In [3]:
```python
import pandas as pd
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score

sns.set()
```

In [5]:
```python
data = pd.read_csv('xAPI-Edu-Data.csv')
```

In [6]:
```python
data.head()
```

Out[6]:

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Rel
0	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	F
1	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	F
2	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	F
3	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	F
4	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	F

In [7]:
```python
data.shape
```

Out[7]:
```
(480, 17)
```

In [8]:
```python
data.dtypes
```

Out[8]:
```
gender                      object
NationalITy                 object
PlaceofBirth                object
StageID                     object
GradeID                     object
SectionID                   object
Topic                       object
Semester                    object
Relation                    object
raisedhands                  int64
VisITedResources             int64
AnnouncementsView            int64
Discussion                   int64
ParentAnsweringSurvey       object
ParentschoolSatisfaction    object
StudentAbsenceDays          object
Class                       object
dtype: object
```

In [9]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 480 entries, 0 to 479
Data columns (total 17 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   gender                    480 non-null    object
 1   NationalITy               480 non-null    object
 2   PlaceofBirth              480 non-null    object
 3   StageID                   480 non-null    object
 4   GradeID                   480 non-null    object
 5   SectionID                 480 non-null    object
 6   Topic                     480 non-null    object
 7   Semester                  480 non-null    object
 8   Relation                  480 non-null    object
 9   raisedhands               480 non-null    int64 
 10  VisITedResources          480 non-null    int64 
 11  AnnouncementsView         480 non-null    int64 
 12  Discussion                480 non-null    int64 
 13  ParentAnsweringSurvey     480 non-null    object
 14  ParentschoolSatisfaction  480 non-null    object
 15  StudentAbsenceDays        480 non-null    object
 16  Class                     480 non-null    object
dtypes: int64(4), object(13)
memory usage: 63.9+ KB
```

In [14]:
```
data.describe(include="object")
```

Out[14]:

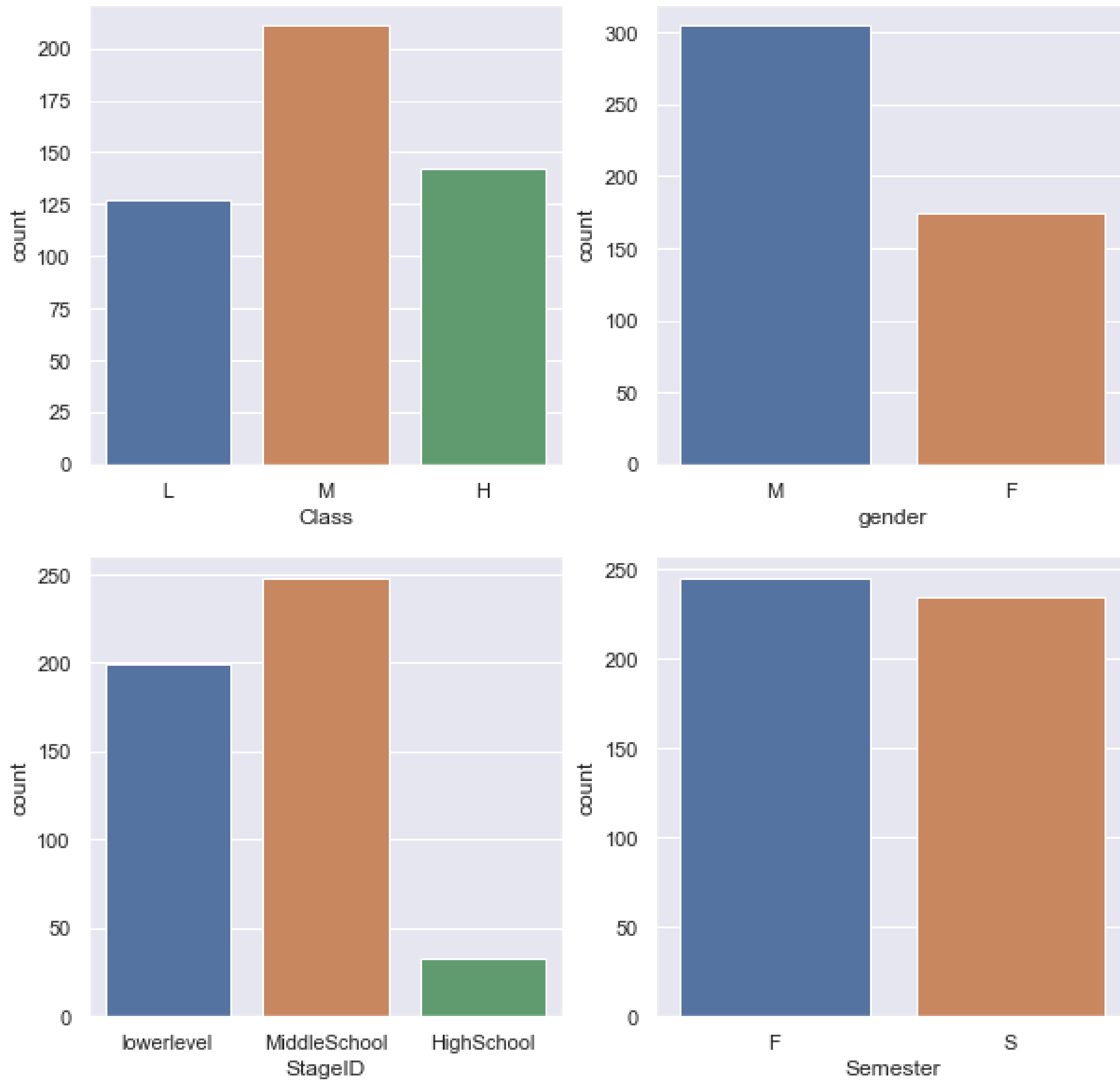
	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Seme
count	480	480	480	480	480	480	480	
unique	2	14	14	3	10	3	12	
top	M	KW	KuwaIT	MiddleSchool	G-02	A	IT	
freq	305	179	180	248	147	283	95	

1. Visualize just the categorical features individually to see what options are included and how each option fares when it comes to count(how many times it appears) and see what can be deduce from that?

In [3]:
```
fig, axarr  = plt.subplots(2,2,figsize=(10,10))
sns.countplot(x='Class', data=data, ax=axarr[0,0], order=['L','M','H'])
sns.countplot(x='gender', data=data, ax=axarr[0,1], order=['M','F'])
sns.countplot(x='StageID', data=data, ax=axarr[1,0])
sns.countplot(x='Semester', data=data, ax=axarr[1,1])
```

Out[3]:
```
<AxesSubplot:xlabel='Semester', ylabel='count'>
```

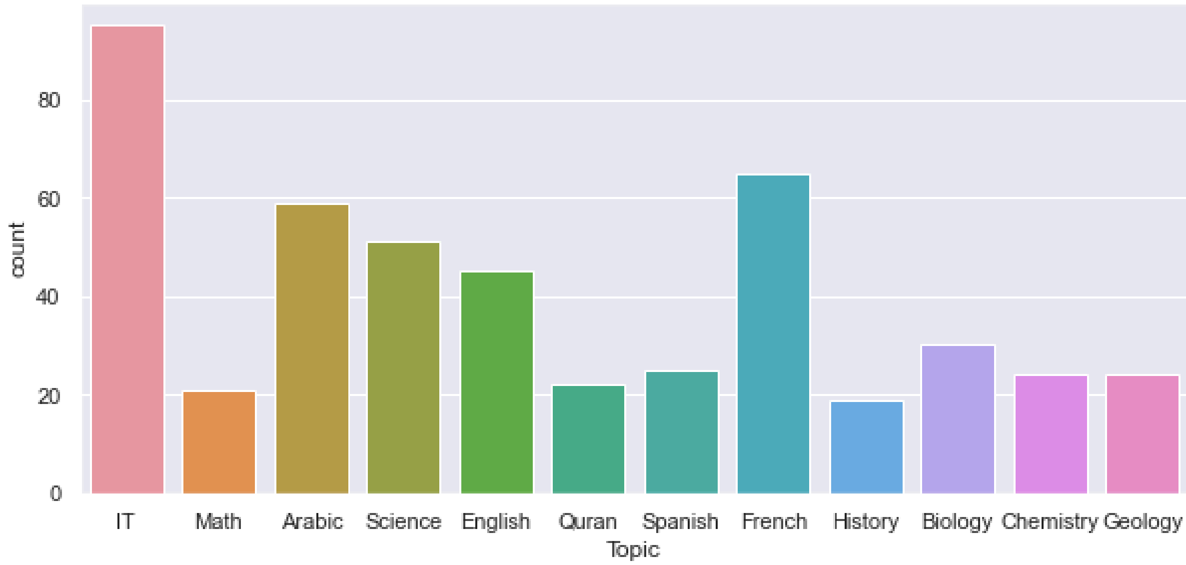
Loading [MathJax]/extensions/Safe.js

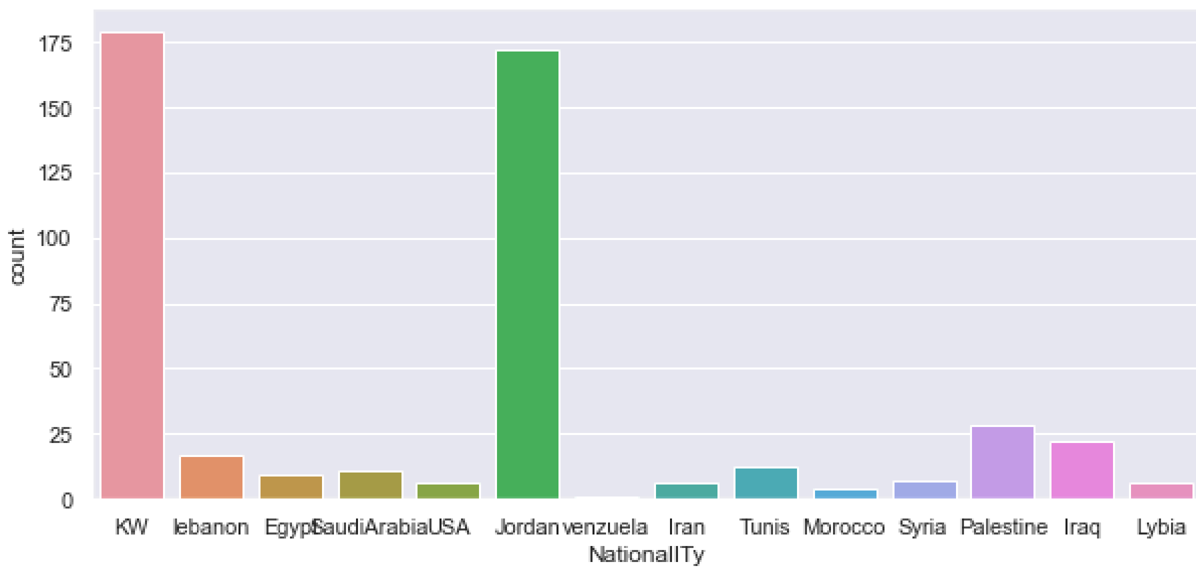


In [4]:
```
fig, (axis1, axis2)  = plt.subplots(2, 1,figsize=(10,10))
sns.countplot(x='Topic', data=data, ax=axis1)
sns.countplot(x='NationalITy', data=data, ax=axis2)
```

Out[4]: <AxesSubplot:xlabel='NationalITy', ylabel='count'>

Loading [MathJax]/extensions/Safe.js



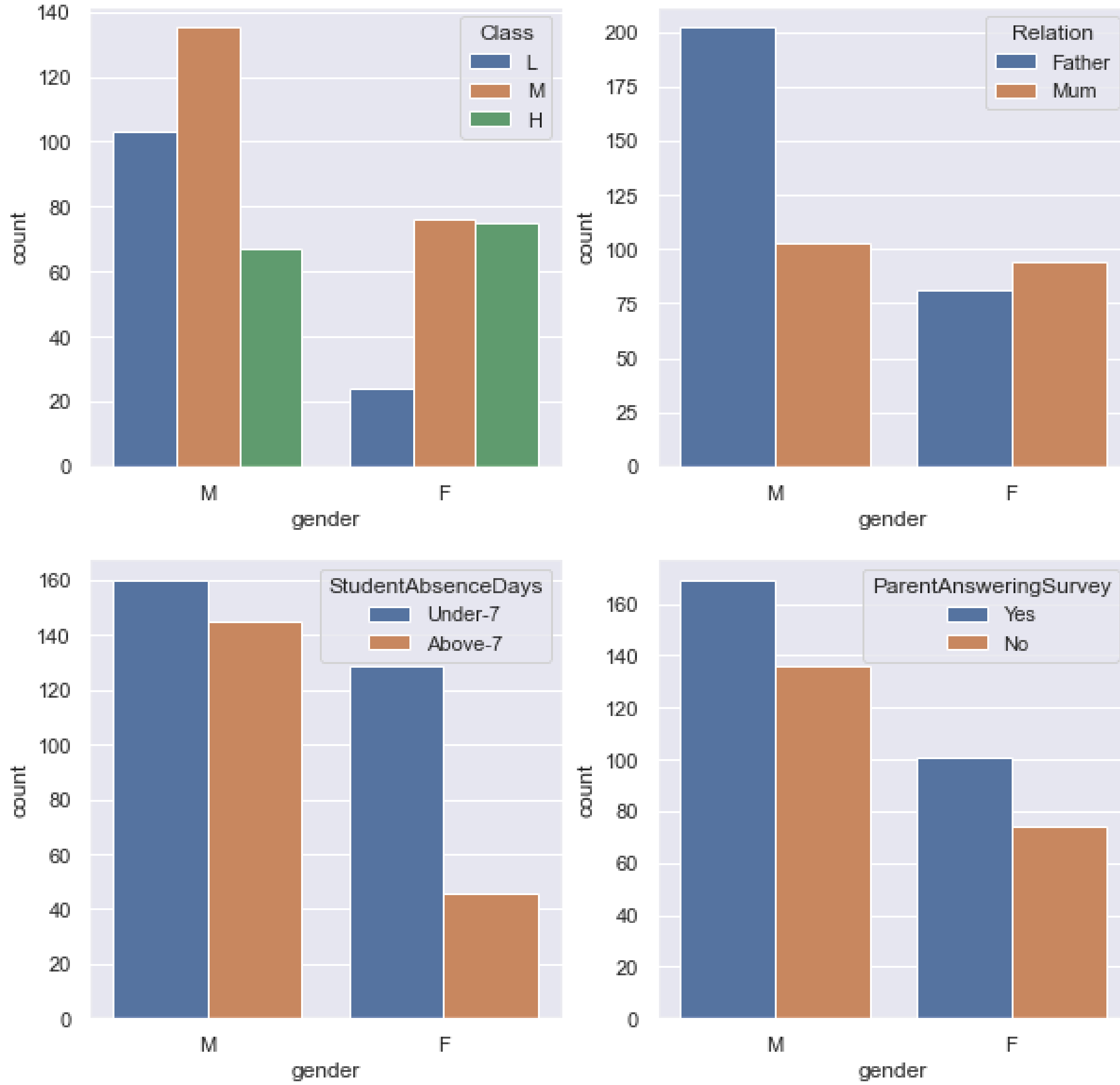


Ans : Most of these countries are in the middle east(Islamic states)

2. Look at some categorical features in relation to each other, to see what insights could be possibly read?

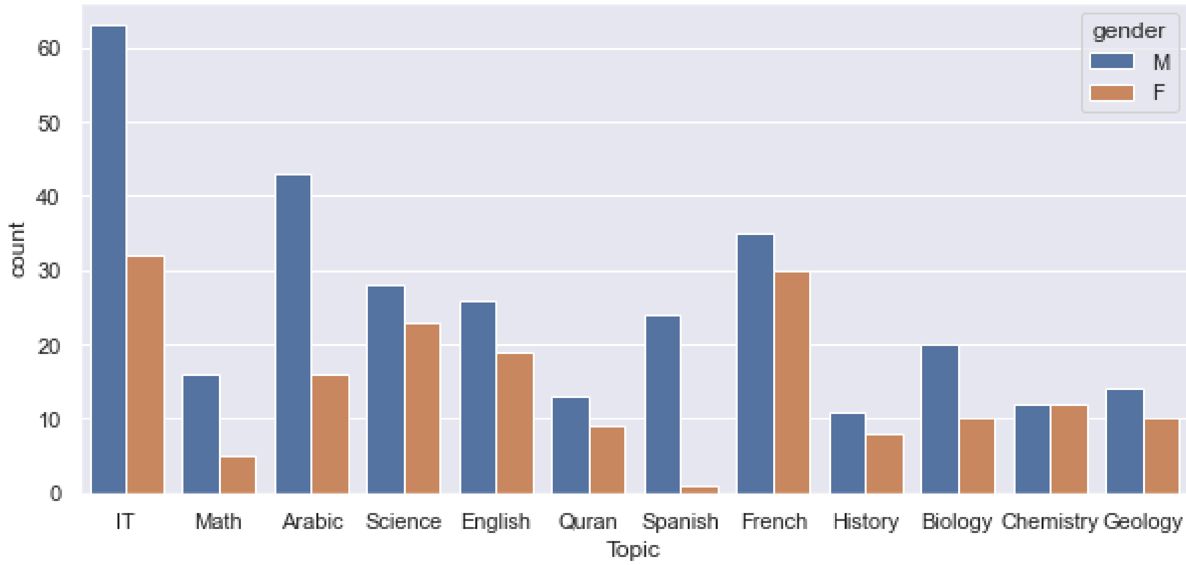
In [5]:
```
fig, axarr  = plt.subplots(2,2,figsize=(10,10))
sns.countplot(x='gender', hue='Class', data=data, ax=axarr[0,0], order=['M',
sns.countplot(x='gender', hue='Relation', data=data, ax=axarr[0,1], order=['
sns.countplot(x='gender', hue='StudentAbsenceDays', data=data, ax=axarr[1,0]
sns.countplot(x='gender', hue='ParentAnsweringSurvey', data=data, ax=axarr[1
```

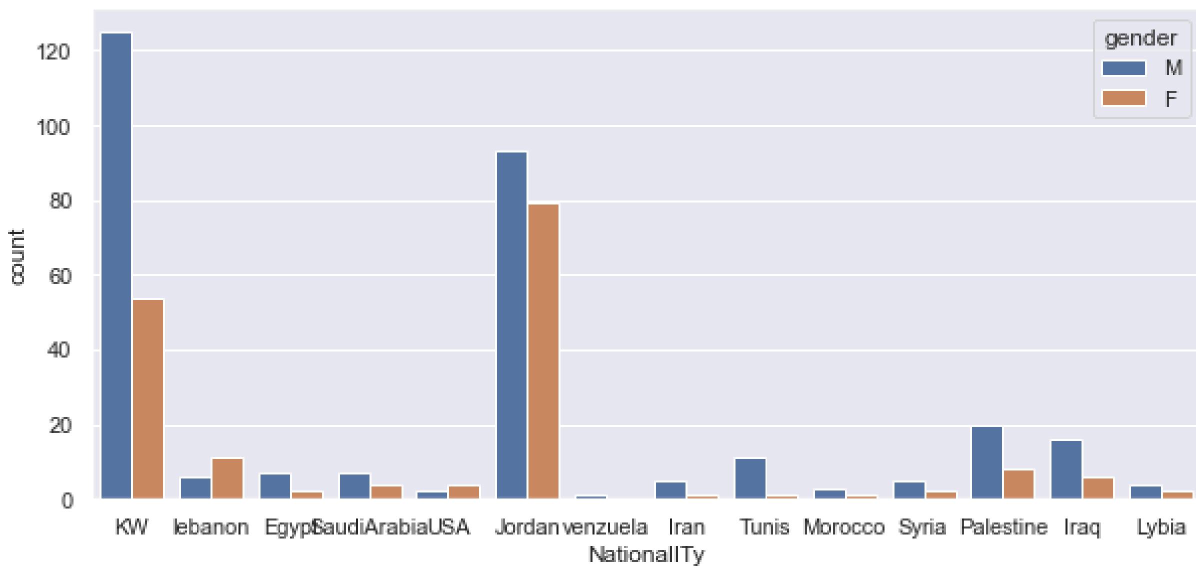
Out[5]:
```
<AxesSubplot:xlabel='gender', ylabel='count'>
```



In [6]:
```
fig, (axis1, axis2)  = plt.subplots(2, 1,figsize=(10,10))
sns.countplot(x='Topic', hue='gender', data=data, ax=axis1)
sns.countplot(x='NationalITy', hue='gender', data=data, ax=axis2)
```

Out[6]:
```
<AxesSubplot:xlabel='NationalITy', ylabel='count'>
```





Ans :

- Girls seem to have performed better than boys
- In the case of girls, mothers seem to be more interested in their education than fathers
- Girls had much better attendance than boys
- No apparent gender bias when it comes to subject/topic choices, we cannot conclude that girls performed better because they perhaps took less technical subjects
- Gender disparity holds even at a country level. May just be as a result of the sampling

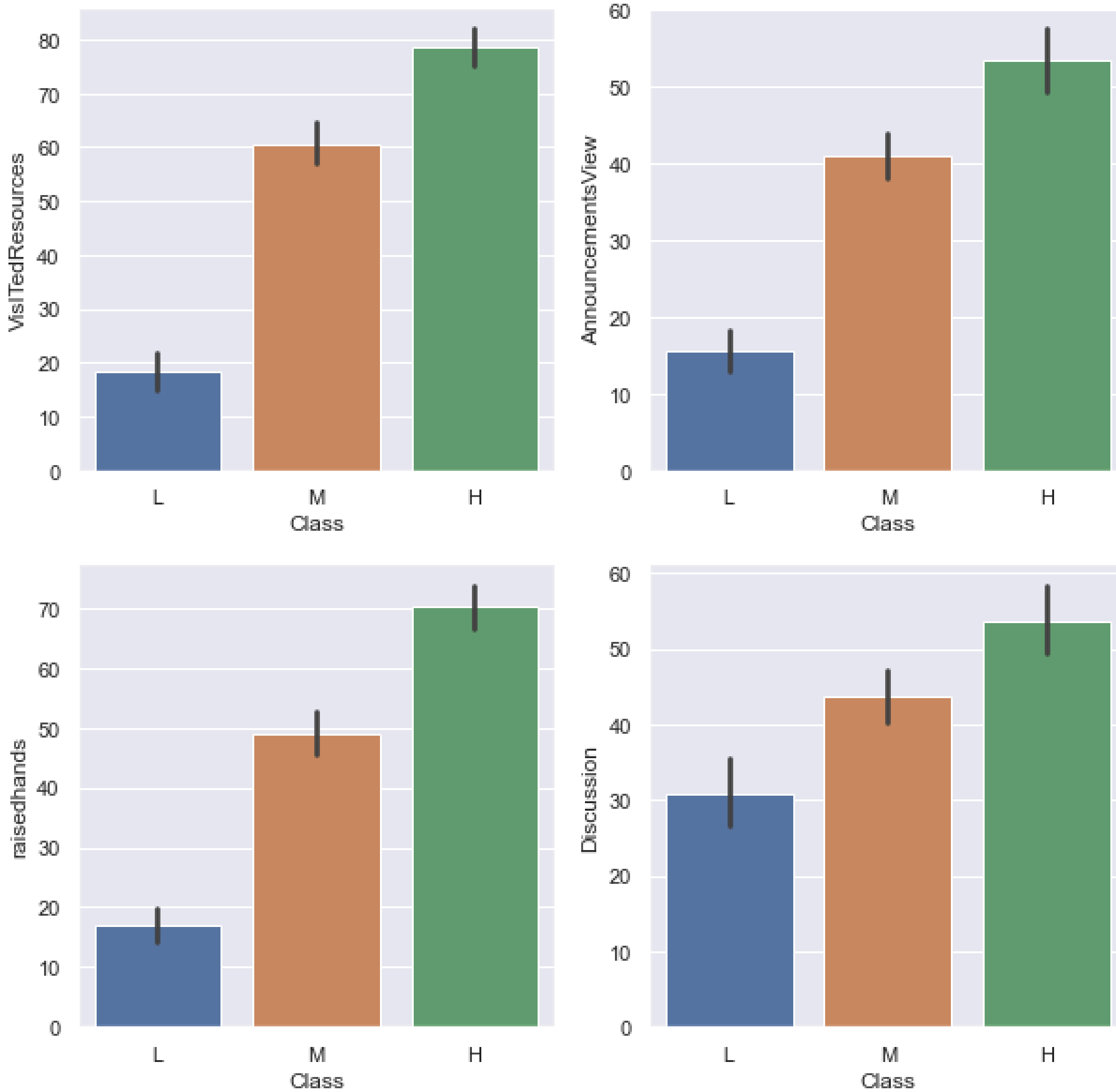
3. Visualize categorical variables with numerical variables and give conclusions?

In [7]:

```
fig, axarr  = plt.subplots(2,2,figsize=(10,10))
sns.barplot(x='Class', y='VisITedResources', data=data, order=['L','M','H'],
sns.barplot(x='Class', y='AnnouncementsView', data=data, order=['L','M','H']
sns.barplot(x='Class', y='raisedhands', data=data, order=['L','M','H'], ax=a
sns.barplot(x='Class', y='Discussion', data=data, order=['L','M','H'], ax=ax
```

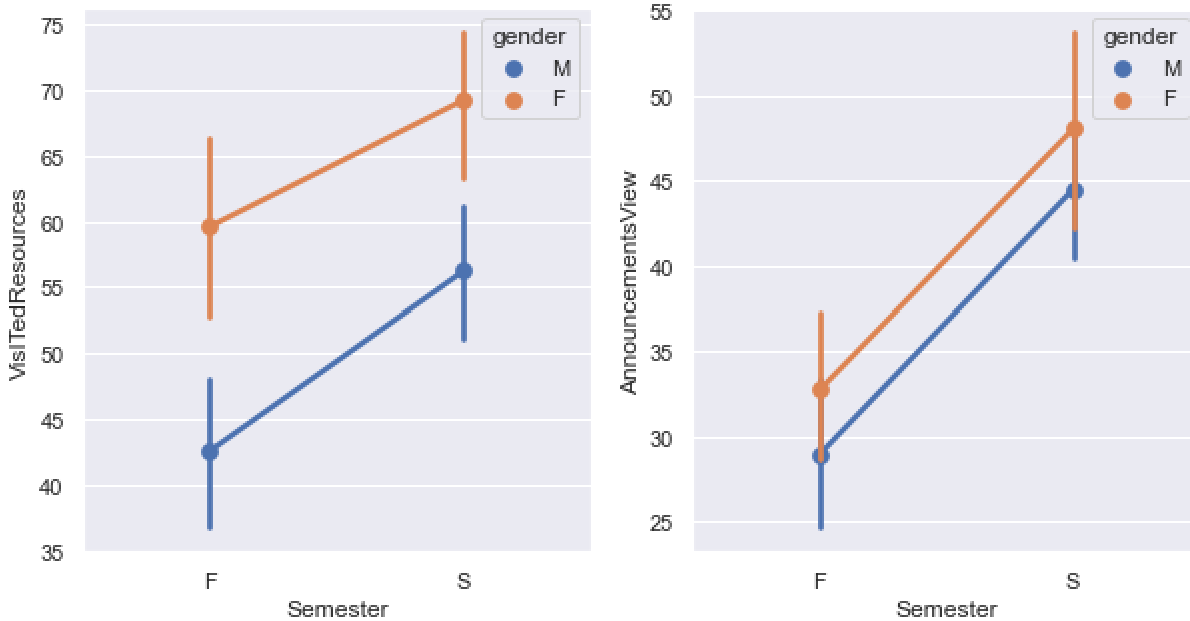
Loading [MathJax]/extensions/Safe.js

Out[7]: <AxesSubplot:xlabel='Class', ylabel='Discussion'>



In [8]:
```
fig, (axis1, axis2)  = plt.subplots(1, 2,figsize=(10,5))
sns.pointplot(x='Semester', y='VisITedResources', hue='gender', data=data, a
sns.pointplot(x='Semester', y='AnnouncementsView', hue='gender', data=data,
```

Out[8]: <AxesSubplot:xlabel='Semester', ylabel='AnnouncementsView'>



Ans :

- As expected, those that participated more (higher counts in Discussion, raisedhands, AnnouncementViews, RaisedHands), performed better
- In the case of both visiting resources and viewing announcements, students were more vigilant in the second semester, perhaps that last minute need to boost your final grade

In [9]:

```
ave_raisedhands = sum(data['raisedhands'])/len(data['raisedhands'])
ave_VisITedResources = sum(data['VisITedResources'])/len(data['VisITedResour
ave_AnnouncementsView = sum(data['AnnouncementsView'])/len(data['Announcemer
unsuccess = data.loc[(data['raisedhands'] >= ave_raisedhands) & (data['VisIT
```

In [10]:

```
unsuccess
```

Out[10]:

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Sem
444	M	Jordan	Jordan	MiddleSchool	G-08	A	Chemistry	
445	M	Jordan	Jordan	MiddleSchool	G-08	A	Chemistry	

4. From the above result, what are the factors that leads to get low grades of the students?

Note : Above two students have features ('raisedhands' , 'VisITedResources' , 'AnnouncementsView') greater than average

In [11]:

```
data['numeric_class'] = [1 if data.loc[i,'Class'] == 'L' else 2 if data.loc[
```

In [12]:

```
grade_male_ave = sum(data[data.gender == 'M'].numeric_class)/float(len(data[
grade_female_ave = sum(data[data.gender == 'F'].numeric_class)/float(len(dat
```

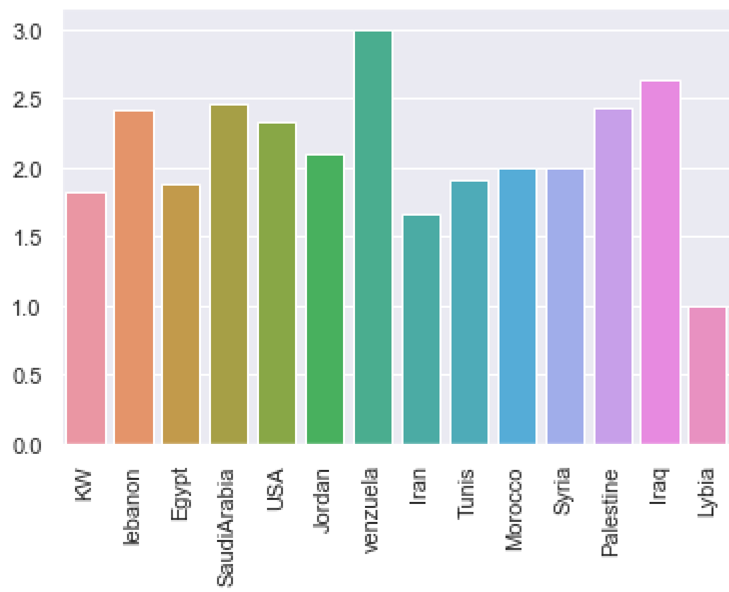
Loading [MathJax]/extensions/Safe.js

- Gender comparison cannot completely explain low level grades

In [13]:
```
# Now lets look at nationality
nation = data.NationalITy.unique()
nation_grades_ave = [sum(data[data.NationalITy == i].numeric_class)/float(le
ax = sns.barplot(x=nation, y=nation_grades_ave)
jordan_ave = sum(data[data.NationalITy == 'Jordan'].numeric_class)/float(ler
print('Jordan average: '+str(jordan_ave))
plt.xticks(rotation=90)
```

```
Jordan average: 2.0930232558139537
```

Out[13]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13]),
 [Text(0, 0, 'KW'),
  Text(1, 0, 'lebanon'),
  Text(2, 0, 'Egypt'),
  Text(3, 0, 'SaudiArabia'),
  Text(4, 0, 'USA'),
  Text(5, 0, 'Jordan'),
  Text(6, 0, 'venzuela'),
  Text(7, 0, 'Iran'),
  Text(8, 0, 'Tunis'),
  Text(9, 0, 'Morocco'),
  Text(10, 0, 'Syria'),
  Text(11, 0, 'Palestine'),
  Text(12, 0, 'Iraq'),
  Text(13, 0, 'Lybia')])
```



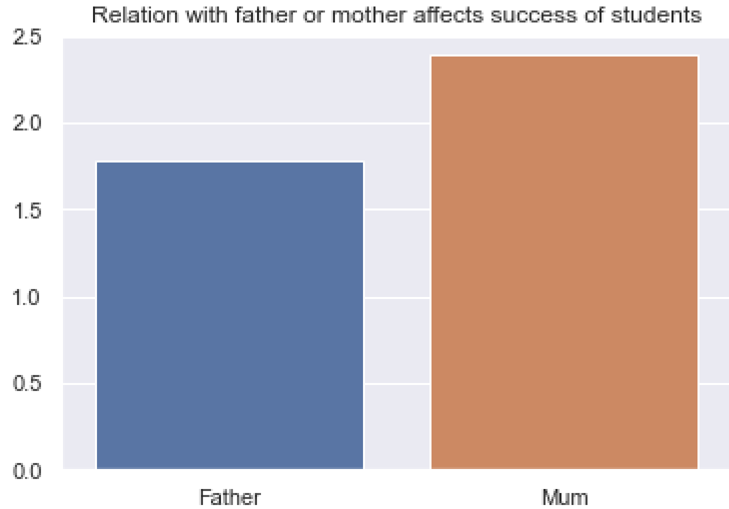
- As it can be seen in bar plot Jordan is seventh country with average 2.09 so 'Jordan' has positive impact on these two students actually

In [14]:
```
# Lets look at relation with family members
relation = data.Relation.unique()
relation_grade_ave = [sum(data[data.Relation == i].numeric_class)/float(len(
```

Loading [MathJax]/extensions/Safe.js

```
ax = sns.barplot(x=relation, y=relation_grade_ave)
plt.title('Relation with father or mother affects success of students')
```

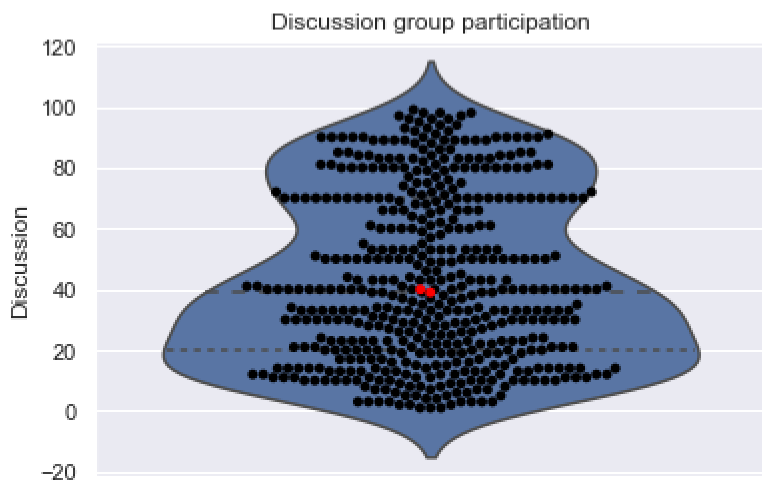
Out[14]: Text(0.5, 1.0, 'Relation with father or mother affects success of student
s')



- Having relation has positive effect on these students

In [15]:
```
#Lets look at how many times the student participate on discussion groups
discussion = data.Discussion
discussion_ave = sum(discussion)/len(discussion)
ax = sns.violinplot(y=discussion,split=True,inner='quart')
ax = sns.swarmplot(y=discussion,color='black')
ax = sns.swarmplot(y = unsuccess.Discussion, color='red')
plt.title('Discussion group participation')
```

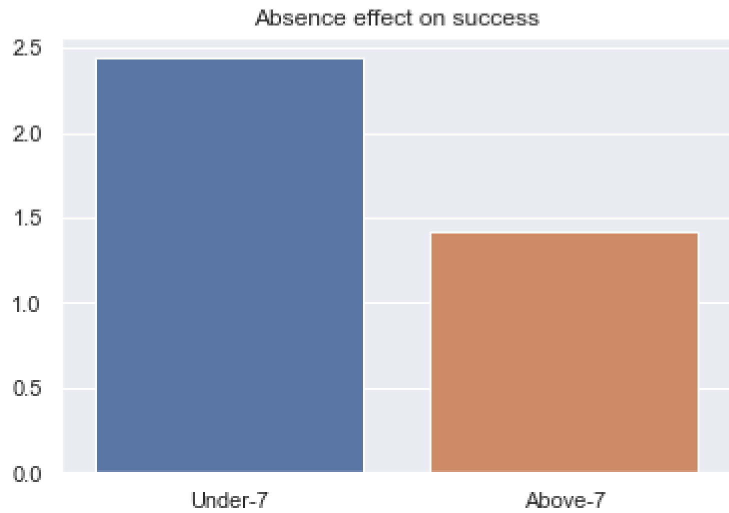
Out[15]: Text(0.5, 1.0, 'Discussion group participation')



In [16]:
```
# Now lastly lets look at
absence_day = data.StudentAbsenceDays.unique()
absense_day_ave = [sum(data[data.StudentAbsenceDays == i].numeric_class)/flo
ax = sns.barplot(x=absence_day, y=absense_day_ave)
plt.title('Absence effect on success')
```

Loading [MathJax]/extensions/Safe.js

Out[16]: Text(0.5, 1.0, 'Absence effect on success')



5. Build classification model and present it's classification report ?

In [17]:
```
data.head()
```

Out[17]:

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Rel
0	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	F
1	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	F
2	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	F
3	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	F
4	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	F

In [18]:
```
data1 = data.drop('Class',axis = 1)
data_with_dummies = pd.get_dummies(data1, drop_first=True)
```

In [19]:
```
data_with_dummies.head()
```

Out[19]:

	raisedhands	VisITedResources	AnnouncementsView	Discussion	numeric_class	genc
0	15	16	2	20	2	
1	20	20	3	25	2	
2	10	7	0	30	1	
3	30	25	5	35	1	
4	40	50	12	50	2	

5 rows × 61 columns

Loading [MathJax]/extensions/Safe.js

In [20]:
```python
Features = data_with_dummies.drop(['numeric_class'],axis = 1)
Target = data_with_dummies['numeric_class']
```

In [21]:
```python
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaler.fit(Features)
```

Out[21]:
```
StandardScaler()
```

In [22]:
```python
X = scaler.fit_transform(Features)
```

In [23]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, Target, test_size=0.3
```

In [24]:
```python
Logit_Model = LogisticRegression()
Logit_Model.fit(X_train,y_train)
```

Out[24]:
```
LogisticRegression()
```

In [25]:
```python
Prediction = Logit_Model.predict(X_test)
Score = accuracy_score(y_test,Prediction)
Report = classification_report(y_test,Prediction)
```

In [26]:
```python
Prediction
```

Out[26]:
```
array([2, 2, 3, 1, 1, 1, 1, 3, 2, 2, 2, 3, 2, 2, 1, 1, 1, 2, 1, 1, 3, 3,
       2, 3, 2, 2, 3, 2, 2, 3, 3, 3, 3, 2, 2, 2, 3, 2, 2, 3, 1, 3, 2, 1,
       2, 2, 3, 2, 2, 2, 2, 1, 2, 2, 2, 2, 3, 2, 3, 1, 3, 1, 2, 2, 2, 2,
       1, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 3, 2, 1, 2, 2, 3, 2, 3, 3, 3, 3,
       2, 3, 2, 1, 2, 1, 3, 3, 2, 3, 2, 3, 2, 1, 2, 1, 2, 2, 3, 2, 2, 1,
       3, 2, 2, 3, 2, 2, 2, 2, 1, 1, 3, 1, 3, 1, 3, 3, 1, 3, 3, 3, 1, 3,
       3, 3, 2, 1, 1, 1, 3, 2, 2, 1, 2, 2])
```

In [27]:
```python
Score
```

Out[27]:
```
0.7361111111111112
```

In [28]:
```python
print(Report)
```

```
              precision    recall  f1-score   support

           1       0.76      0.87      0.81        30
           2       0.78      0.70      0.74        74
           3       0.65      0.70      0.67        40

    accuracy                           0.74       144
   macro avg       0.73      0.76      0.74       144
weighted avg       0.74      0.74      0.74       144
```

In [ ]: