Data Analysis

In [1]:

```
from IPython.display import Image
Image("Desktop/1427972_0.jpg")
```

Out[1]:

In [78]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Set the styles to Seaborn
sns.set()

# Import the KMeans module so we can perform k-means clustering with sklearn
from sklearn.cluster import KMeans
```

In [133]:

```
df=pd.read_csv('Desktop/student_data.csv')
df.head()
```

Out[133]:

```
   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 [134]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   school      395 non-null    object
 1   sex         395 non-null    object
 2   age         395 non-null    int64 
 3   address     395 non-null    object
 4   famsize     395 non-null    object
 5   Pstatus     395 non-null    object
 6   Medu        395 non-null    int64 
 7   Fedu        395 non-null    int64 
 8   Mjob        395 non-null    object
 9   Fjob        395 non-null    object
 10  reason      395 non-null    object
 11  guardian    395 non-null    object
 12  traveltime  395 non-null    int64 
 13  studytime   395 non-null    int64 
 14  failures    395 non-null    int64 
 15  schoolsup   395 non-null    object
 16  famsup      395 non-null    object
 17  paid        395 non-null    object
 18  activities  395 non-null    object
 19  nursery     395 non-null    object
 20  higher      395 non-null    object
 21  internet    395 non-null    object
 22  romantic    395 non-null    object
 23  famrel      395 non-null    int64 
 24  freetime    395 non-null    int64 
 25  goout       395 non-null    int64 
 26  Dalc        395 non-null    int64 
 27  Walc        395 non-null    int64 
 28  health      395 non-null    int64 
 29  absences    395 non-null    int64 
 30  G1          395 non-null    int64 
 31  G2          395 non-null    int64 
 32  G3          395 non-null    int64 
dtypes: int64(16), object(17)
memory usage: 75.7+ KB
```

In [135]:
```
df.describe()
```

Out[135]:

	age	Medu	Fedu	traveltime	studytime	failures	famrel	fr
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 [136]:
```
df['studytime'].mean()
```

Out[136]:
```
2.0354430379746837
```

In [137]:
```
df['studytime'].median()
```

Out[137]:
```
2.0
```

In [138]:
```
df['studytime'].var()
```

Out[138]:
```
0.704324359056738
```

In [139]:
```
df['studytime'].mode()
```

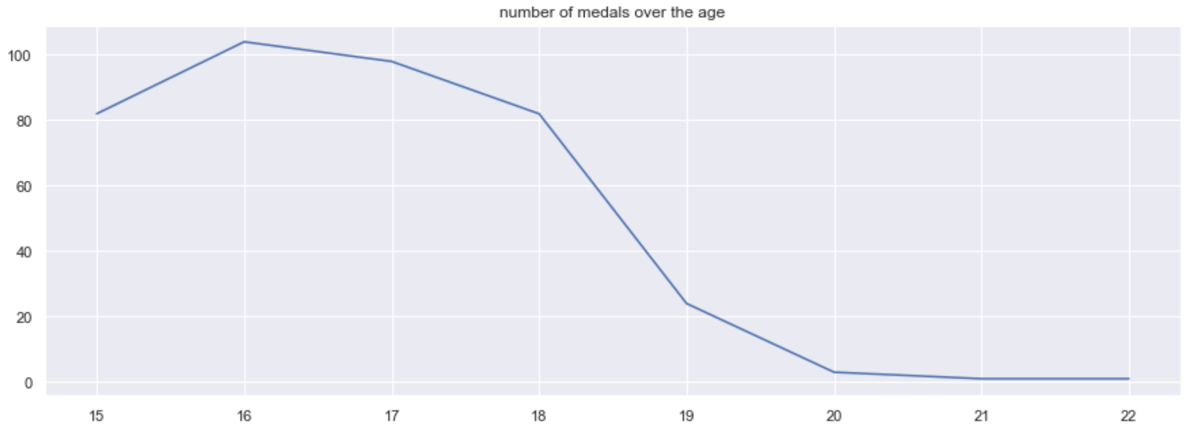
Out[139]:
```
0    2
Name: studytime, dtype: int64
```

In [140]:
```
df['studytime'].std()
```

Out[140]:
```
0.839240346418556
```

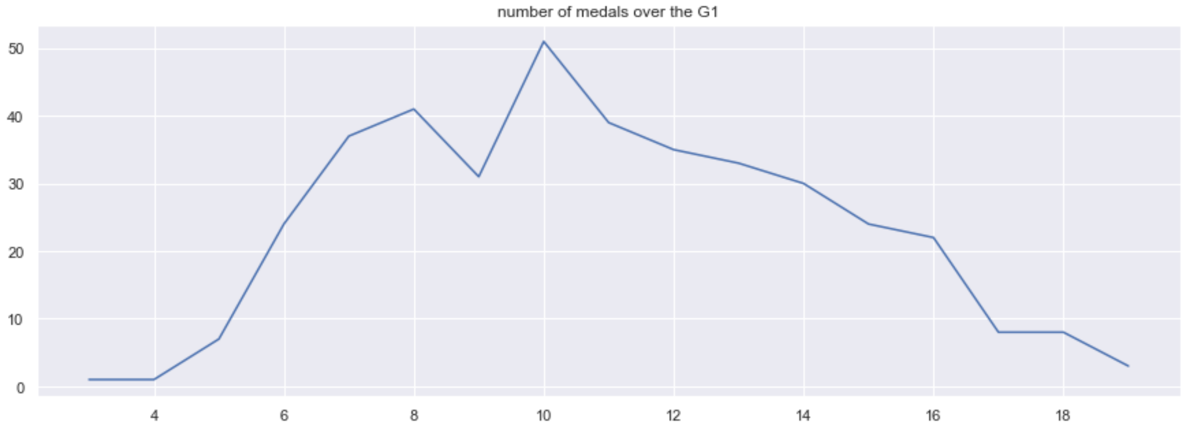
```
In [161]: plt.figure(figsize=(15,5))
          plt.title('number of medals over the age')
          df.age.value_counts().sort_index().plot()

Out[161]: <AxesSubplot:title={'center':'number of medals over the age'}>
```



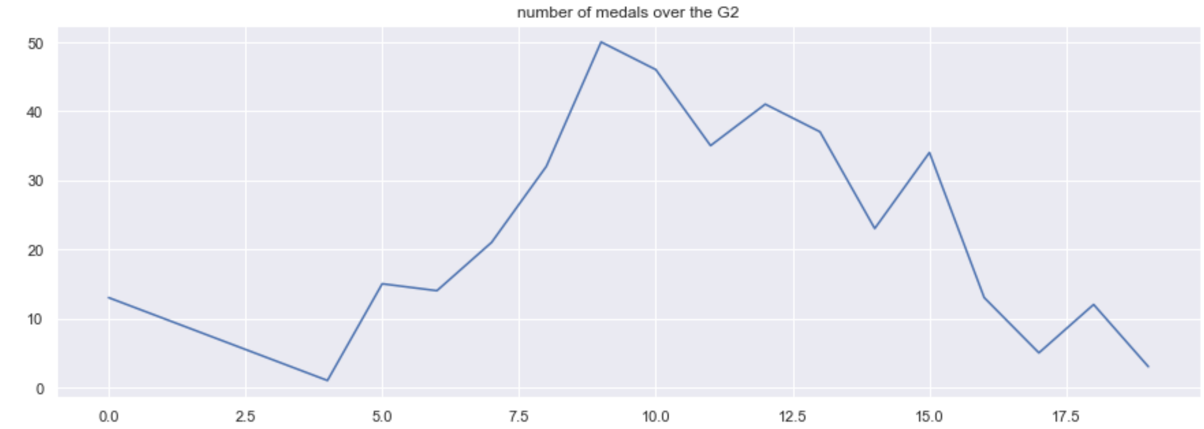
```
In [160]: plt.figure(figsize=(15,5))
          plt.title('number of medals over the G1')
          df.G1.value_counts().sort_index().plot()

Out[160]: <AxesSubplot:title={'center':'number of medals over the G1'}>
```



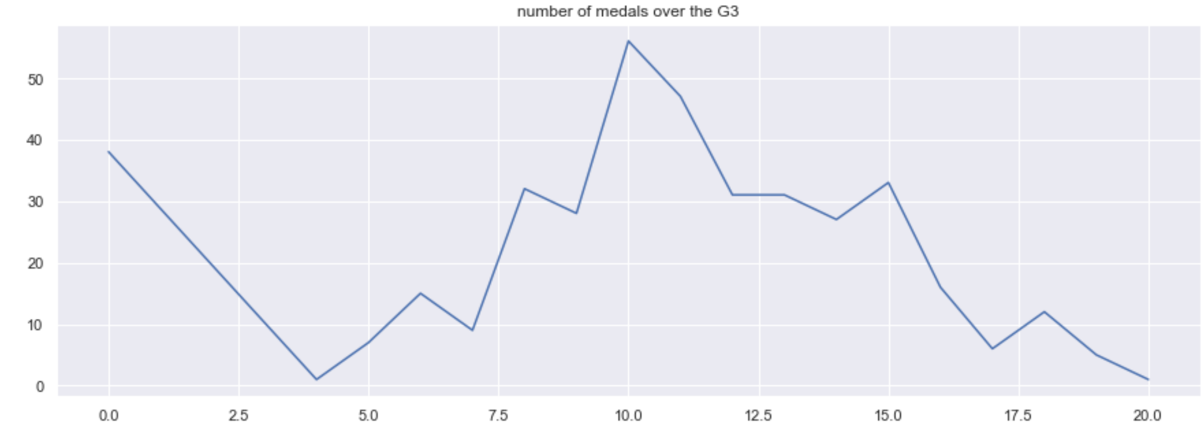
In [159]:
```
plt.figure(figsize=(15,5))
plt.title('number of medals over the G2')
df.G2.value_counts().sort_index().plot()
```

Out[159]:
```
<AxesSubplot:title={'center':'number of medals over the G2'}>
```



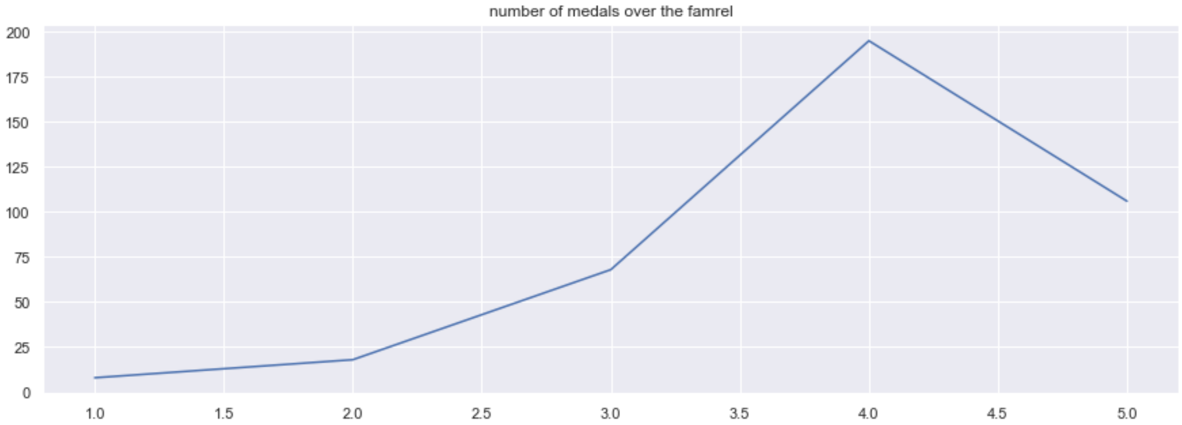
In [158]:
```
plt.figure(figsize=(15,5))
plt.title('number of medals over the G3')
df.G3.value_counts().sort_index().plot()
```

Out[158]:
```
<AxesSubplot:title={'center':'number of medals over the G3'}>
```



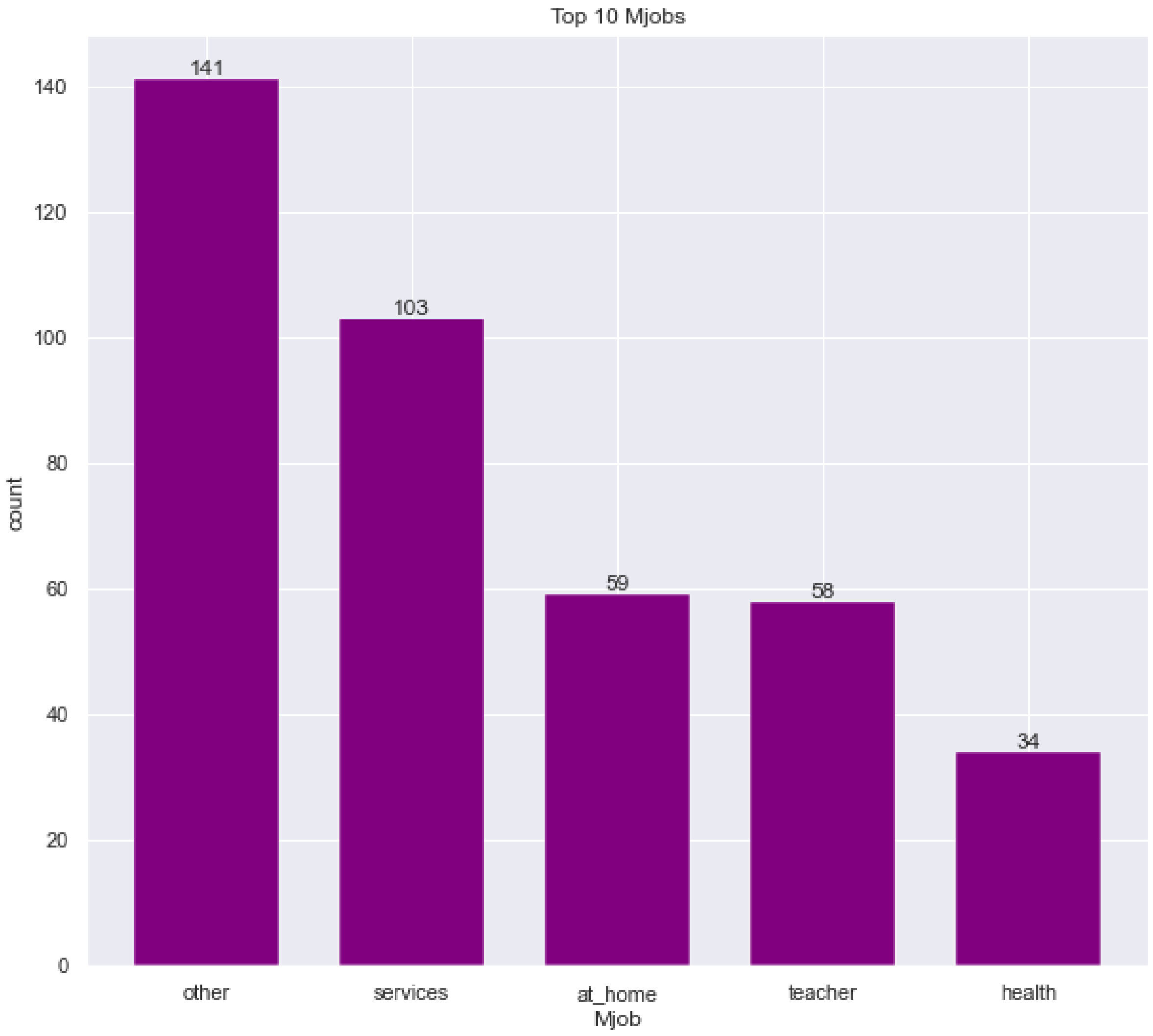
In [157]:
```
plt.figure(figsize=(15,5))
plt.title('number of medals over the famrel')
df.famrel.value_counts().sort_index().plot()
```

Out[157]:
```
<AxesSubplot:title={'center':'number of medals over the famrel'}>
```



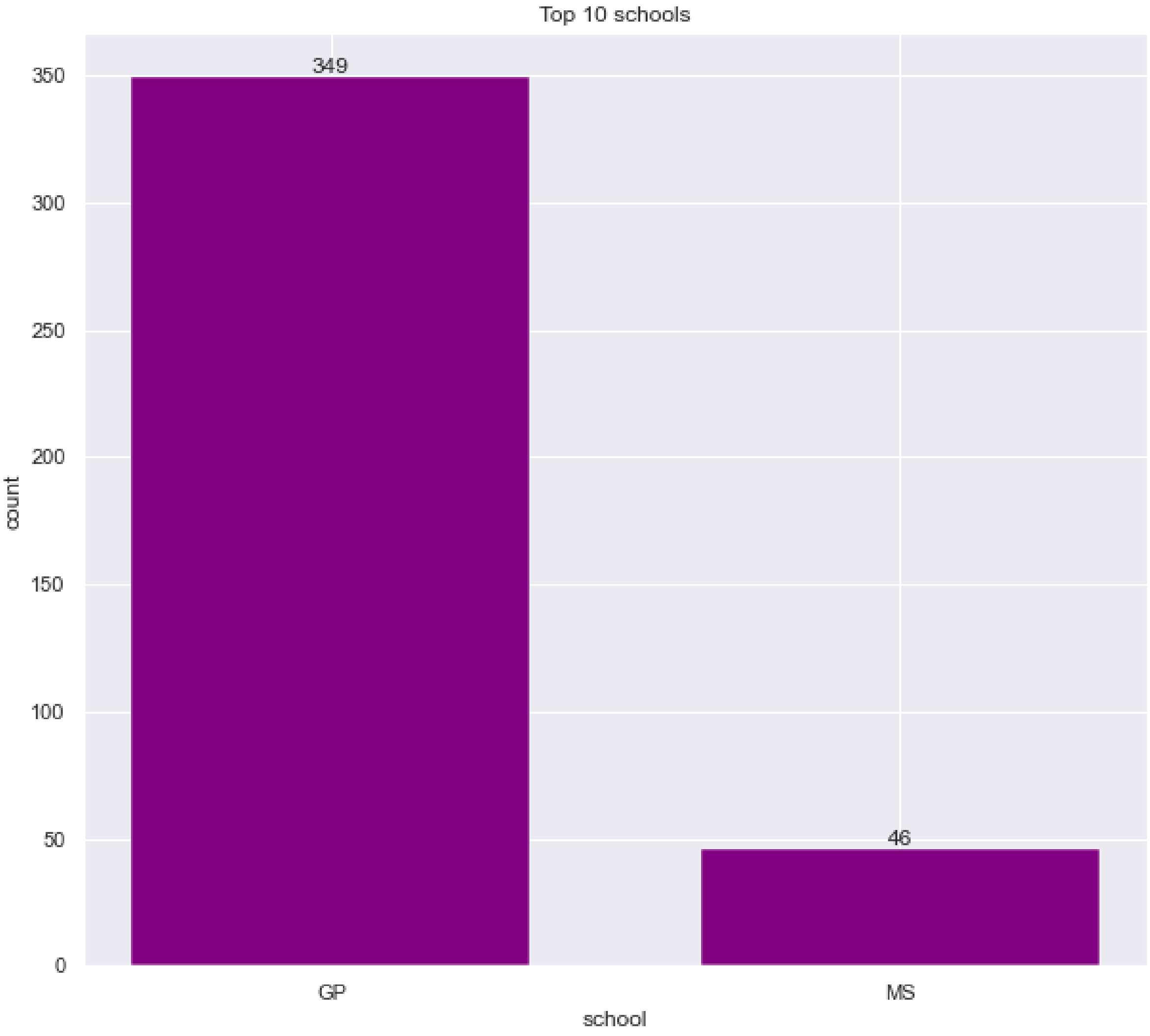
In [243]:

```
itemNames = df['Mjob'].value_counts().index[:10]
itemValues = df['Mjob'].value_counts().values[:10]
plt.figure(figsize=(10,9))
plt.ylabel('count', fontsize='medium')
plt.xlabel('Mjob', fontsize='medium')
plt.title('Top 10 Mjobs')
plt.bar(itemNames,itemValues, width = 0.7,color='purple',linewidth=0.4)
for i in range(len(itemNames)):
    plt.text(i,itemValues[i],itemValues[i],ha='center',va='bottom')
plt.show()
```

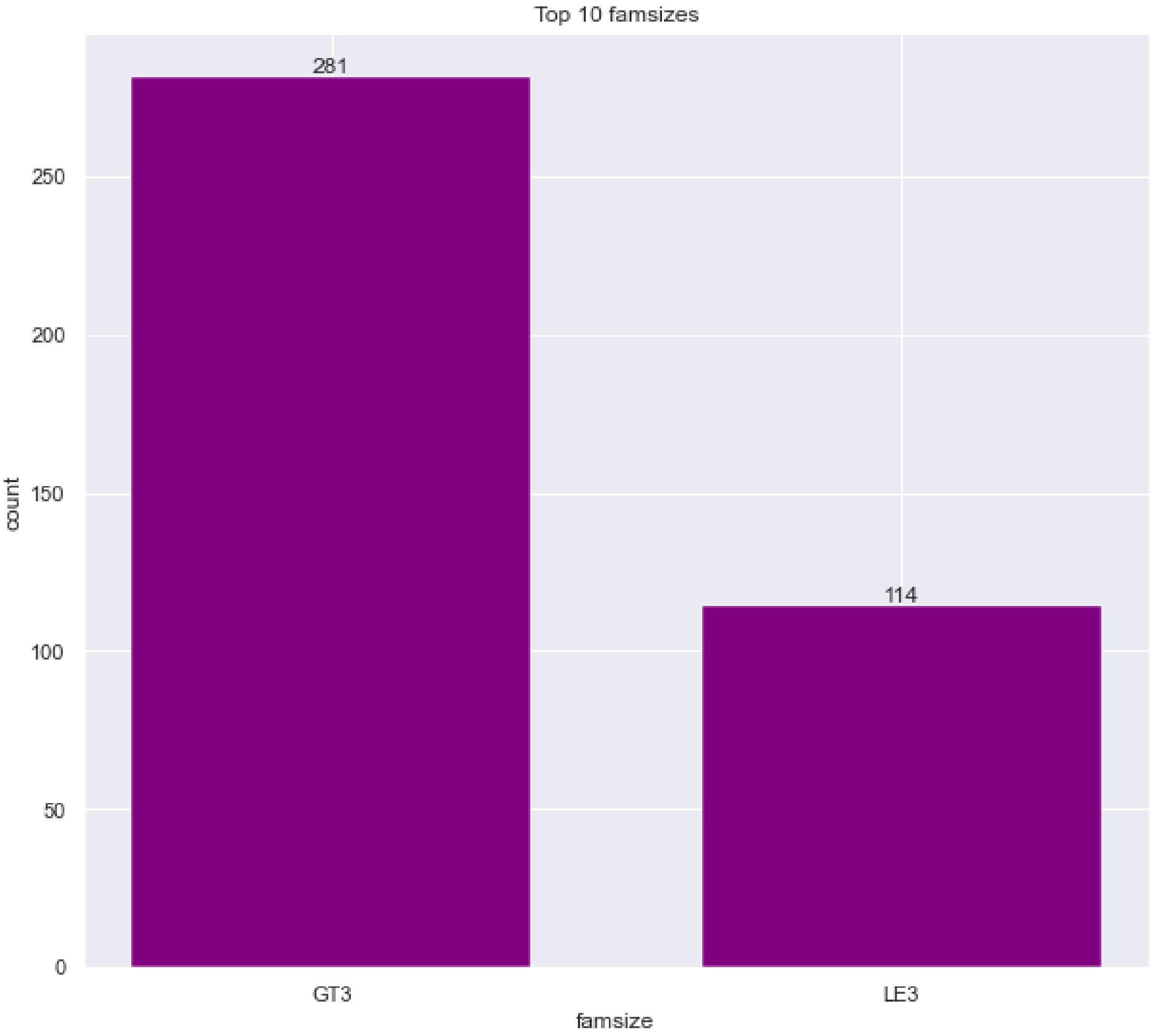


In [241]:

```
itemNames = df['school'].value_counts().index[:10]
itemValues = df['school'].value_counts().values[:10]
plt.figure(figsize=(10,9))
plt.ylabel('count', fontsize='medium')
plt.xlabel('school', fontsize='medium')
plt.title('Top 10 schools')
plt.bar(itemNames,itemValues, width = 0.7,color='purple',linewidth=0.4)
for i in range(len(itemNames)):
    plt.text(i,itemValues[i],itemValues[i],ha='center',va='bottom')
plt.show()
```

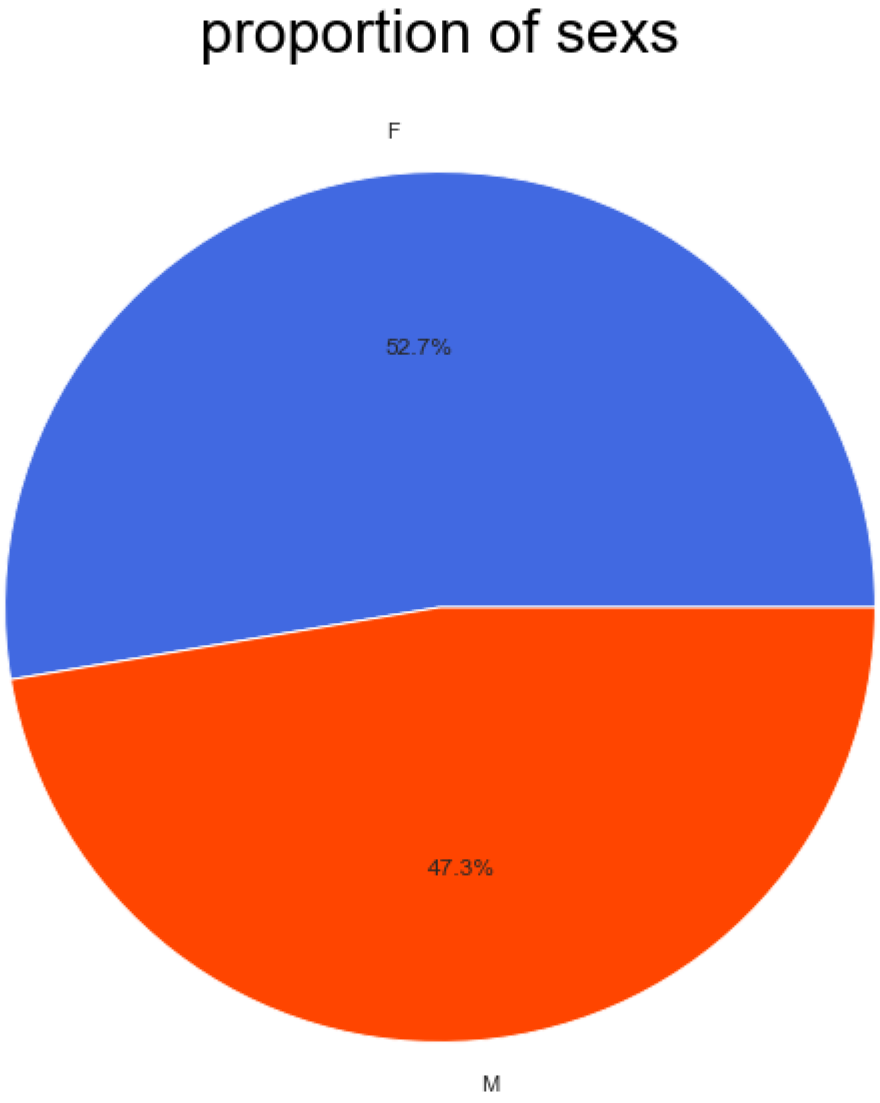


In [242]:
```
itemNames = df['famsize'].value_counts().index[:10]
itemValues = df['famsize'].value_counts().values[:10]
plt.figure(figsize=(10,9))
plt.ylabel('count', fontsize='medium')
plt.xlabel('famsize', fontsize='medium')
plt.title('Top 10 famsizes')
plt.bar(itemNames,itemValues, width = 0.7,color='purple',linewidth=0.4)
for i in range(len(itemNames)):
    plt.text(i,itemValues[i],itemValues[i],ha='center',va='bottom')
plt.show()
```



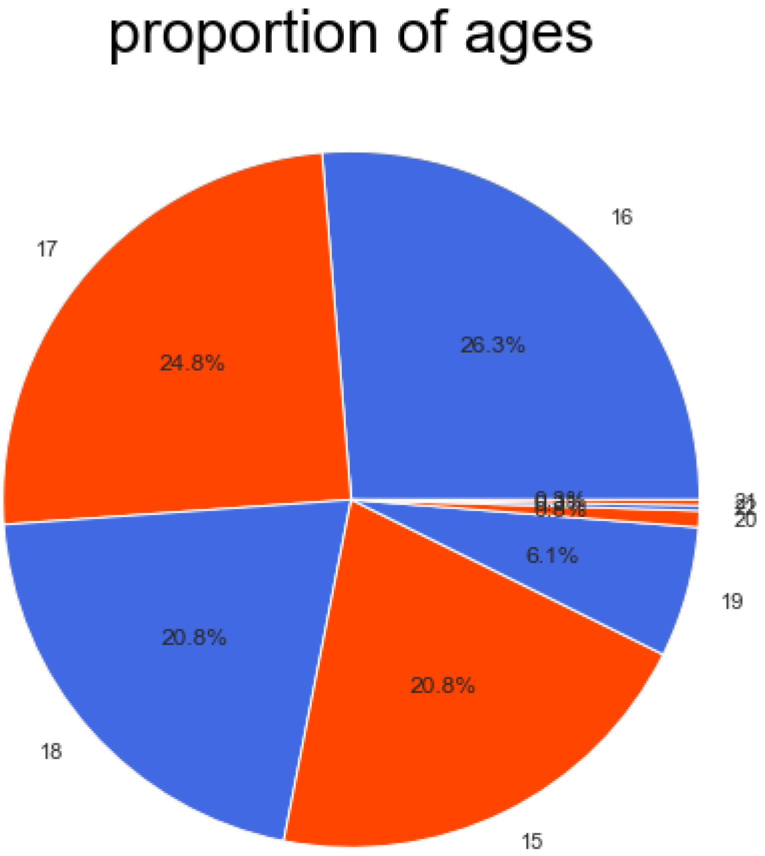
In [166]:
```
labels = df.sex.value_counts().index
colors = ['royalblue','orangered']
sex = df.sex.value_counts().values
plt.figure(figsize = (10,10))
plt.pie(sex, labels=labels, colors=colors, autopct='%1.1f%%')
plt.title('proportion of sexs',color = 'black',fontsize = 30)
```

Out[166]:
```
Text(0.5, 1.0, 'proportion of sexs')
```



In [170]:
```
labels = df.age.value_counts().index
colors = ['royalblue','orangered']
age = df.age.value_counts().values
plt.figure(figsize = (8,8))
plt.pie(age, labels=labels, colors=colors, autopct='%1.1f%%')
plt.title('proportion of ages',color = 'black',fontsize = 30)
```

Out[170]:
```
Text(0.5, 1.0, 'proportion of ages')
```

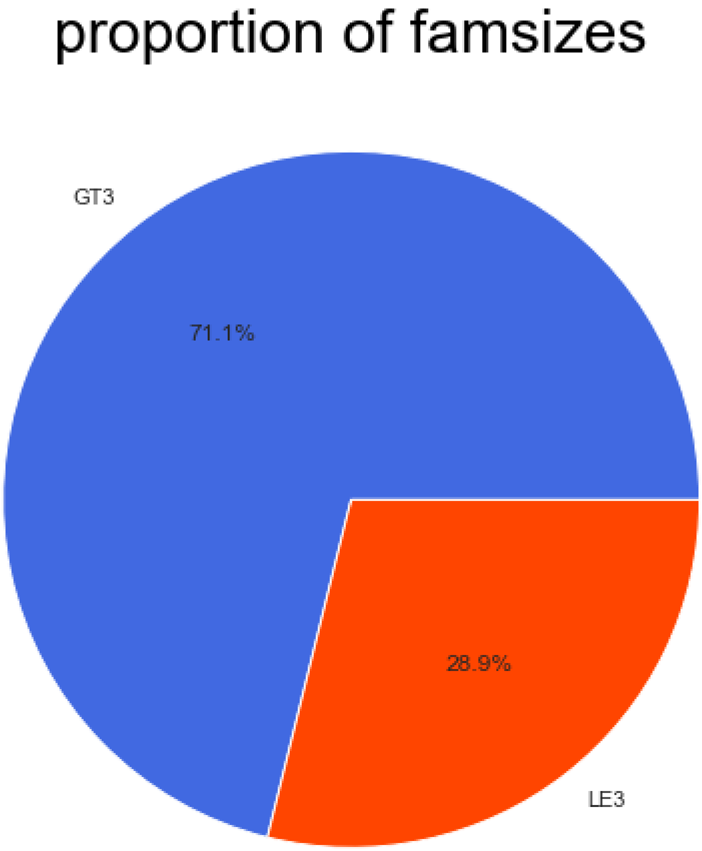


In [171]:

```
labels = df.famsize.value_counts().index
colors = ['royalblue','orangered']
famsize = df.famsize.value_counts().values
plt.figure(figsize = (8,8))
plt.pie(famsize, labels=labels, colors=colors, autopct='%1.1f%%')
plt.title('proportion of famsizes',color = 'black',fontsize = 30)
```

Out[171]:

```
Text(0.5, 1.0, 'proportion of famsizes')
```

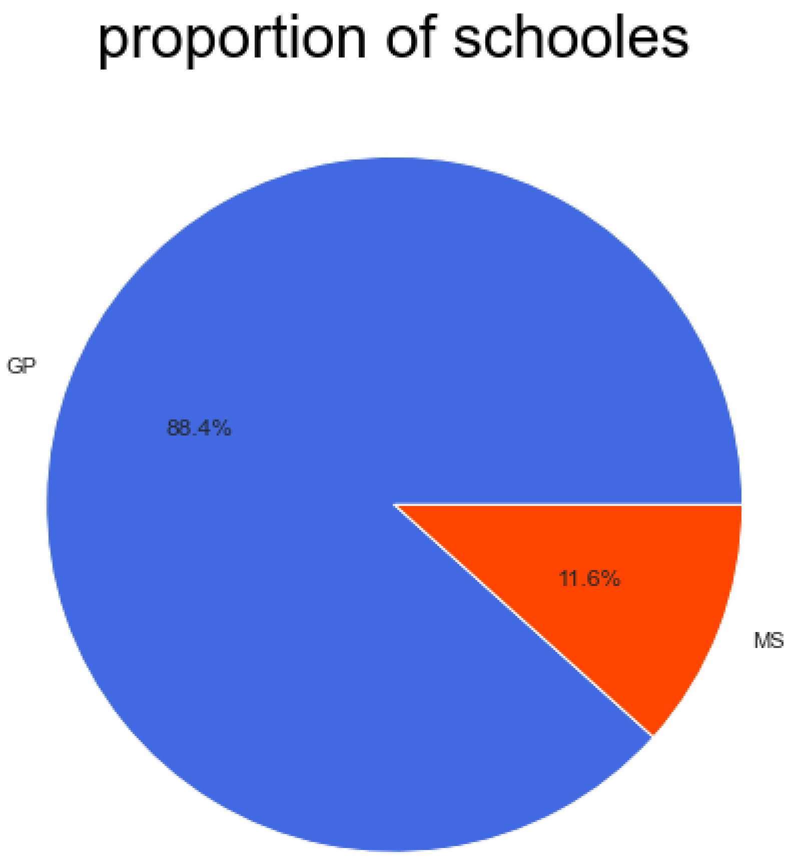


In [172]:

```
labels = df.school.value_counts().index
colors = ['royalblue','orangered']
school = df.school.value_counts().values
plt.figure(figsize = (8,8))
plt.pie(school, labels=labels, colors=colors, autopct='%1.1f%%')
plt.title('proportion of schooles',color = 'black',fontsize = 30)
```

Out[172]:

```
Text(0.5, 1.0, 'proportion of schooles')
```



In [173]:

```
female_df=df[df['sex']== 'F']
female_df.head()
```

Out[173]:

	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

In [182]:
```
female_df=df[df['sex']== 'M']
female_df.head()
```

Out[182]:

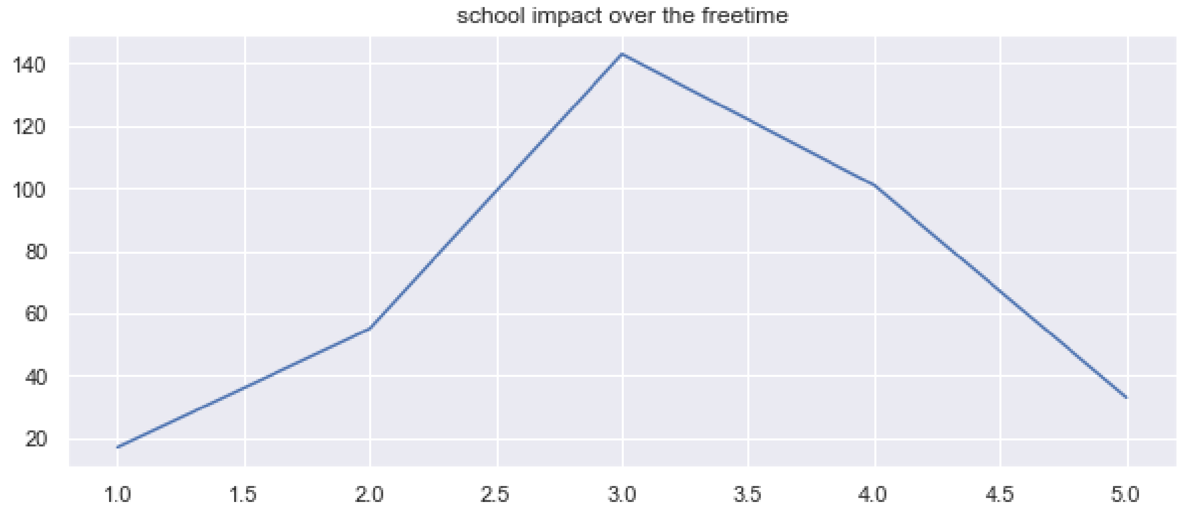
	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	free
5	GP	M	16	U	LE3	T	4	3	services	other	...	5	
6	GP	M	16	U	LE3	T	2	2	other	other	...	4	
8	GP	M	15	U	LE3	A	3	2	services	other	...	4	
9	GP	M	15	U	GT3	T	3	4	other	other	...	5	
12	GP	M	15	U	LE3	T	4	4	health	services	...	4	

5 rows × 33 columns

In [ ]:

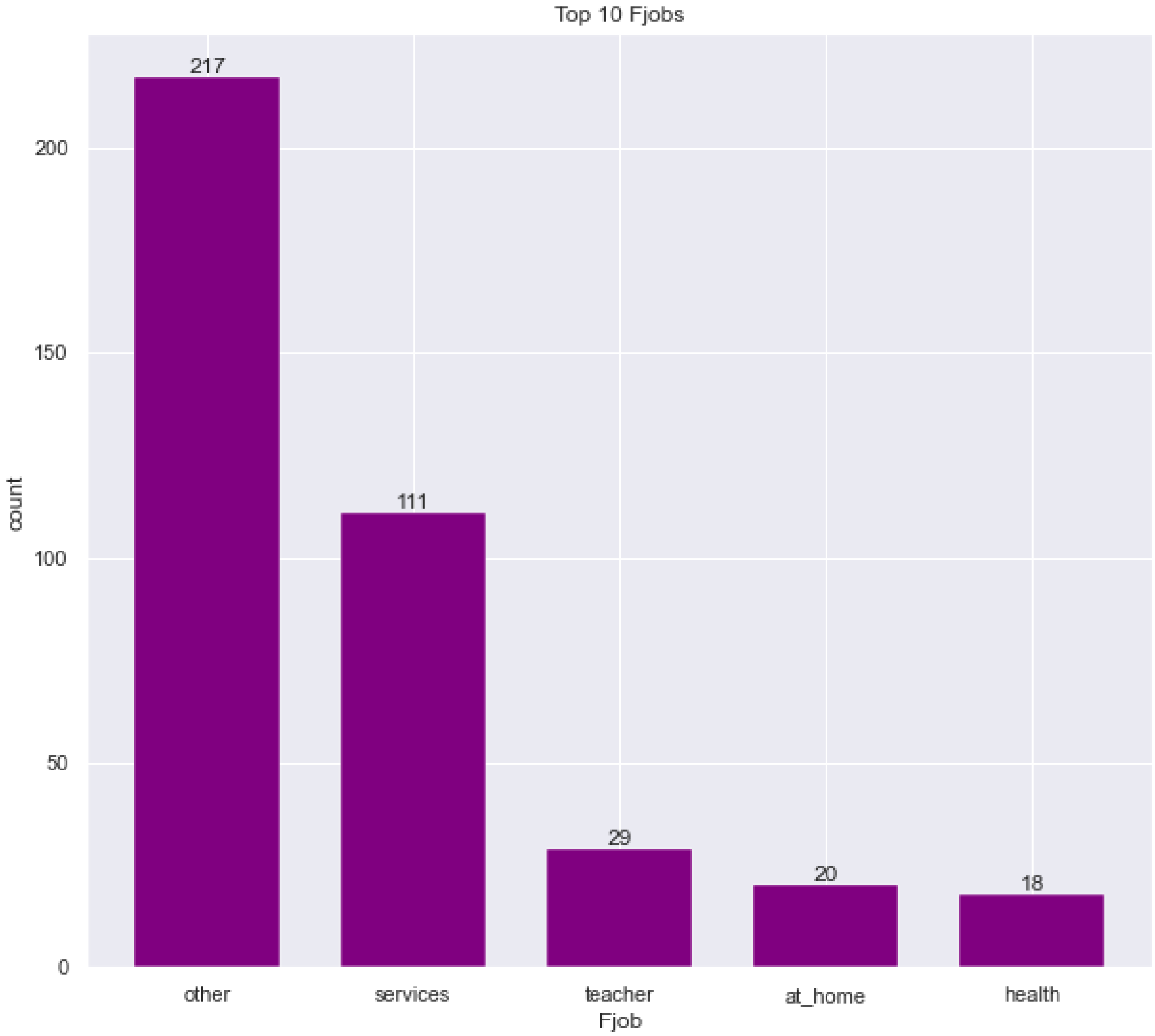
In [245]:
```
plt.figure(figsize=(10,4))
plt.title('school impact over the freetime')
school_df.freetime.value_counts().sort_index().plot()
```

Out[245]:
```
<AxesSubplot:title={'center':'school impact over the freetime'}>
```



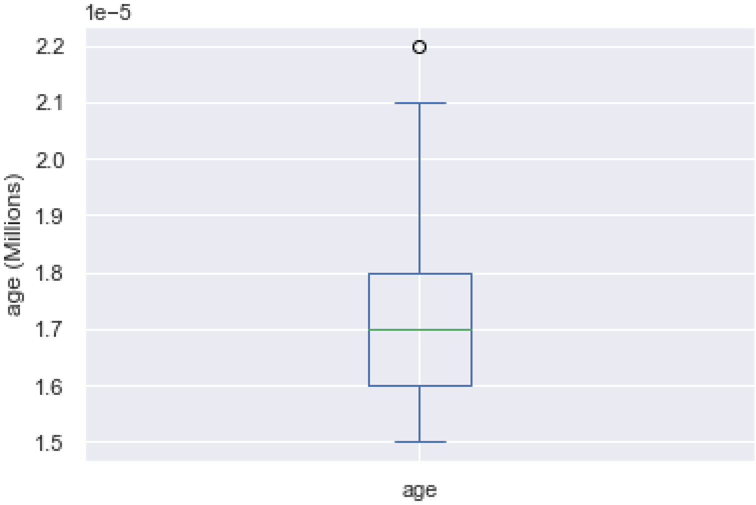
In [246]:

```
itemNames = df['Fjob'].value_counts().index[:10]
itemValues = df['Fjob'].value_counts().values[:10]
plt.figure(figsize=(10,9))
plt.ylabel('count', fontsize='medium')
plt.xlabel('Fjob', fontsize='medium')
plt.title('Top 10 Fjobs')
plt.bar(itemNames,itemValues, width = 0.7,color='purple',linewidth=0.4)
for i in range(len(itemNames)):
    plt.text(i,itemValues[i],itemValues[i],ha='center',va='bottom')
plt.show()
```



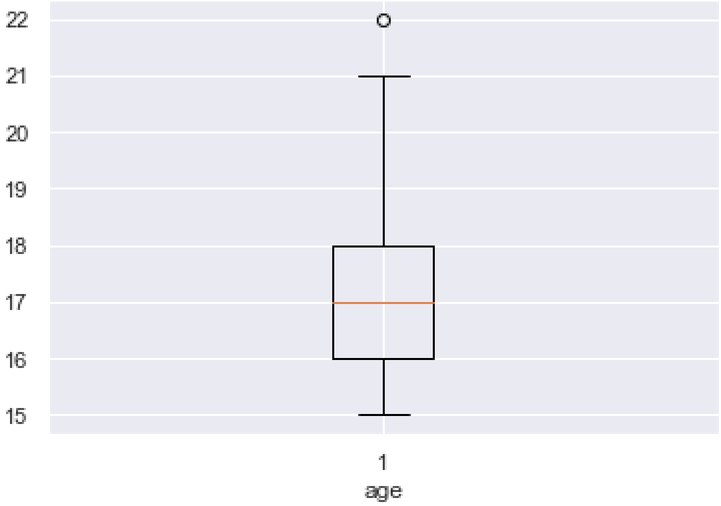
In [192]:
```
# Visualizing a boxplot using Pandas
ax = (df['age']/1000000).plot.box()
ax.set_ylabel('age (Millions)')
```

Out[192]:
```
Text(0, 0.5, 'age (Millions)')
```



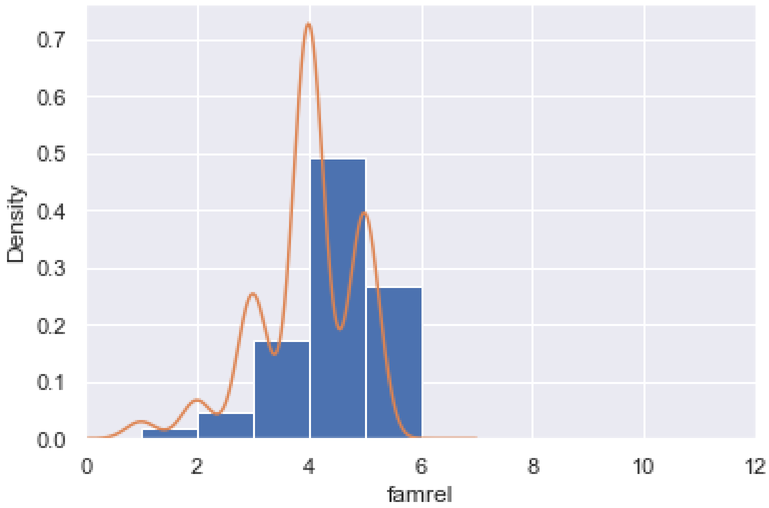
In [193]:
```
# Visualizing a boxplot using Matplotlib's boxplot() method
import matplotlib.pyplot as plt
plt.boxplot(df['age'])
plt.xlabel('age')
```

Out[193]:
```
Text(0.5, 0, 'age')
```



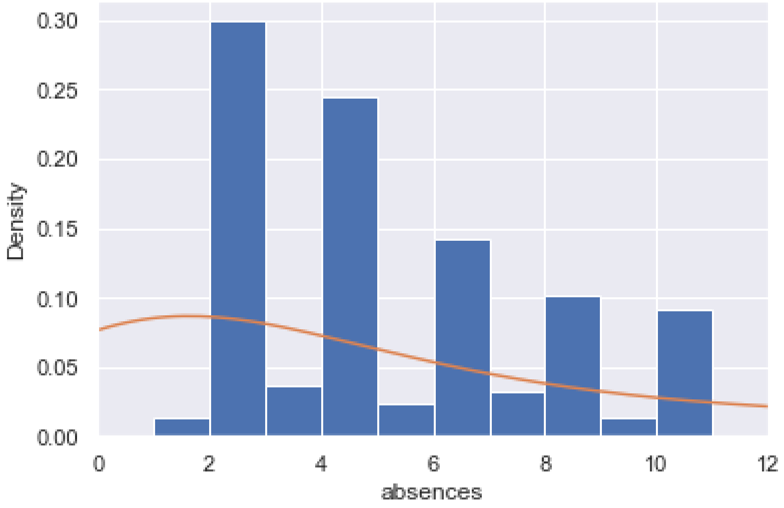
In [195]:
```
# Let's plot the density plot using hist() method in Pandas
ax = df['famrel'].plot.hist(density=True, xlim=[0,12], bins=range(1,12))
df['famrel'].plot.density(ax=ax)
ax.set_xlabel('famrel')
```

Out[195]:
```
Text(0.5, 0, 'famrel')
```



In [206]:
```
# Let's plot the density plot using hist() method in Pandas
ax = df['absences'].plot.hist(density=True, xlim=[0,12], bins=range(1,12))
df['absences'].plot.density(ax=ax)
ax.set_xlabel('absences')
```

Out[206]:
```
Text(0.5, 0, 'absences')
```

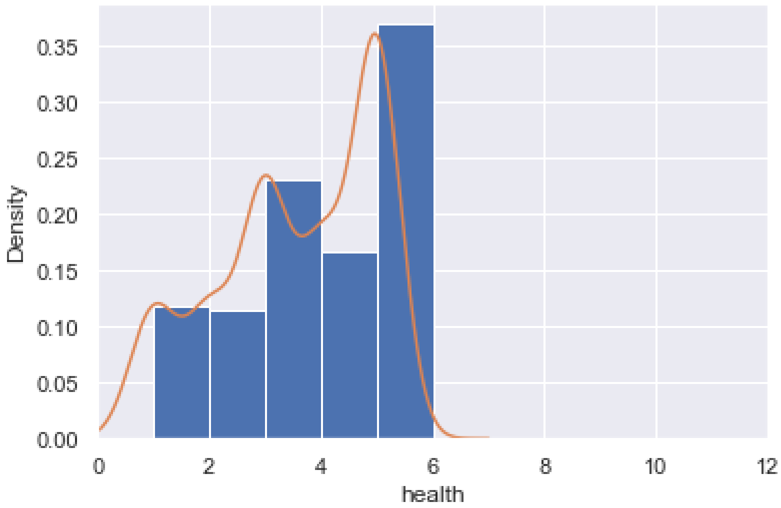


In [207]:

```
# Let's plot the density plot using hist() method in Pandas
ax = df['health'].plot.hist(density=True, xlim=[0,12], bins=range(1,12))
df['health'].plot.density(ax=ax)
ax.set_xlabel('health')
```

Out[207]:

```
Text(0.5, 0, 'health')
```

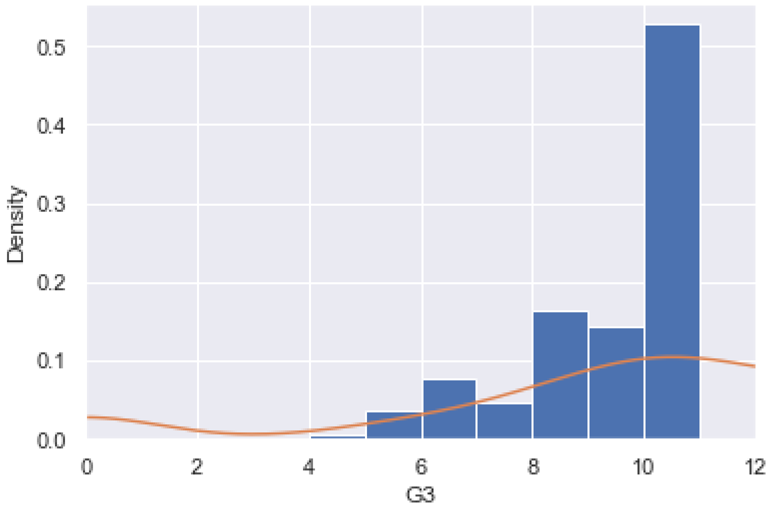


In [208]:

```
# Let's plot the density plot using hist() method in Pandas
ax = df['G3'].plot.hist(density=True, xlim=[0,12], bins=range(1,12))
df['G3'].plot.density(ax=ax)
ax.set_xlabel('G3')
```

Out[208]:

```
Text(0.5, 0, 'G3')
```



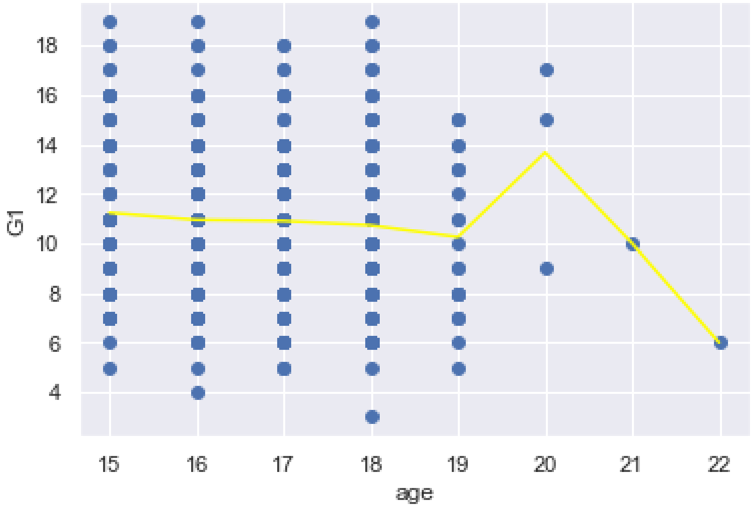
In [223]:

```
# Plotting relationships between variables using Matplotlib's scatter() method
plt.scatter(df['age'], df['G1'])
plt.xlabel('age')
plt.ylabel('G1')
plt.plot(np.unique(df['age']), np.poly1d(np.polyfit(df['age'], df['G1'], 10))(
```

```
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3
369: RankWarning: Polyfit may be poorly conditioned
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Out[223]:

```
[<matplotlib.lines.Line2D at 0x178a12b0>]
```



In [ ]:

```
G2  G3
```

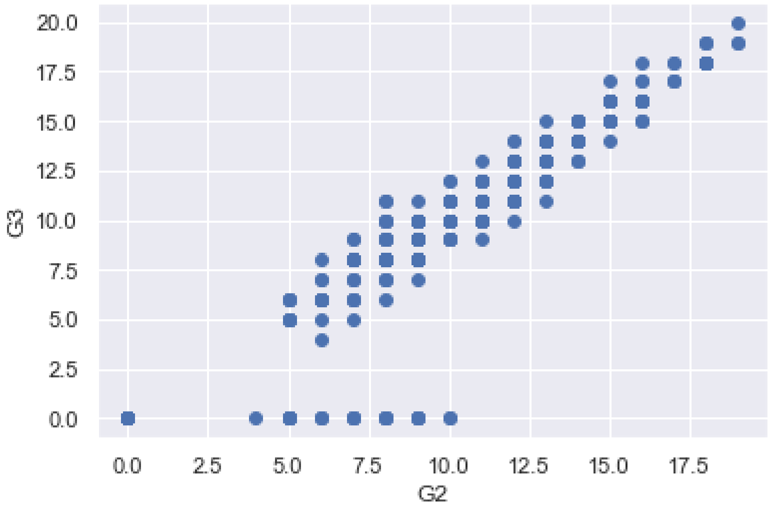
In [213]:

```
# We are creating a scatter plot of the two variables
plt.scatter(df['G2'],df['G3'])

# Name your axes
plt.xlabel('G2')
plt.ylabel('G3')
```

Out[213]:

```
Text(0, 0.5, 'G3')
```



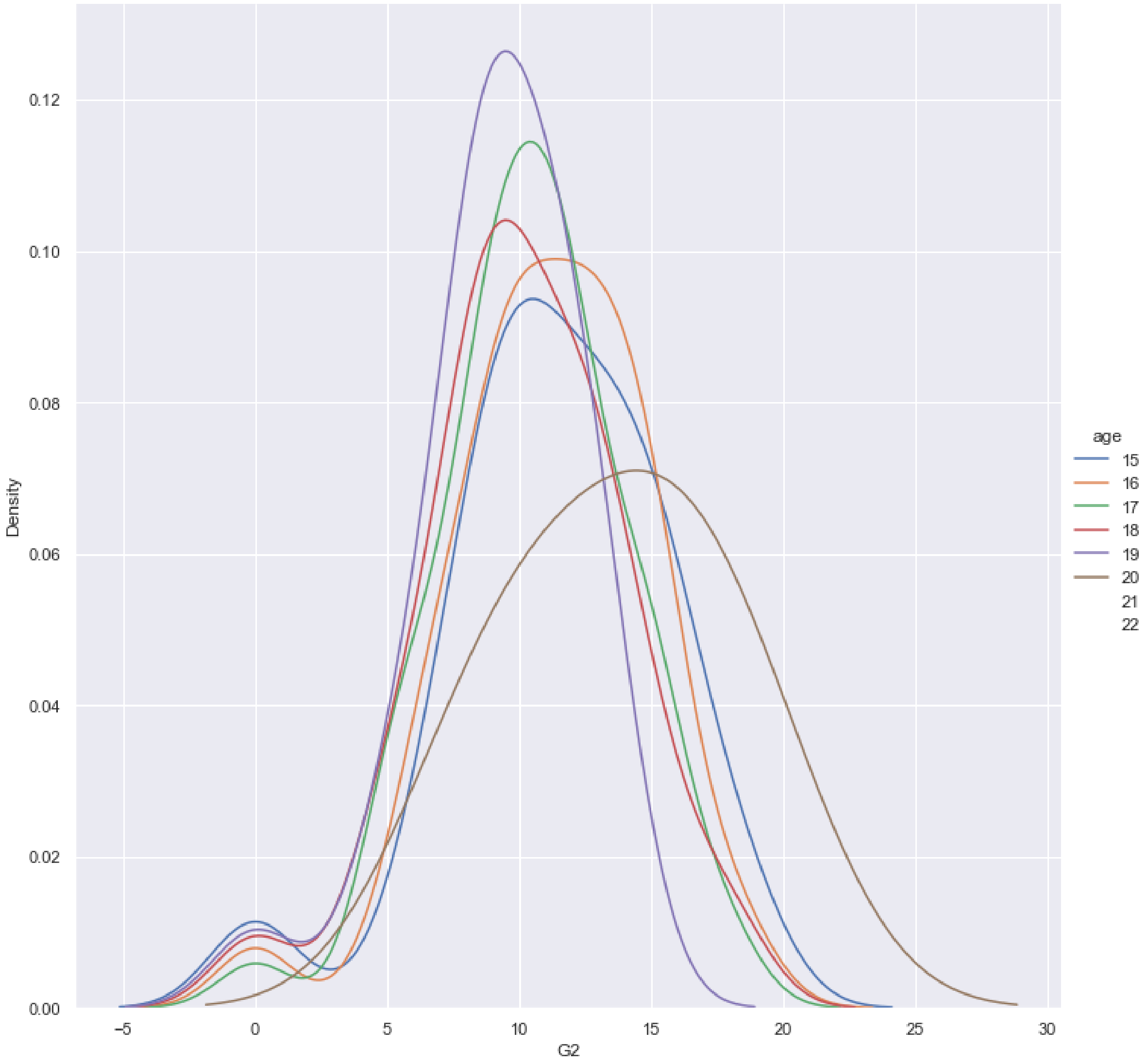
In [231]:

```
# A final seaborn plot useful for looking at univariate relations is the kdeplo
# which creates and visualizes a kernel density estimate of the underlying feat
sns.FacetGrid(df, hue="age", size=10) \
   .map(sns.kdeplot, "G2") \
   .add_legend()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\axisgrid.py:337: UserWarni
ng: The `size` parameter has been renamed to `height`; please update your cod
e.
  warnings.warn(msg, UserWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:316: User
Warning: Dataset has 0 variance; skipping density estimate. Pass `warn_singul
ar=False` to disable this warning.
  warnings.warn(msg, UserWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:316: User
Warning: Dataset has 0 variance; skipping density estimate. Pass `warn_singul
ar=False` to disable this warning.
  warnings.warn(msg, UserWarning)
```

Out[231]:

```
<seaborn.axisgrid.FacetGrid at 0x17892028>
```



THANK YOU !!