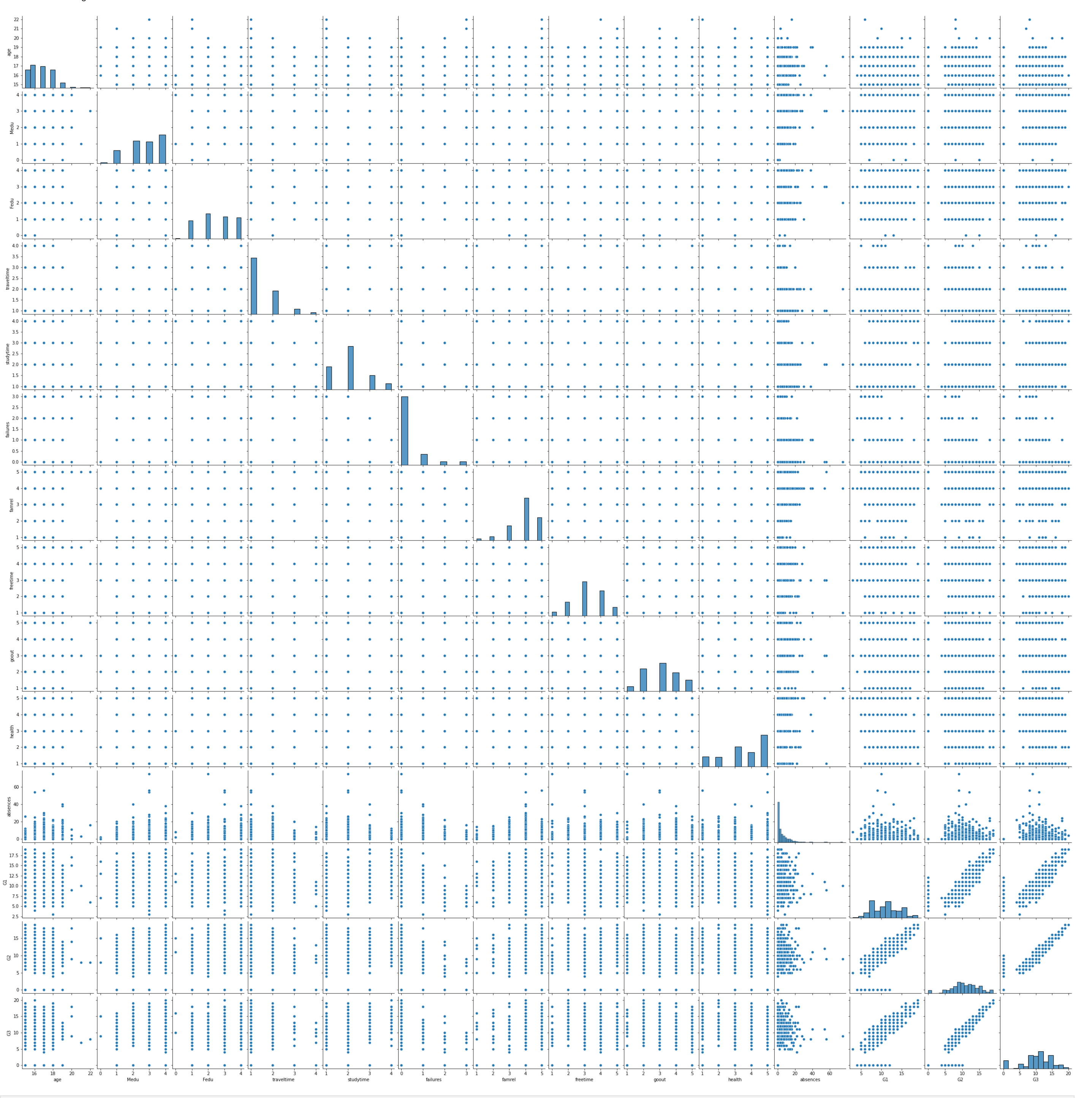
In [7]:
```python
df.isnull().sum()
```

Out[7]:

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

In [8]:
```python
sns.pairplot(df)
```

Out[8]: <seaborn.axisgrid.PairGrid at 0x201c5528c10>



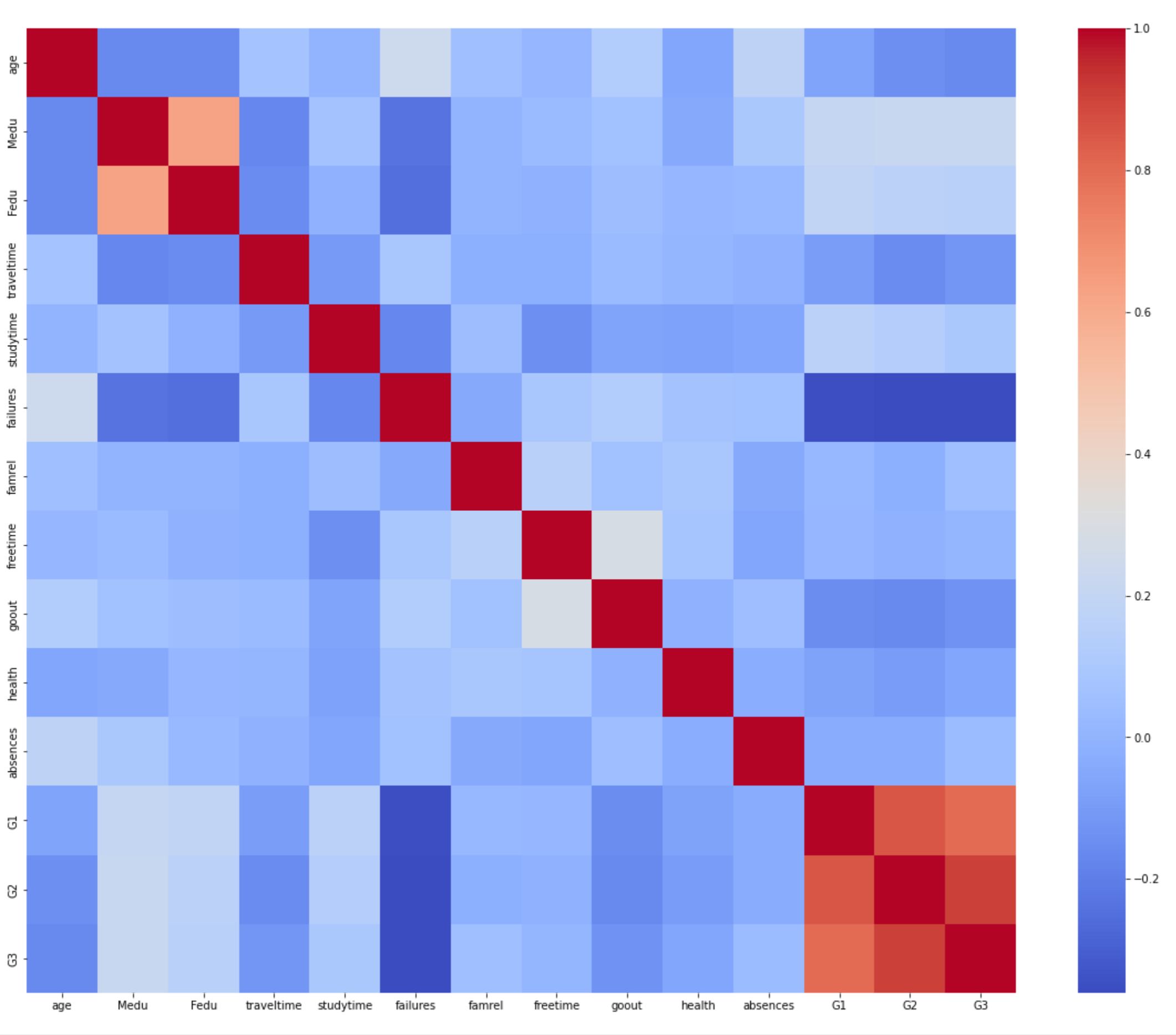
In [9]:
```python
correlation = df.corr()
correlation
```

Out[9]:

	age	Medu	Fedu	traveltime	studytime	failures	famrel	freetime	goout	health	absences	G1	G2	G3
age	1.000000	-0.163658	-0.163438	0.070641	-0.004140	0.243665	0.053940	0.016434	0.126964	-0.062187	0.175230	-0.064081	-0.143474	-0.161579
Medu	-0.163658	1.000000	0.623455	-0.171639	0.064944	-0.236680	-0.003914	0.030891	0.064094	-0.046878	0.100285	0.205341	0.215527	0.217147
Fedu	-0.163438	0.623455	1.000000	-0.158194	-0.009175	-0.250408	-0.001370	-0.012846	0.043105	0.014742	0.024473	0.190270	0.164893	0.152457
traveltime	0.070641	-0.171639	-0.158194	1.000000	-0.100909	0.092239	-0.016808	-0.017025	0.028540	0.007501	-0.012944	-0.093040	-0.153198	-0.117142
studytime	-0.004140	0.064944	-0.009175	-0.100909	1.000000	-0.173563	0.039731	-0.143198	-0.063904	-0.075616	-0.062700	0.160612	0.135880	0.097820
failures	0.243665	-0.236680	-0.250408	0.092239	-0.173563	1.000000	-0.044337	0.091987	0.124561	0.065827	0.063726	-0.354718	-0.355896	-0.360415
famrel	0.053940	-0.003914	-0.001370	-0.016808	0.039731	-0.044337	1.000000	0.150701	0.064568	0.094056	-0.044354	0.022168	-0.018281	0.051363
freetime	0.016434	0.030891	-0.012846	-0.017025	-0.143198	0.091987	0.150701	1.000000	0.285019	0.075733	-0.058078	0.012613	-0.013777	0.011307
goout	0.126964	0.064094	0.043105	0.028540	-0.063904	0.124561	0.064568	0.285019	1.000000	-0.009577	0.044302	-0.149104	-0.162250	-0.132791
health	-0.062187	-0.046878	0.014742	0.007501	-0.075616	0.065827	0.094056	0.075733	-0.009577	1.000000	-0.029937	-0.073172	-0.097720	-0.061335
absences	0.175230	0.100285	0.024473	-0.012944	-0.062700	0.063726	-0.044354	-0.058078	0.044302	-0.029937	1.000000	-0.031003	-0.031777	0.034247
G1	-0.064081	0.205341	0.190270	-0.093040	0.160612	-0.354718	0.022168	0.012613	-0.149104	-0.073172	-0.031003	1.000000	0.852118	0.801468
G2	-0.143474	0.215527	0.164893	-0.153198	0.135880	-0.355896	-0.018281	-0.013777	-0.162250	-0.097720	-0.031777	0.852118	1.000000	0.904868
G3	-0.161579	0.217147	0.152457	-0.117142	0.097820	-0.360415	0.051363	0.011307	-0.132791	-0.061335	0.034247	0.801468	0.904868	1.000000

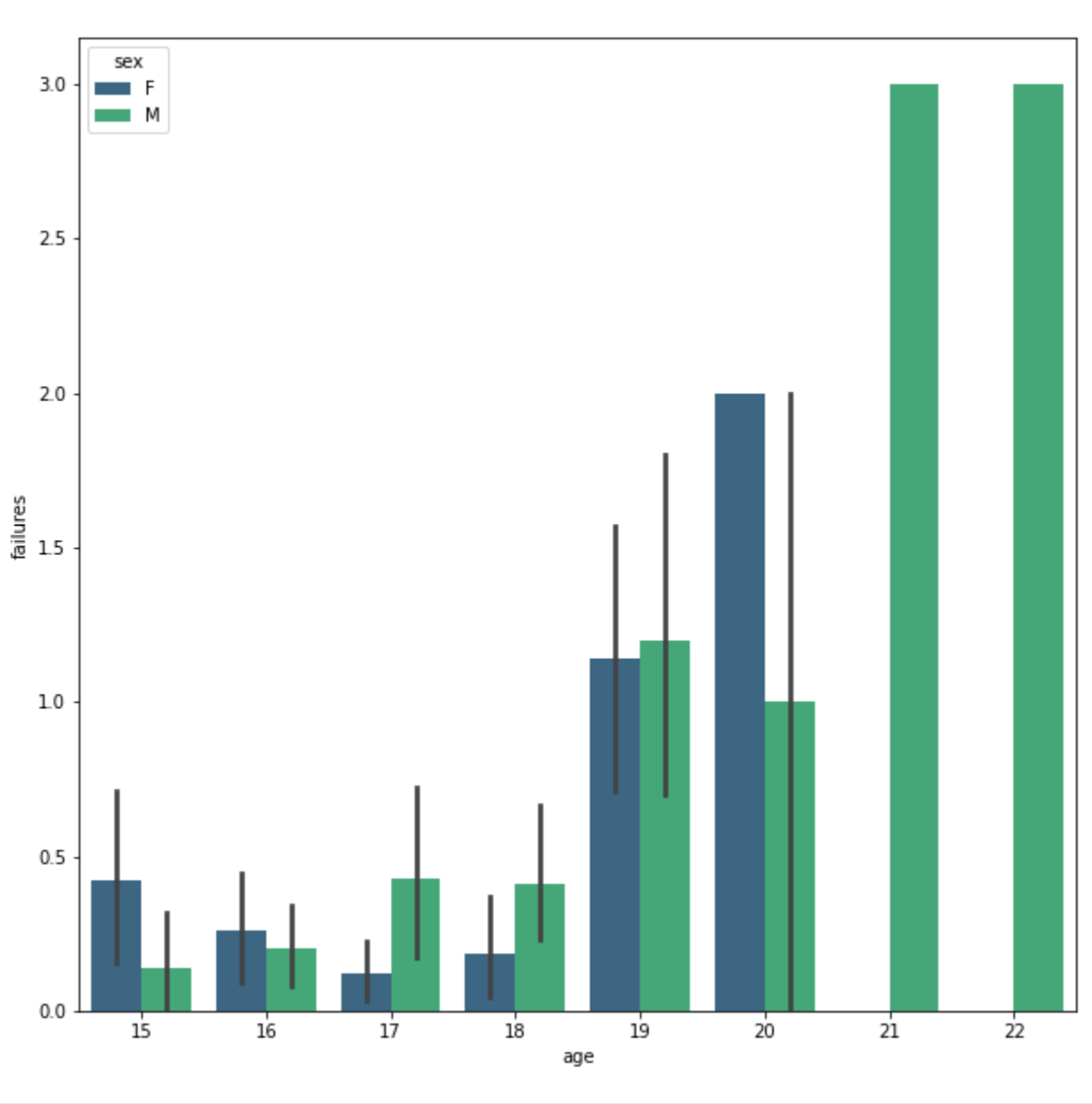
In [10]:
```python
plt.figure(figsize= (20,16))
sns.heatmap(correlation, cmap= 'coolwarm')
```

Out[10]: <AxesSubplot:>



In [11]:
```python
plt.figure(figsize=(10,10))
sns.barplot(df['age'],df['failures'],hue= df['sex'],palette='viridis')
```

Out[11]: <AxesSubplot:xlabel='age', ylabel='failures'>



In [12]:
```python
from sklearn.model_selection import train_test_split
```

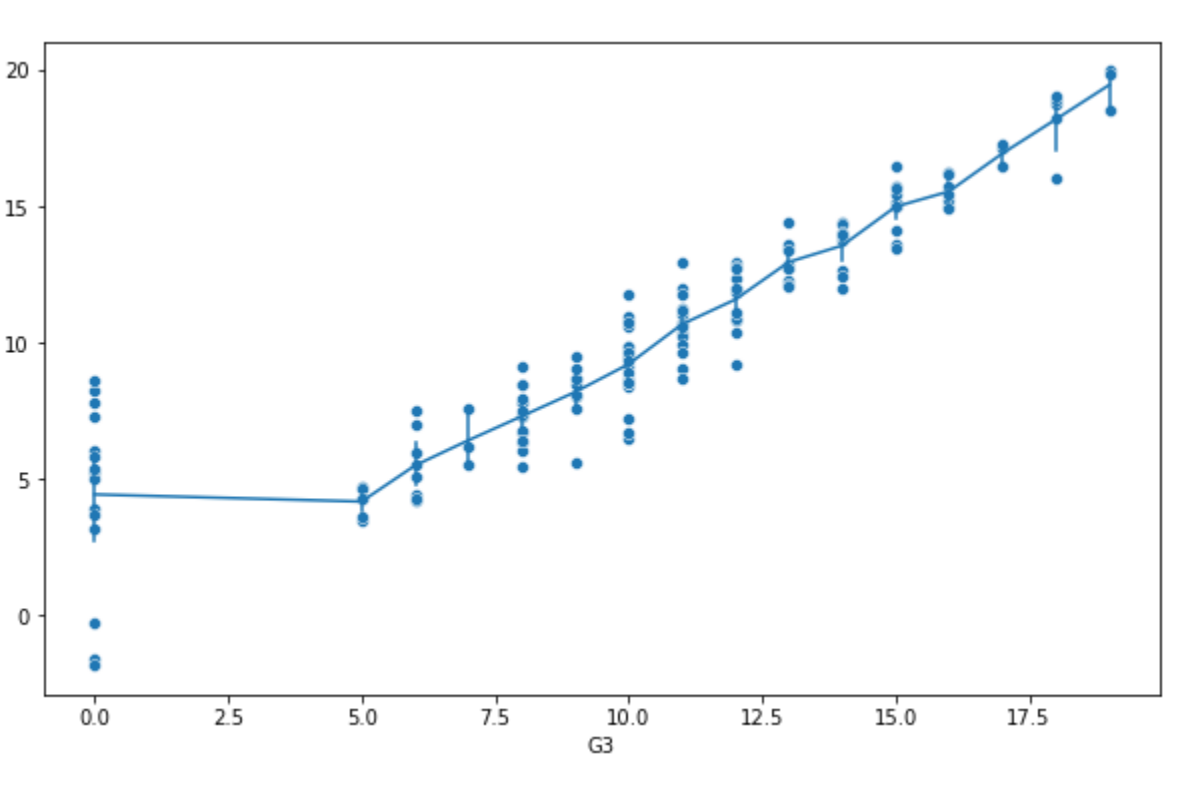
In [13]:
```python
#independent Variable
X = df[['studytime','failures', 'absences', 'G1', 'G2']]

#dependent variable
y = df['G3']

#Train Test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state= 42)
```

In [14]:
```python
from sklearn.linear_model import LinearRegression

lr = LinearRegression(normalize = True)

lr.fit(X_train,y_train)

our_pred = lr.predict(X_test)

lr.intercept_
```

Out[14]: -1.3499391057720995

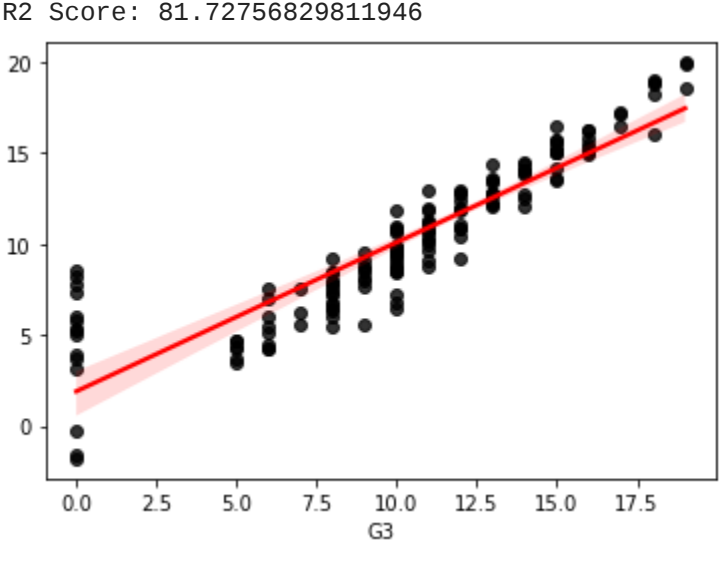
In [15]:
```python
plt.figure(figsize= (10,6))
sns.lineplot(y_test,our_pred, err_style="bars")
sns.scatterplot(y_test, our_pred)
```

Out[15]: <AxesSubplot:xlabel='G3'>



In [16]:
```python
sns.regplot(y_test,our_pred,scatter_kws={"color": "black"}, line_kws={"color": "red"})

from sklearn import metrics
from sklearn.metrics import r2_score

print('MSE', metrics.mean_squared_error(y_test,our_pred))
print('R2 Score:',r2_score(y_test , our_pred)*100)
```

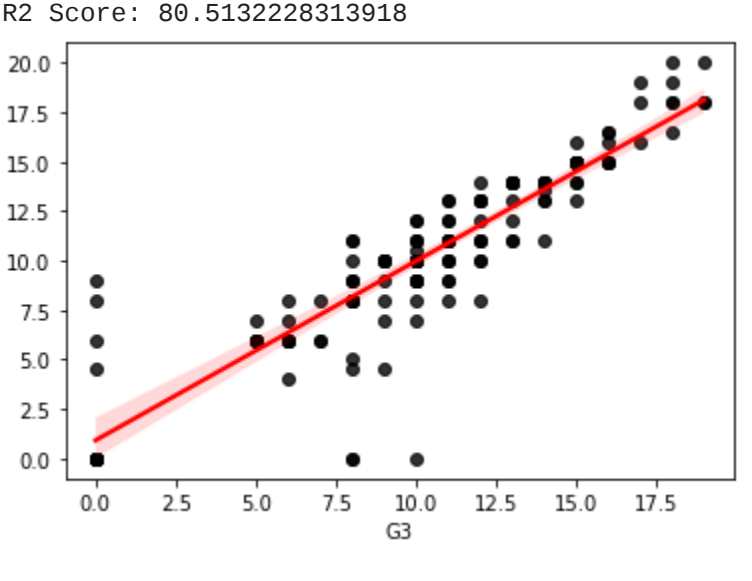
```
MSE 4.011861551818311
R2 Score: 81.72756829811946
```



In [17]:
```python
from sklearn.tree import DecisionTreeRegressor

dt = DecisionTreeRegressor()
dt.fit(X_train,y_train)
pred_t = dt.predict(X_test)
```

In [18]:
```python
sns.regplot(y_test,pred_t,scatter_kws={"color": "black"}, line_kws={"color": "red"})
print('MSE', metrics.mean_squared_error(y_test,pred_t))
print('R2 Score:',r2_score(y_test , pred_t)*100)
```

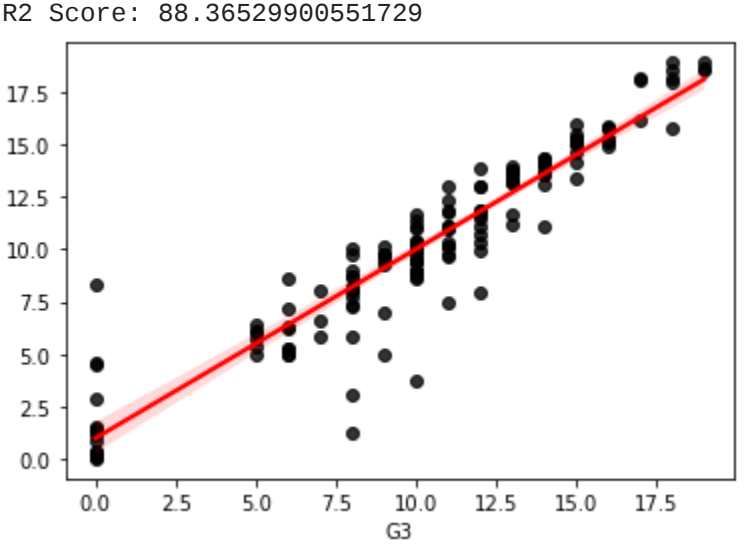
```
MSE 4.2784810126582276
R2 Score: 80.5132228313918
```



In [19]:
```python
from sklearn.ensemble import RandomForestRegressor

rf = RandomForestRegressor()
rf.fit(X_train,y_train)
rf_pred = rf.predict(X_test)
```

In [20]:
```python
sns.regplot(y_test,rf_pred,scatter_kws={"color": "black"}, line_kws={"color": "red"})
print('MSE', metrics.mean_squared_error(y_test,rf_pred))
print('R2 Score:',r2_score(y_test , rf_pred)*100)
```

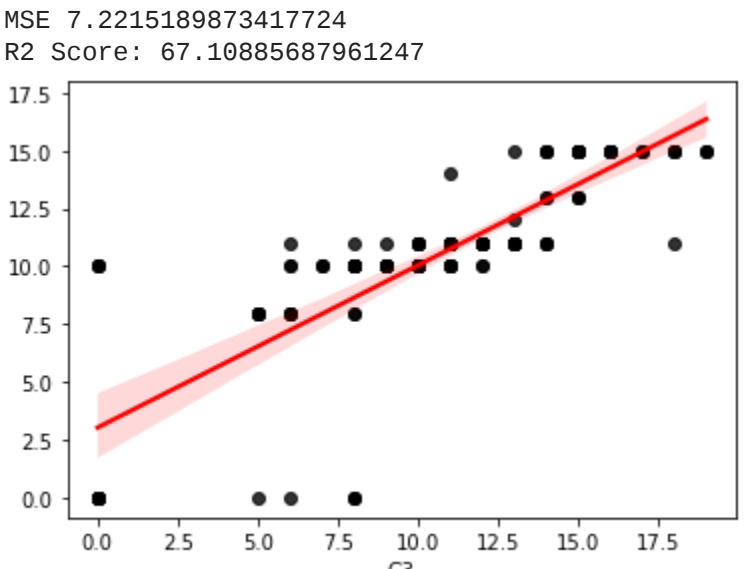
```
MSE 2.554493586196504
R2 Score: 88.36529900551729
```



In [21]:
```python
from sklearn.svm import SVC
svc = SVC()
svc.fit(X_train, y_train)
pred_svm = svc.predict(X_test)
```

In [22]:
```python
sns.regplot(y_test,pred_svm,scatter_kws={"color": "black"}, line_kws={"color": "red"})
print('MSE', metrics.mean_squared_error(y_test,pred_svm))
print('R2 Score:',r2_score(y_test , pred_svm)*100)
```

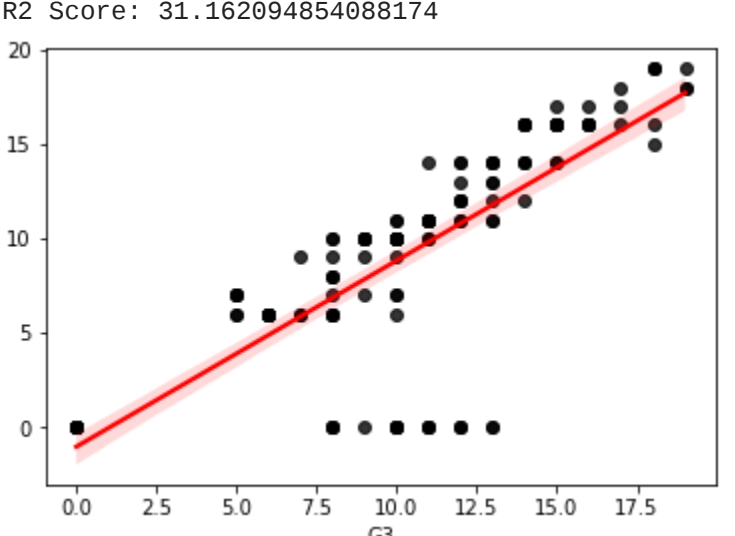
```
MSE 7.2215189873417724
R2 Score: 67.10885687961247
```



In [23]:
```python
from sklearn.naive_bayes import GaussianNB
gaussian = GaussianNB()
gaussian.fit(X_train, y_train)
pred_gb = gaussian.predict(X_test)
```

In [24]:
```python
sns.regplot(y_test,pred_gb,scatter_kws={"color": "black"}, line_kws={"color": "red"})
print('MSE', metrics.mean_squared_error(y_test,pred_gb))
print('R2 Score:',r2_score(y_test , pred_gb)*100)
```

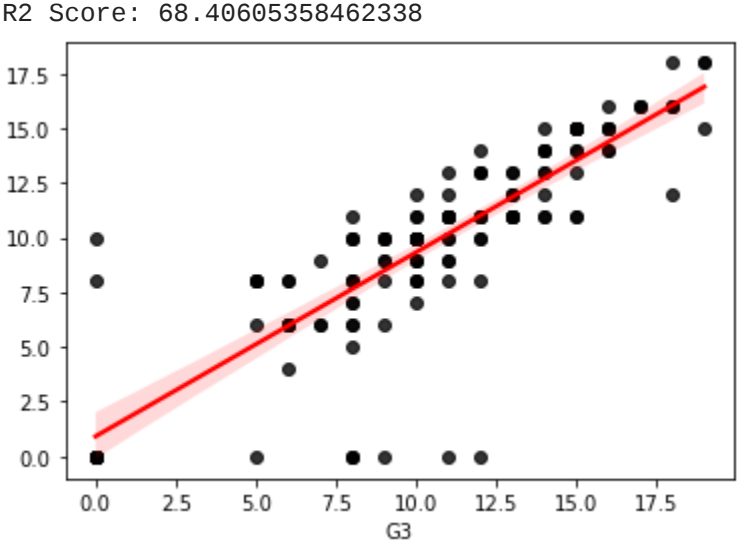
```
MSE 15.113924050632912
R2 Score: 31.162094854088174
```



In [25]:
```python
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 3)
knn.fit(X_train, y_train)
pred_knn = knn.predict(X_test)
```

In [26]:
```python
sns.regplot(y_test,pred_knn,scatter_kws={"color": "black"}, line_kws={"color": "red"})
print('MSE', metrics.mean_squared_error(y_test,pred_knn))
print('R2 Score:',r2_score(y_test , pred_knn)*100)
```

```
MSE 6.936708860759493
R2 Score: 68.40605358462338
```



In [27]:
```python
#Linear Regression
print('Linear Regression R2 Score:',r2_score(y_test , our_pred)*100)
print("\n")
#Decision Tree
print('Decision Tree R2 Score:',r2_score(y_test , pred_t)*100)
print("\n")
#Random Forest
print('Random Forest R2 Score:',r2_score(y_test , rf_pred)*100)
print("\n")
#SVM
print('SVM R2 Score:',r2_score(y_test , pred_svm)*100)
print("\n")
#Naive Bayes
print('Naive Bayes  R2 Score:',r2_score(y_test , pred_gb)*100)
print("\n")
#K- Nearest Neighbour
print('K- Nearest Neighbour R2 Score:',r2_score(y_test , pred_knn)*100)
print("\n")
```

```
Linear Regression R2 Score: 81.72756829811946


Decision Tree R2 Score: 80.5132228313918


Random Forest R2 Score: 88.36529900551729


SVM R2 Score: 67.10885687961247


Naive Bayes  R2 Score: 31.162094854088174


K- Nearest Neighbour R2 Score: 68.40605358462338
```