What is EDA?

EDA refers to a set of procedures for producing descriptive and graphical summaries of data. A benefit of EDA is that is allows you to examine the data as it is without making any assumptions.

It is a useful way to examine your data, analyse relationships among variables and identify any problems such as data entry errors.

In [1]:
```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory

from subprocess import check_output
print(check_output(["ls", "../input"]).decode("utf8"))

# Any results you write to the current directory are saved as output.
```

```
xAPI-Edu-Data.csv
```

In [2]:
```
data = pd.read_csv('../input/xAPI-Edu-Data.csv')
data.head()
```

Out[2]:

	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Relation	raisedhands	VisITedResources	AnnouncementsView
0	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	15	16	2
1	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	20	20	3
2	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	10	7	0
3	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	30	25	5
4	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	40	50	12

Get a concise summary of the dataframe

In [3]:
```
1 data.info()
```

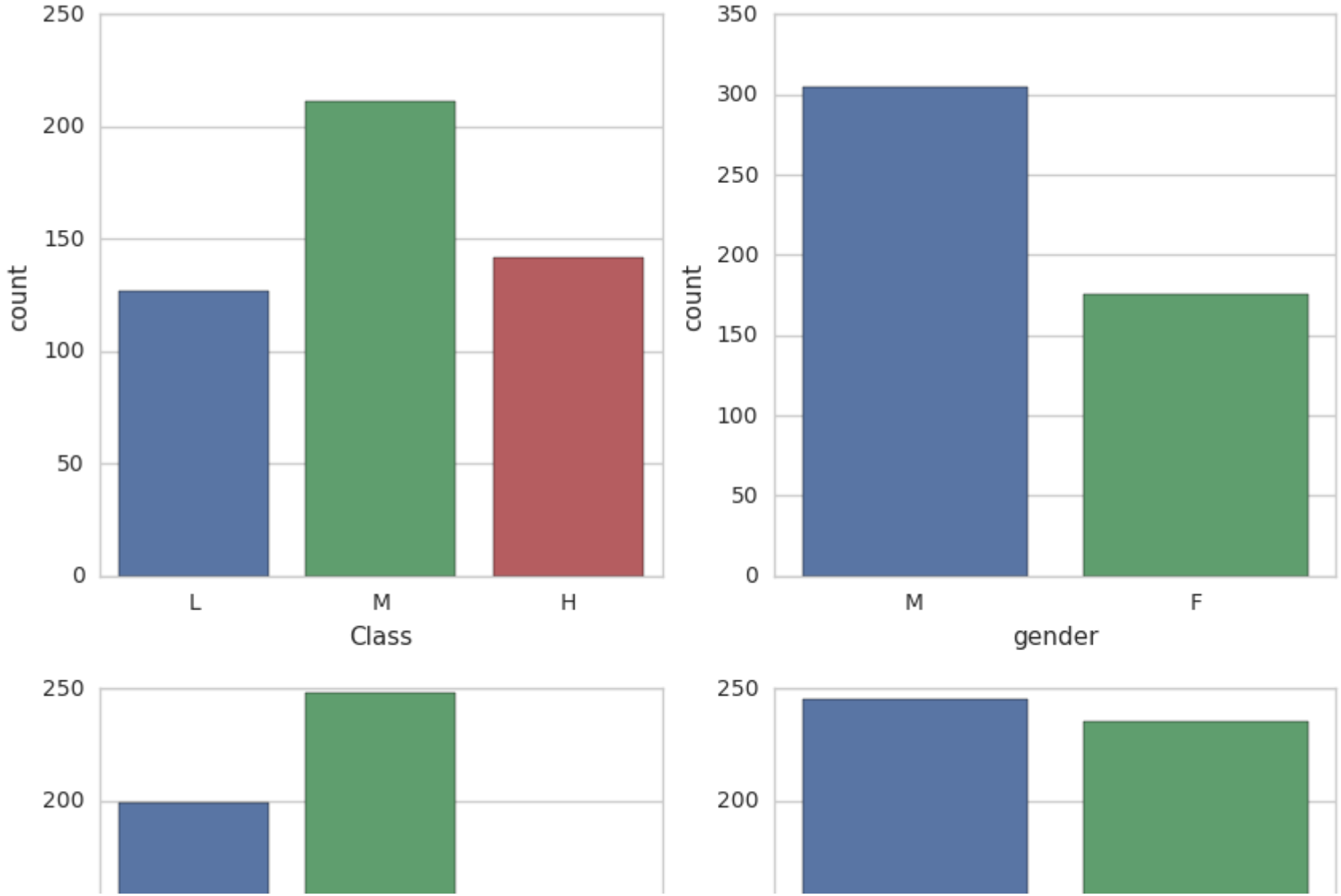
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 480 entries, 0 to 479
Data columns (total 17 columns):
gender                      480 non-null object
NationalITy                 480 non-null object
PlaceofBirth                480 non-null object
StageID                     480 non-null object
GradeID                     480 non-null object
SectionID                   480 non-null object
Topic                       480 non-null object
Semester                    480 non-null object
Relation                    480 non-null object
raisedhands                 480 non-null int64
VisITedResources            480 non-null int64
AnnouncementsView           480 non-null int64
Discussion                  480 non-null int64
ParentAnsweringSurvey       480 non-null object
ParentschoolSatisfaction    480 non-null object
StudentAbsenceDays          480 non-null object
Class                       480 non-null object
dtypes: int64(4), object(13)
memory usage: 63.8+ KB
```

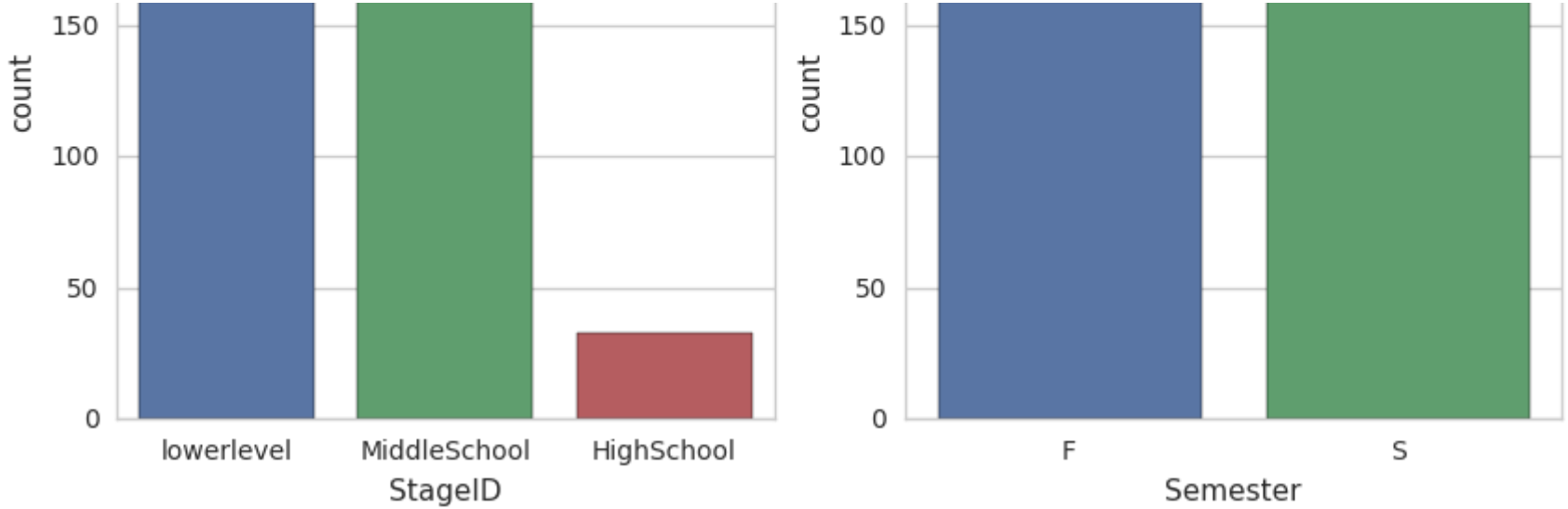
In [4]:
```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
3 sns.set_style('whitegrid')
4 %matplotlib inline
```

I'll start with visualizing 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 I can deduce from that.

In [5]:
```
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])
```

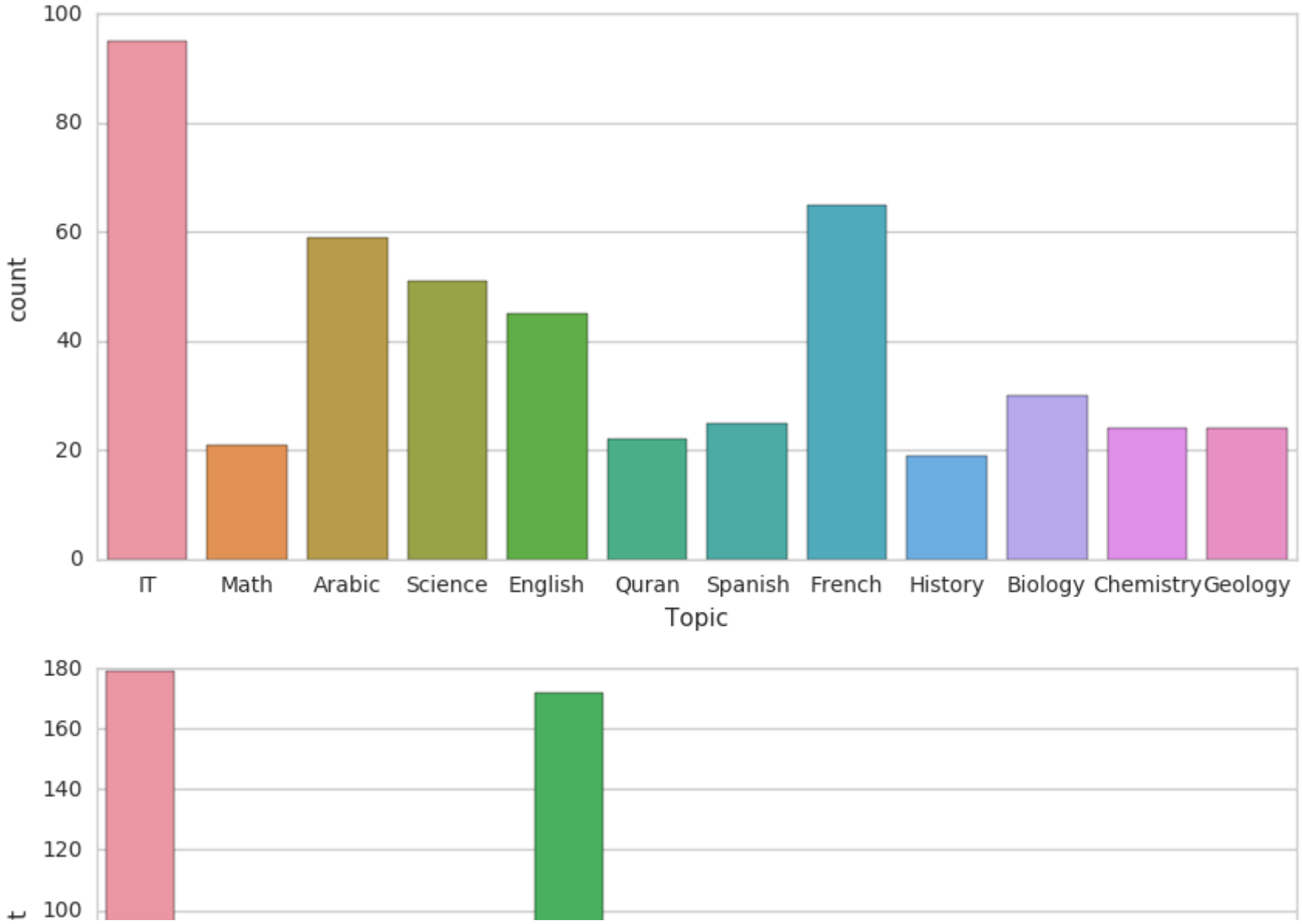
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x7f247f249780>

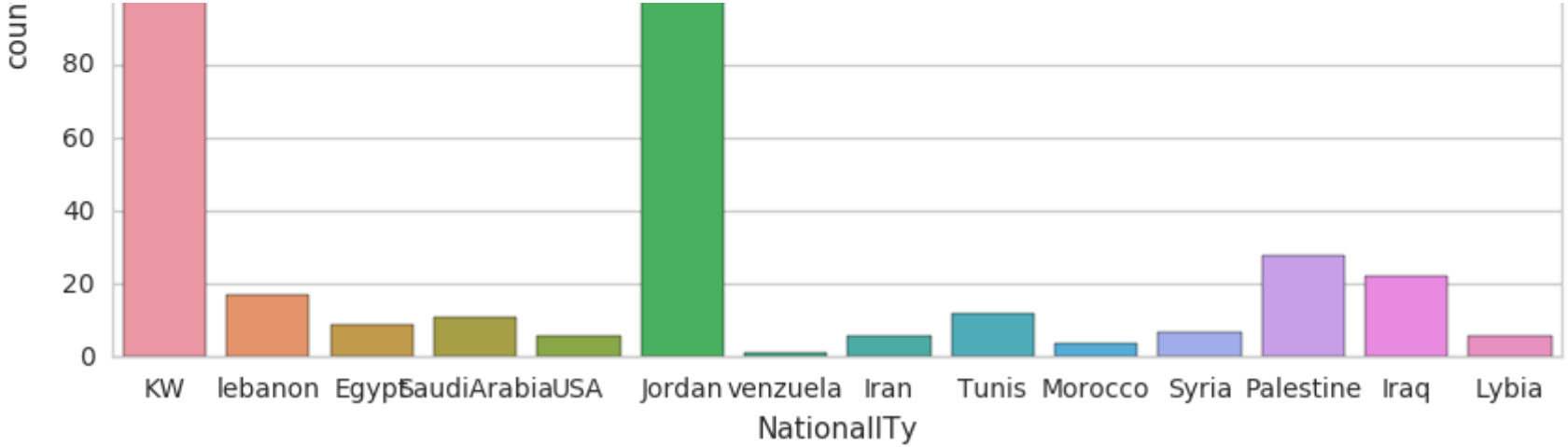



150
100
50
0
count
lowerlevel
MiddleSchool
HighSchool
StageID
150
100
50
0
count
F
S
Semester

In [6]:
```
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[6]: <matplotlib.axes._subplots.AxesSubplot at 0x7f247f203ac8>



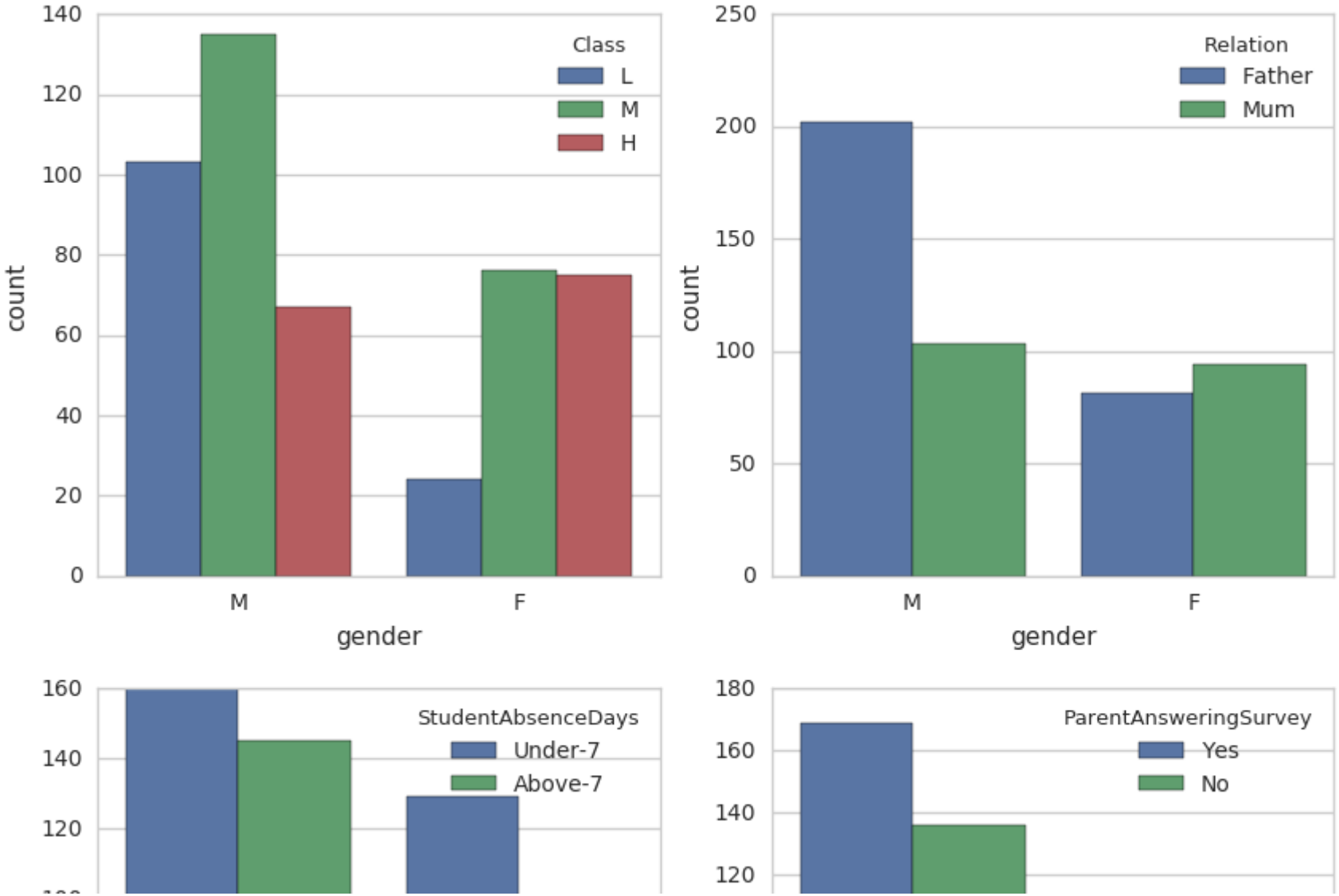


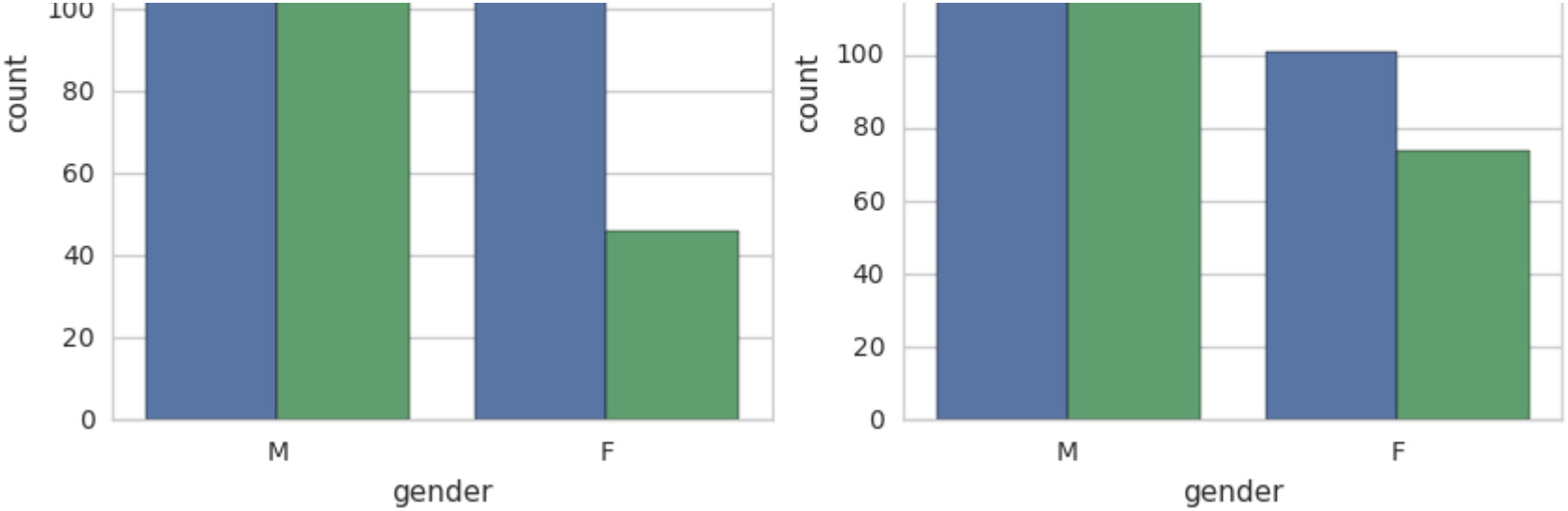
- Most of these countries are in the middle east(Islamic states), perhaps this explains the gender disparity

Next I will look at some categorical features in relation to each other, to see what insights that could possibly read

In [7]:
```
fig, axarr  = plt.subplots(2,2,figsize=(10,10))
sns.countplot(x='gender', hue='Class', data=data, ax=axarr[0,0], order=['M','F'], hue_order=['L','M','H'])
sns.countplot(x='gender', hue='Relation', data=data, ax=axarr[0,1], order=['M','F'])
sns.countplot(x='gender', hue='StudentAbsenceDays', data=data, ax=axarr[1,0], order=['M','F'])
sns.countplot(x='gender', hue='ParentAnsweringSurvey', data=data, ax=axarr[1,1], order=['M','F'])
```

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f247c6280b8>

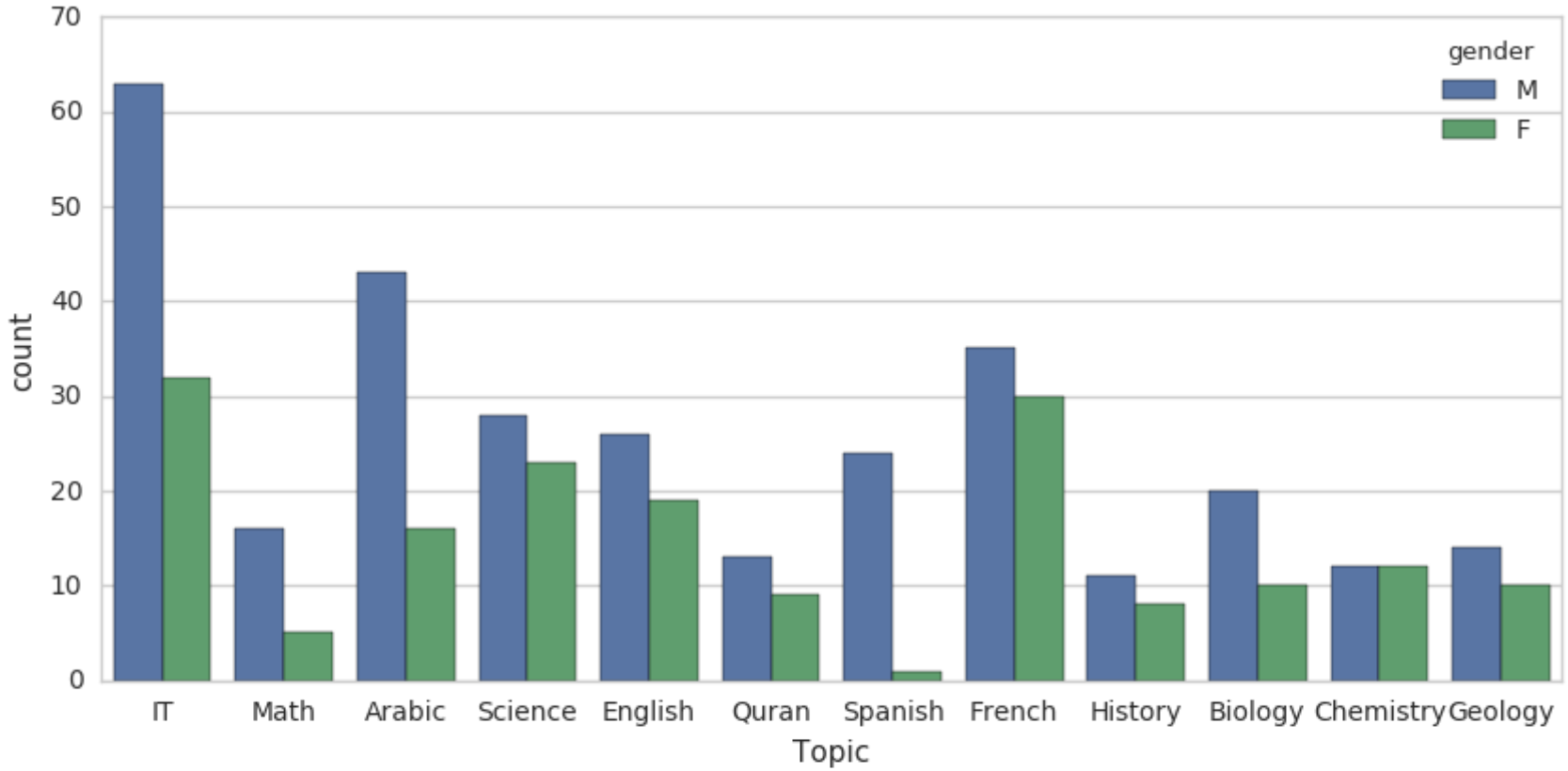


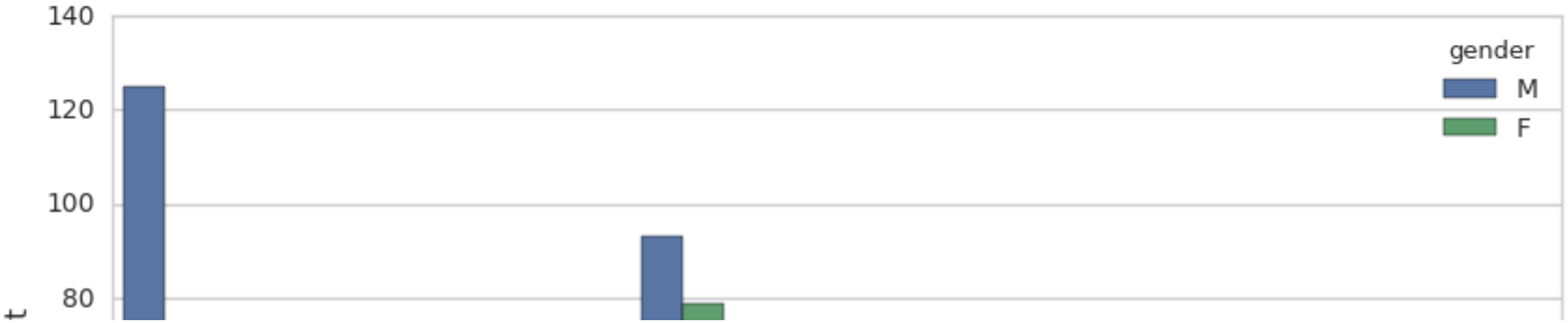


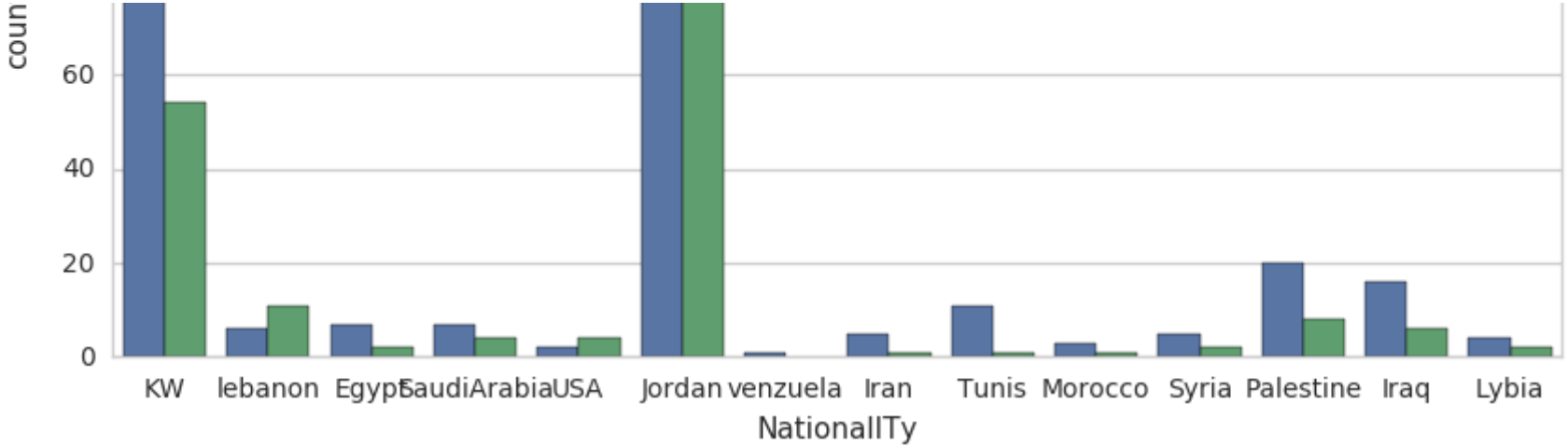
- Girls seem to have performed better than boys
- In the case of girls, mothers seem to be more interested in their education than fathers
- Girls had much better attendance than boys

In [8]:
```
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[8]: <matplotlib.axes._subplots.AxesSubplot at 0x7f247c4ed2b0>



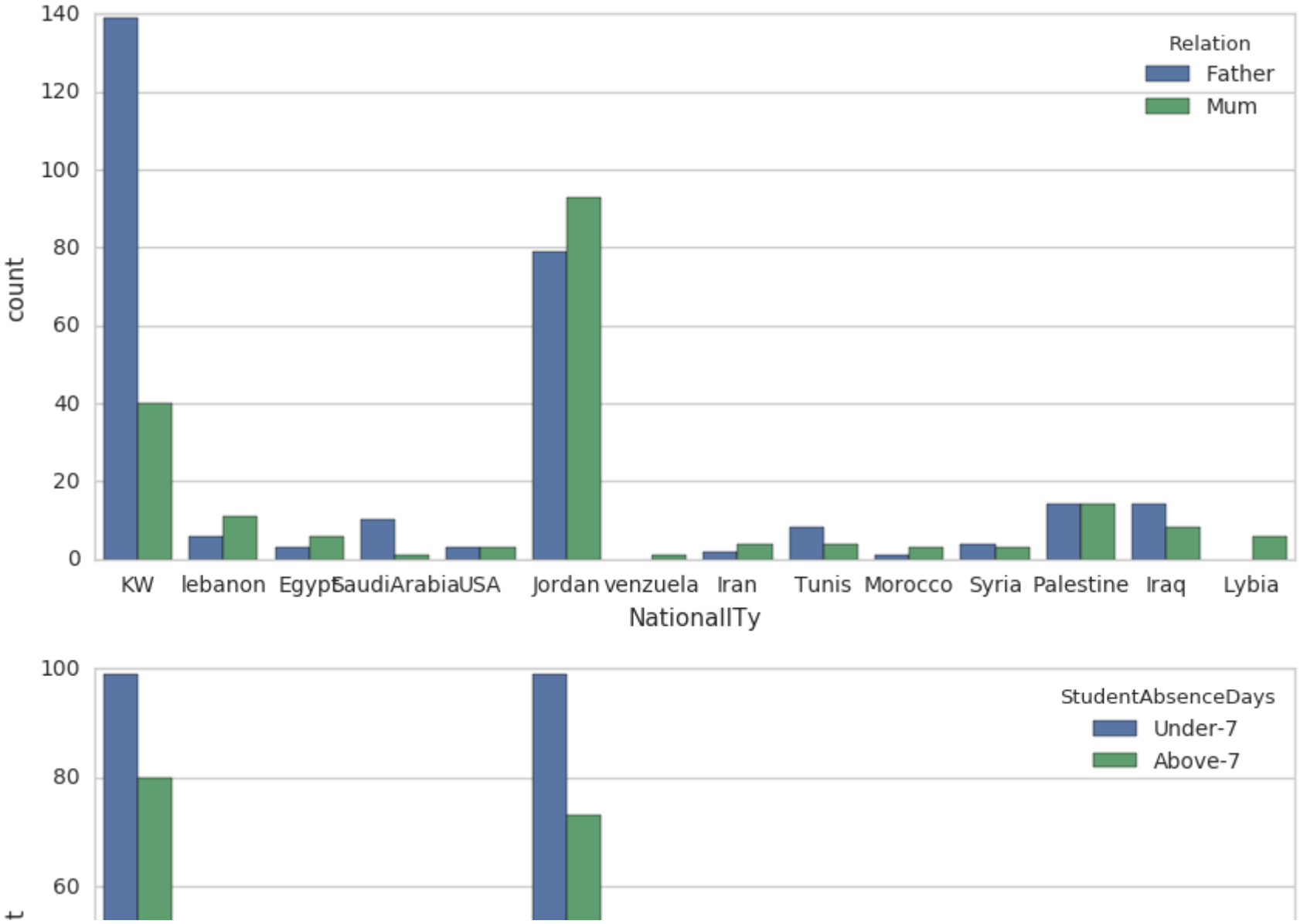


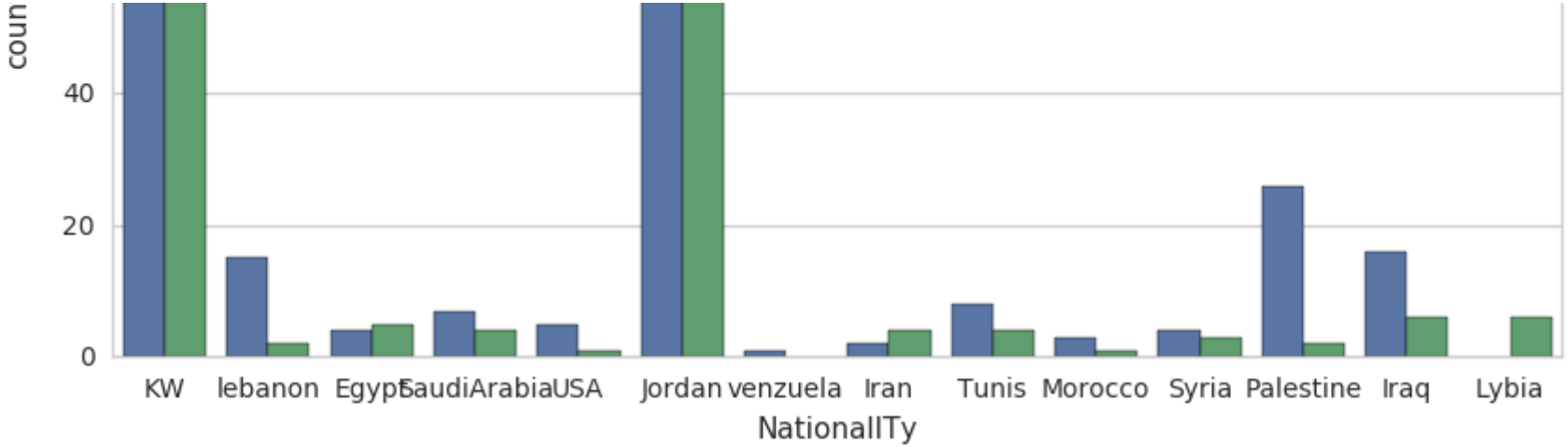


- No apparent gender bias when it comes to subject/topic choices, we cannot conclude that girls performed better because they perhaps took less technical subjects
- Gender disparity holds even at a country level. May just be as a result of the sampling.

In [9]:
```
fig, (axis1, axis2)  = plt.subplots(2, 1,figsize=(10,10))
sns.countplot(x='NationalITy', hue='Relation', data=data, ax=axis1)
sns.countplot(x='NationalITy', hue='StudentAbsenceDays', data=data, ax=axis2)
```

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f247bb9acc0>

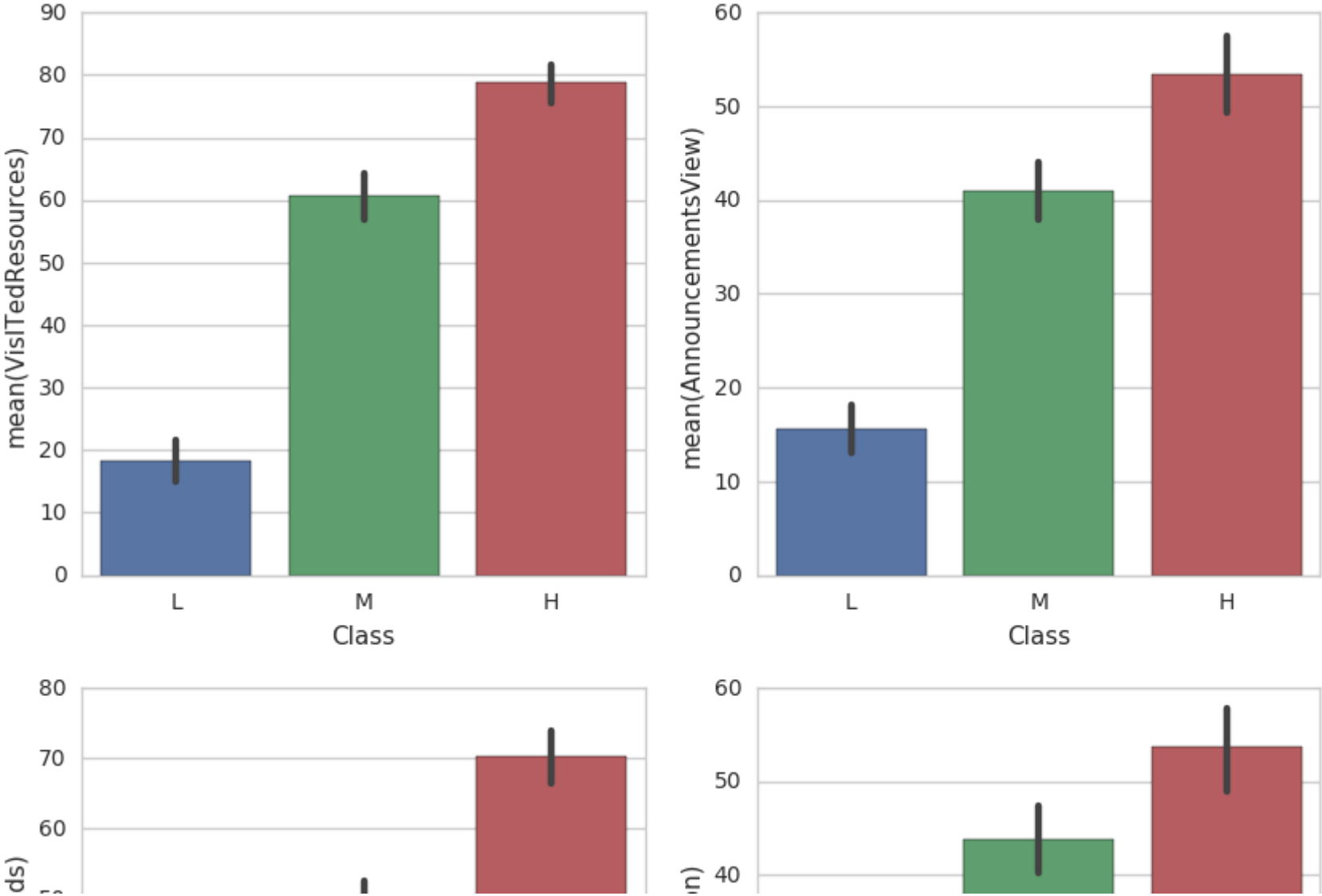


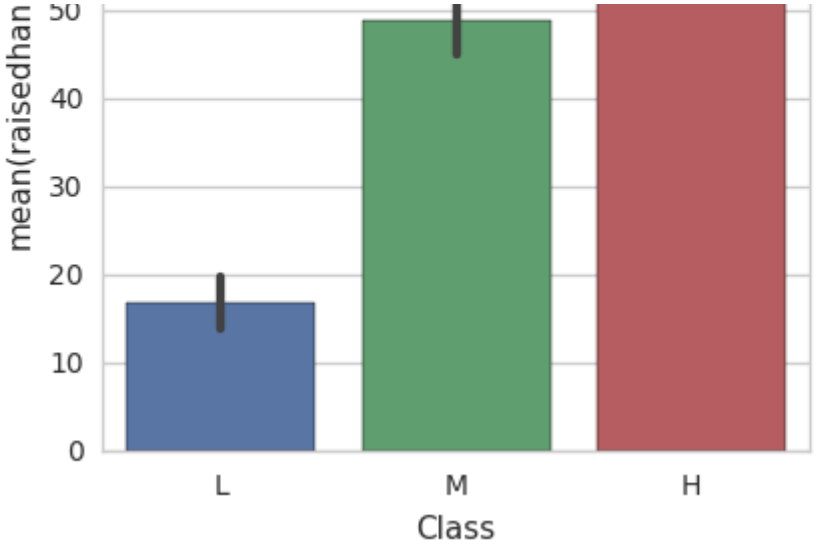


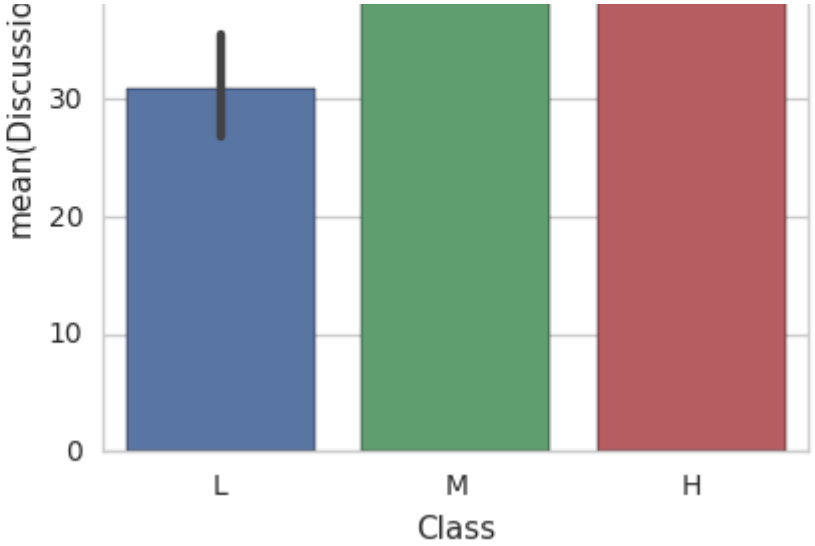
Now I am moving on to visualizing categorical features with numerical features.

In [10]:
```
fig, axarr  = plt.subplots(2,2,figsize=(10,10))
sns.barplot(x='Class', y='VisITedResources', data=data, order=['L','M','H'], ax=axarr[0,0])
sns.barplot(x='Class', y='AnnouncementsView', data=data, order=['L','M','H'], ax=axarr[0,1])
sns.barplot(x='Class', y='raisedhands', data=data, order=['L','M','H'], ax=axarr[1,0])
sns.barplot(x='Class', y='Discussion', data=data, order=['L','M','H'], ax=axarr[1,1])
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7f247b953278>



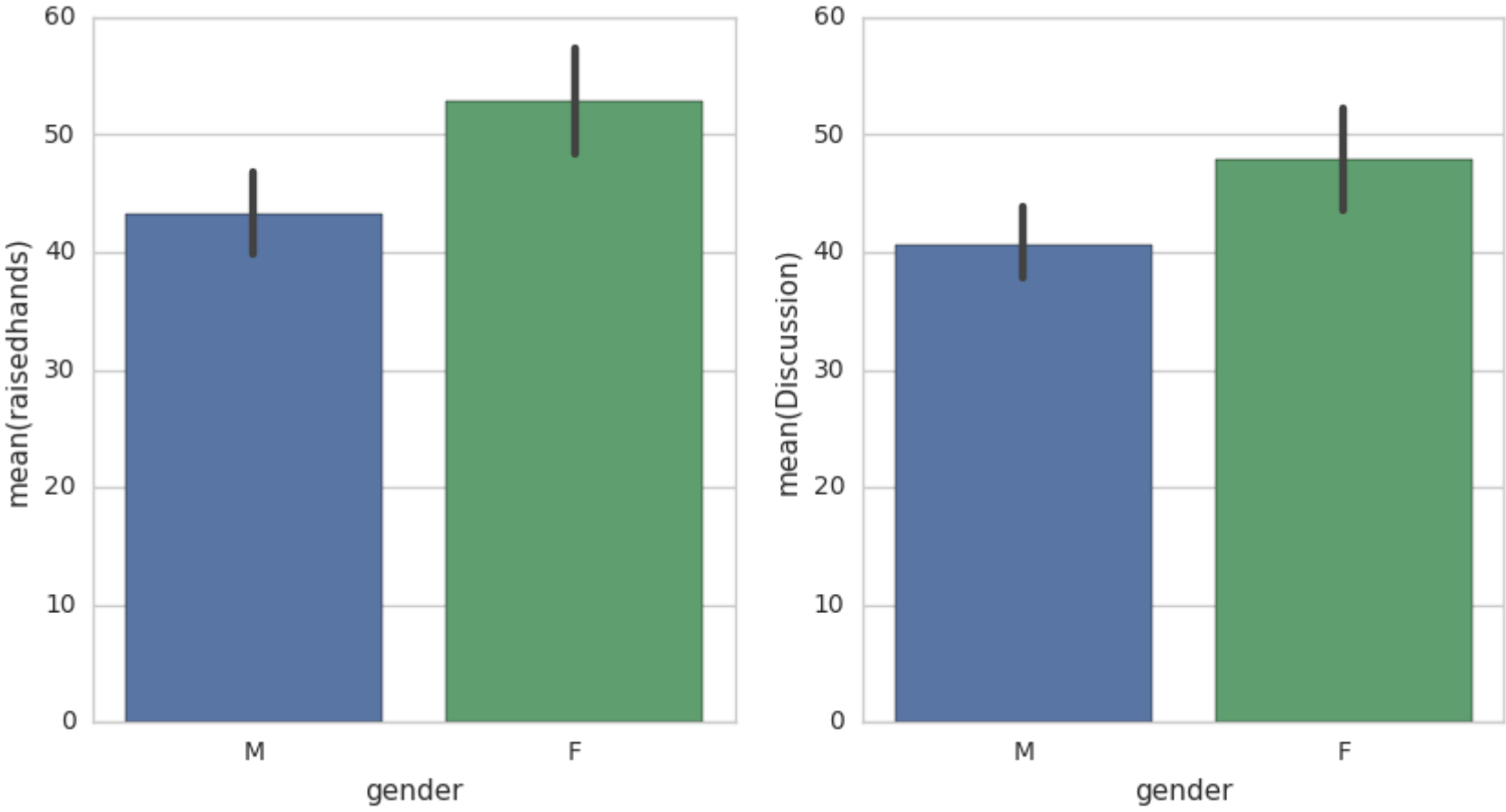




- As expected, those that participated more (higher counts in Discussion, raisedhands, AnnouncementViews, RaisedHands), performed better ...that thing about correlation and causation.

In [11]:
```
fig, (axis1,axis2) = plt.subplots(1,2,figsize=(10,5))
sns.barplot(x='gender', y='raisedhands', data=data, ax=axis1)
sns.barplot(x='gender', y='Discussion', data=data, ax=axis2)
```

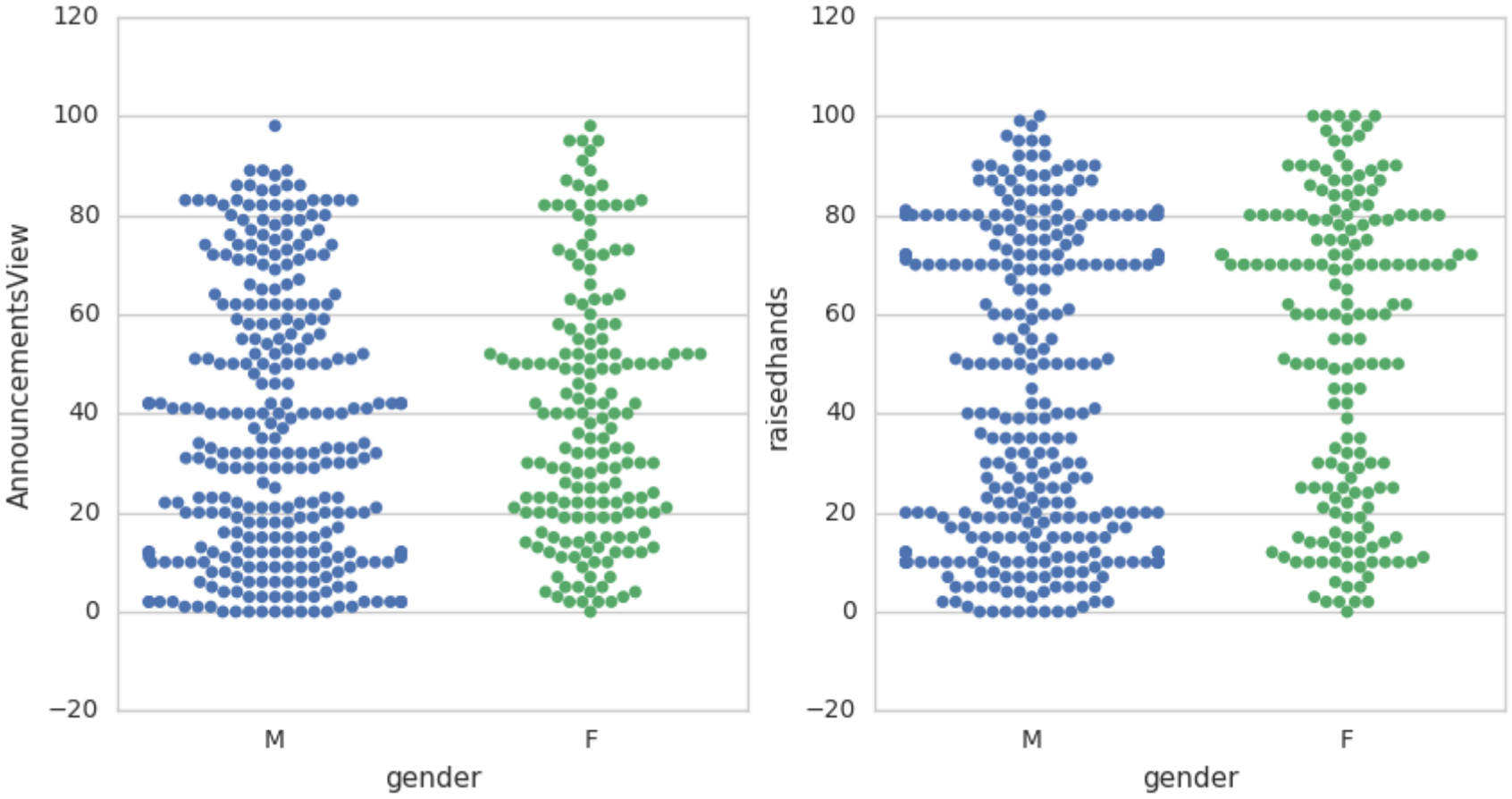
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x7f247b8829b0>



There are various other plots that help visualize Categorical vs Numerical data better.

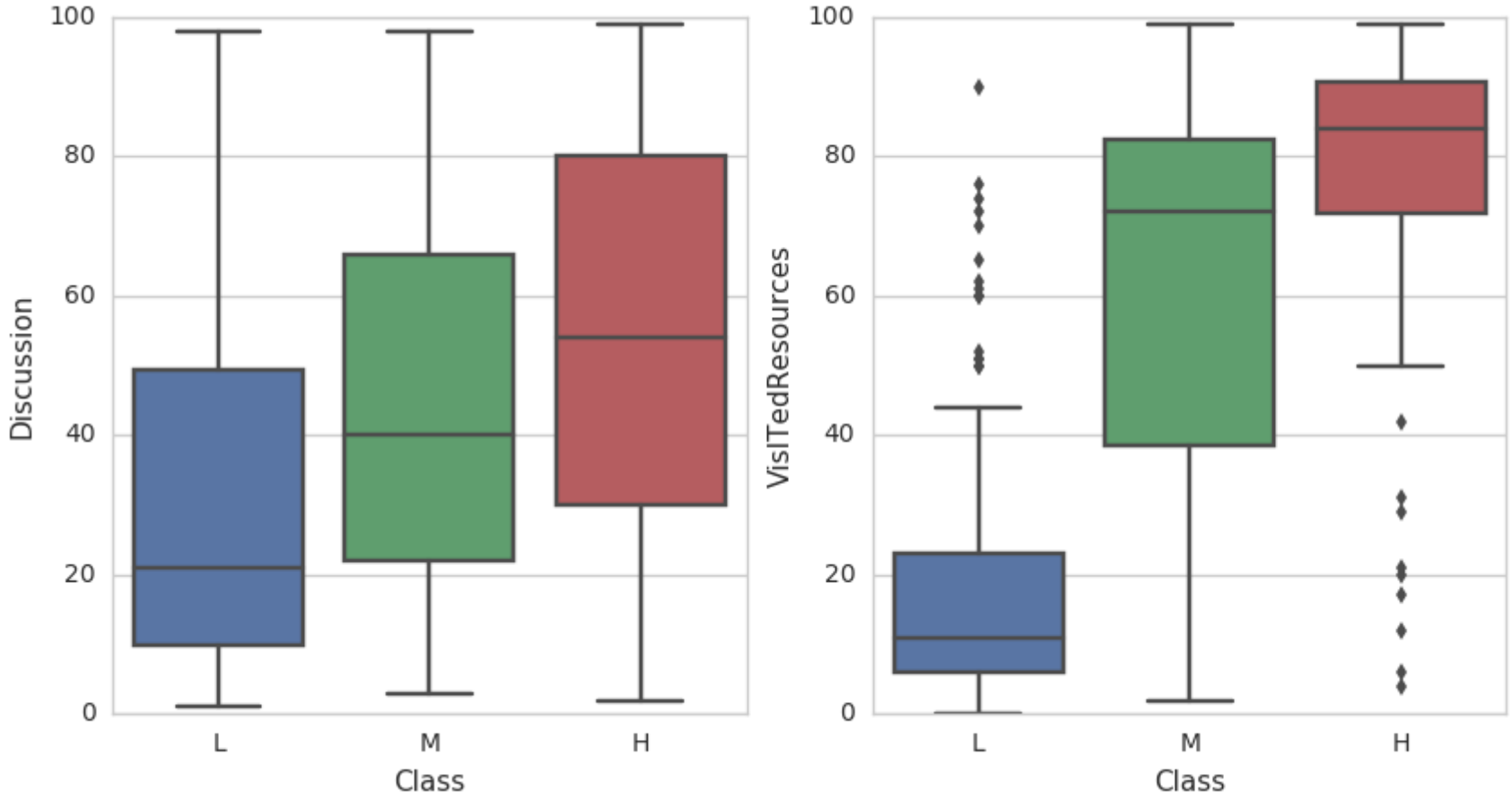
In [12]:
```
fig, (axis1, axis2)  = plt.subplots(1, 2,figsize=(10,5))
sns.swarmplot(x='gender', y='AnnouncementsView', data=data, ax=axis1)
sns.swarmplot(x='gender', y='raisedhands', data=data, ax=axis2)
```

Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f247b74b2b0>



In [13]:
```
fig, (axis1, axis2)  = plt.subplots(1, 2,figsize=(10,5))
sns.boxplot(x='Class', y='Discussion', data=data, order=['L','M','H'], ax=axis1)
sns.boxplot(x='Class', y='VisITedResources', data=data, order=['L','M','H'], ax=axis2)
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2479ef46d8>

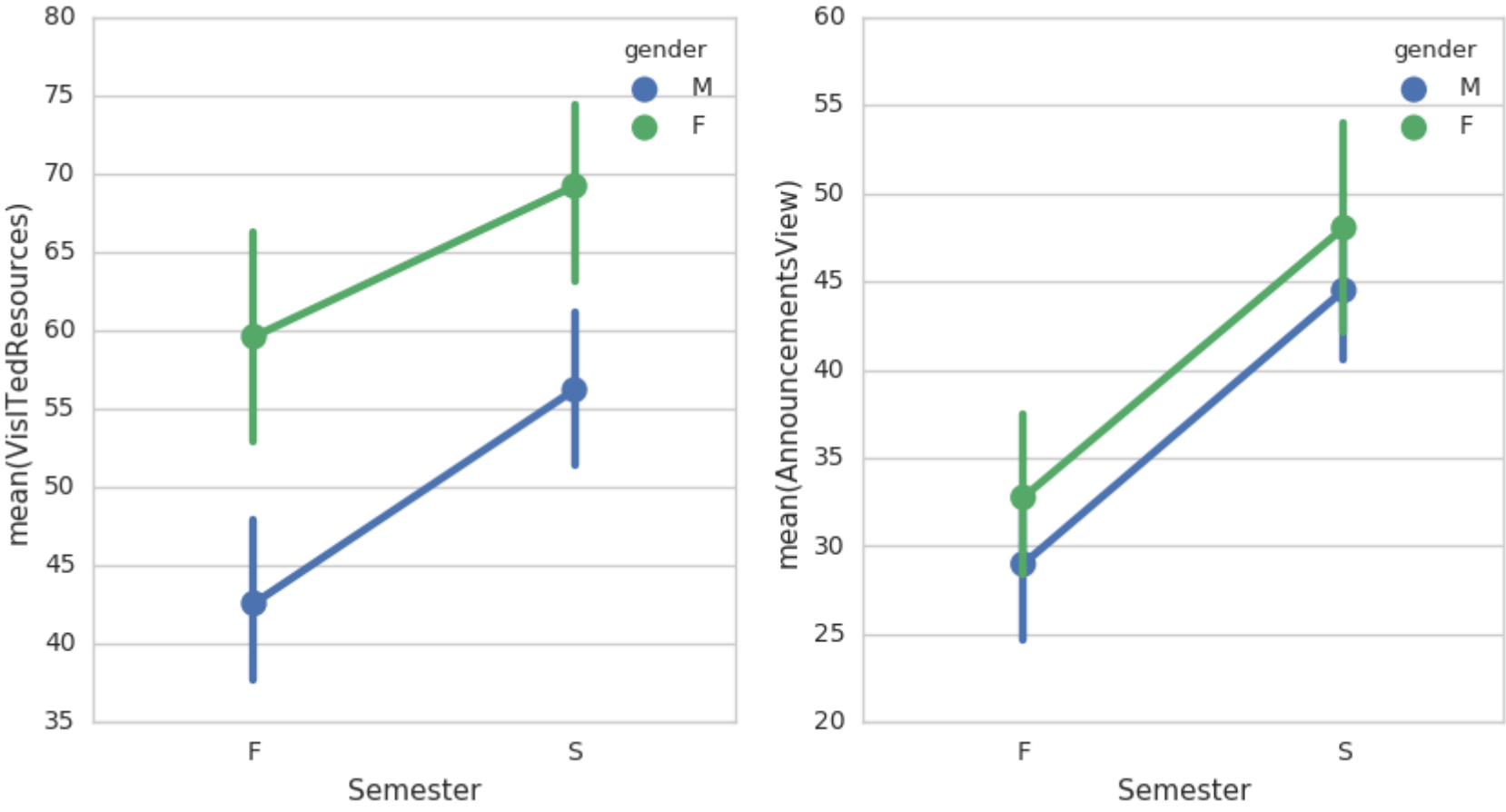


The boxplot the three quartile values of the distribution along with extreme values. The “whiskers” extend to points that lie within 1.5 IQRs of the lower and upper quartile, and then observations that fall outside this range are displayed independently.

- The two plots above tell us that visiting the resources may not be as sure a path to performing well as discussions

In [14]:
```
fig, (axis1, axis2)  = plt.subplots(1, 2,figsize=(10,5))
sns.pointplot(x='Semester', y='VisITedResources', hue='gender', data=data, ax=axis1)
sns.pointplot(x='Semester', y='AnnouncementsView', hue='gender', data=data, ax=axis2)
```

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2479da5748>



pointplots make it very easy to see how relationships change from variable to variable as well as the confidence interval. The confidence interval is a range of values so defined that there is a specified probability that the value of a parameter lies within it.

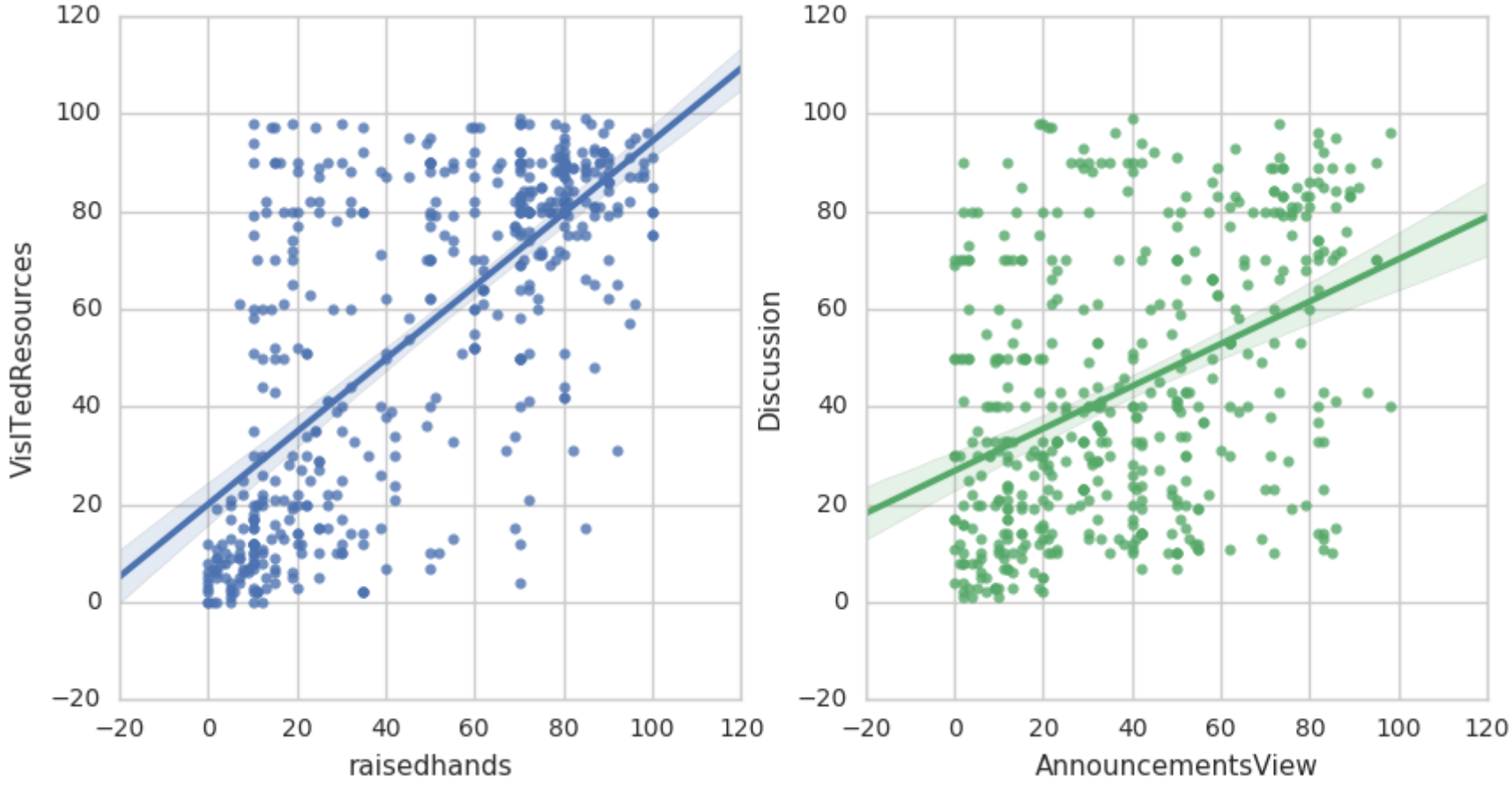
- In the case of both visiting resources and viewing announcements, students were more vigilant in the second semester, perhaps that last minute need to boost your final grade.

Moving on to plots to visualize relationships between numerical features.

In [15]:
```
fig, (axis1, axis2)  = plt.subplots(1, 2,figsize=(10,5))
sns.regplot(x='raisedhands', y='VisITedResources', data=data, ax=axis1)
sns.regplot(x='AnnouncementsView', y='Discussion', data=data, ax=axis2)

```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2479c9c748>



- There does not appear to be much of a linear relationship between the numerical features.

In [16]:
```
1
```