In [120]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import collections
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import chi2_contingency
import researchpy as rp
```

In [121]:

```
#from google.colab import files
#uploaded = files.upload()
```

In [122]:

```
data01 = pd.read_csv(r"datasets/combined.csv")
print(data01.columns)
data01.head()
```

```
Index(['__id__', 'birthYear', 'citySize', 'culturalBackground', 'e
ducation',
       'gender', 'gender2', 'id', 'internetExperience', 'internetF
requency',
       'internetSkill', 'item1ratevalue', 'item2ratevalue', 'mturk
ID',
       'nationality', 'occupation', 'ratingType'],
      dtype='object')
```

Out[122]:

	__id__	birthYear	citySize	culturalBackground	education	gender	ge
0	08TV7jhCOUVRn5wtJ78g	1974	Medium	["Hispanic"]	3	male	
1	16lmLHG1EDrWKdAR8Zgo	1996	Medium	["European"]	3	female	
2	1AUieolpuyHOLaT487Ln	1988	Medium	["African"]	3	male	
3	1K8aLOtQXoNT0ww0pJdU	1990	Medium	["European"]	1	male	
4	21FVW9tKOwjSuJPcVmcW	1990	Medium	["Hispanic"]	2	male	

In [123]:
```
data00 = pd.read_csv(r"datasets/ratingsonly.csv")
print(data00.columns)
data00.head()
```

```
Index(['__id__', 'category', 'id', 'itemRated', 'rateValue', 'rati
ngType'], dtype='object')
```

Out[123]:

	__id__	category	id	itemRated	rateValue	rating
0	03PSoJeZsjU03xRSAucA	1	RREvTRs89ZVldIZ1Alya	cat1item1	5	5
1	0WiiiEZYQg2nmg7hnYm8	2	1K8aLOtQXoNT0ww0pJdU	cat2item3	5	5-
2	0epdzSODx9o4yUPPiwxs	1	6FNW5HNxWk3jG7Luf1kw	cat1item1	4	I
3	0rC2LqYrvjsncUi9Q63O	1	cF8fWPqsK3ellkFva9nt	cat1item1	5	I
4	10EXrgK8dBKzqkU9zfmg	1	gfys04ks931I9aSU0EFe	cat1item1	4	5-

In [124]:
```
data01.head()
```

Out[124]:

	__id__	birthYear	citySize	culturalBackground	education	gender	ge
0	08TV7jhCOUVRn5wtJ78g	1974	Medium	["Hispanic"]	3	male	
1	16lmLHG1EDrWKdAR8Zgo	1996	Medium	["European"]	3	female	
2	1AUieolpuyHOLaT487Ln	1988	Medium	["African"]	3	male	
3	1K8aLOtQXoNT0ww0pJdU	1990	Medium	["European"]	1	male	
4	21FVW9tKOwjSuJPcVmcW	1990	Medium	["Hispanic"]	2	male	

In [125]:
```
conditions = [
    (data01['birthYear']>=1997),
    (data01['birthYear']>=1967) & (data01['birthYear']<1997),
    (data01['birthYear']<1967)
]
values = ['young', 'middle-aged','senior']
data01['ageGroup'] = np.select(conditions, values)
```

In [126]:

```
conditions_2 = [
    (data01['internetSkill']>60),
    (data01['internetSkill']>=51) & (data01['internetSkill']<=60),
    (data01['internetSkill']>=33) & (data01['internetSkill']<=50)
]
values_2 = ['Expert', 'intermediate','beginner']
data01['IT_Skill_level'] = np.select(conditions_2, values_2)
```

In [127]:

```
data01.head()
```

Out[127]:

	__id__	birthYear	citySize	culturalBackground	education	gender	ge
0	08TV7jhCOUVRn5wtJ78g	1974	Medium	["Hispanic"]	3	male	
1	16lmLHG1EDrWKdAR8Zgo	1996	Medium	["European"]	3	female	
2	1AUieolpuyHOLaT487Ln	1988	Medium	["African"]	3	male	
3	1K8aLOtQXoNT0ww0pJdU	1990	Medium	["European"]	1	male	
4	21FVW9tKOwjSuJPcVmcW	1990	Medium	["Hispanic"]	2	male	

In [128]:

```
#data00.loc[(data00['ratingType'] == "5-star") & (data00.loc[data00
#list(category)_(ratingsystem)
```

In [129]:

```
#5-star
data00.loc[(data00['ratingType'] == '5-star') & (data00['category']
list01_star = data00['rateValue'].loc[(data00['ratingType'] == '5-s
list02_star = data00['rateValue'].loc[(data00['ratingType'] == '5-s
# Probability calculations
frequency1 = collections.Counter(list01_star)
frequency2 = collections.Counter(list02_star)
# printing the frequency
print(dict(frequency1))
print(dict(frequency2))
```

```
{5: 18, 3: 6, 4: 9}
{4: 12, 3: 7, 5: 14}
```

In [130]:

```
#Category 1
star_avg01 = sum(list01_star)/len(list01_star)
print(star_avg01)
std01=np.std(list01_star)
print(std01)
#Category 2
star_avg02 = sum(list02_star)/len(list02_star)
print(star_avg02)
std02=np.std(list02_star)
print(std02)
```

```
4.363636363636363
0.77138921583987
4.212121212121212
0.769004699421183
```

In [131]:

```
#5-point slider
data00.loc[(data00['ratingType'] == '5-point slider') & (data00['ca
list01_slider = data00['rateValue'].loc[(data00['ratingType'] == '5
list02_slider = data00['rateValue'].loc[(data00['ratingType'] == '5
frequency3 = collections.Counter(list01_slider)
frequency4 = collections.Counter(list02_slider)
# printing the frequency
print(dict(frequency3))
print(dict(frequency4))
len(list01_slider)
len(list02_slider)
```

```
{4: 23, 5: 19}
{5: 21, 4: 19}
```

Out[131]:

```
40
```

In [132]:

```
#Category 1
slider_avg01 = sum(list01_slider)/len(list01_slider)
print(slider_avg01)
std03=np.std(list01_slider)
print(std03)
#Category 2
star_avg02 = sum(list02_slider)/len(list02_slider)
print(star_avg02)
std04=np.std(list02_slider)
print(std04)
```

```
4.4523809523809526
0.497727260961116
4.525
0.4993746088859545
```

In [133]:

```
#Emoji
data00.loc[(data00['ratingType'] == 'Emoji') & (data00['category']
list01_emoji= data00['rateValue'].loc[(data00['ratingType'] == 'Emo
list02_emoji = data00['rateValue'].loc[(data00['ratingType'] == 'Em
frequency5 = collections.Counter(list01_emoji)
frequency6 = collections.Counter(list02_emoji)
# printing the frequency
print(dict(frequency5))
print(dict(frequency6))
```

```
{4: 15, 5: 17}
{5: 19, 4: 13, 3: 1}
```

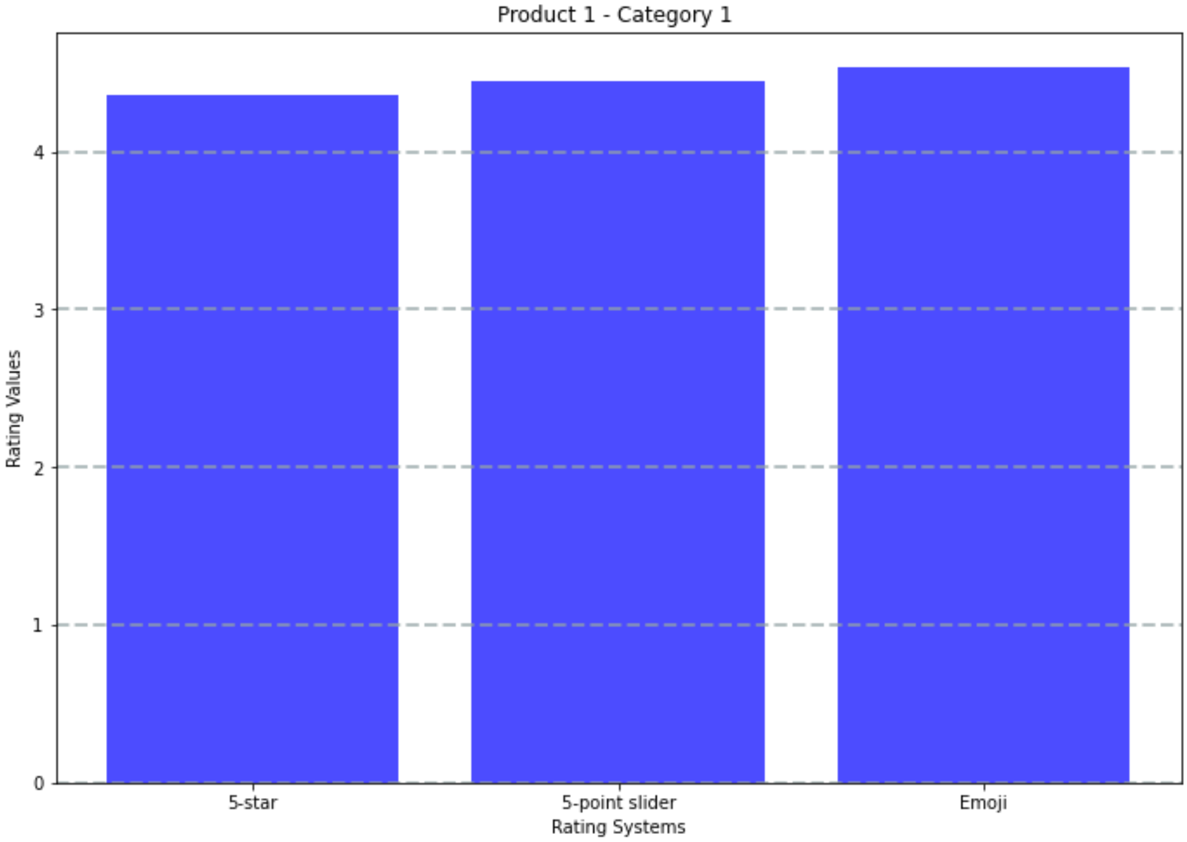
In [134]:

```
#Category 1
emoji_avg01 = sum(list01_emoji)/len(list01_emoji)
print(emoji_avg01)
std05=np.std(list01_emoji)
print(std05)
#Category 2
emoji_avg02 = sum(list02_emoji)/len(list02_emoji)
print(emoji_avg02)
std06=np.std(list02_emoji)
print(std06)
```

```
4.53125
0.4990224819584785
4.545454545454546
0.5554637206007078
```

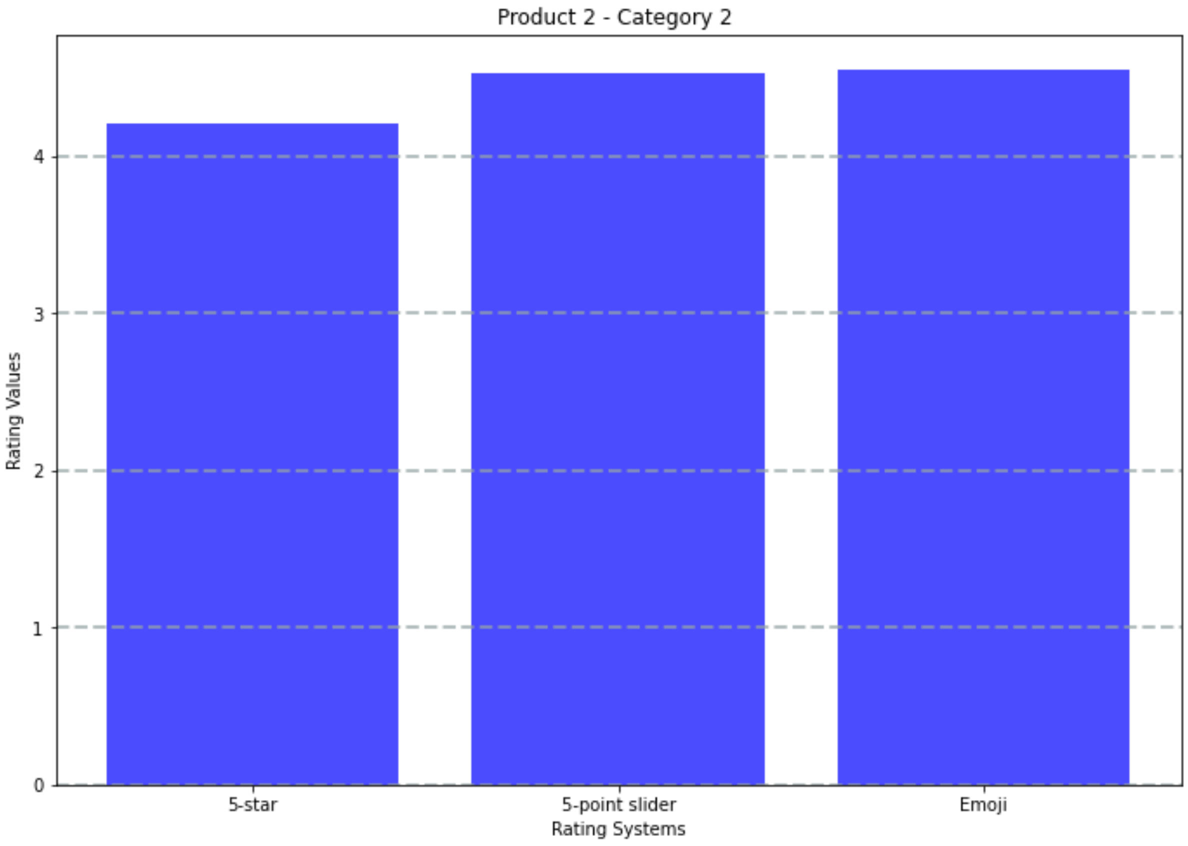
In [135]:

```
data = [[4.364, 4.445, 4.531],
        [4.212, 4.525, 4.545]]
fig = plt.figure(figsize =(9, 6))
ax = fig.add_axes([0,0,1,1])
rateS = ['5-star', '5-point slider', 'Emoji']
plt.bar(rateS, data[0], color='blue', alpha=0.7)
plt.grid(color='#95a5a6', linestyle='--', linewidth=2, axis='y', al
plt.xlabel('Rating Systems')
plt.ylabel('Rating Values')
plt.title('Product 1 - Category 1 ')
plt.show()
```



In [136]:

```
data = [[4.364, 4.445, 4.531],
        [4.212, 4.525, 4.545]]
fig = plt.figure(figsize =(9, 6))
ax = fig.add_axes([0,0,1,1])
rateS = ['5-star', '5-point slider', 'Emoji']
plt.bar(rateS, data[1], color='blue', alpha=0.7)
plt.grid(color='#95a5a6', linestyle='--', linewidth=2, axis='y', al
plt.xlabel('Rating Systems')
plt.ylabel('Rating Values')
plt.title('Product 2 - Category 2 ')
plt.show()
```



In [137]:

```
data01[['birthYear','education','item1ratevalue','item2ratevalue','
```

Out[137]:

```
birthYear         1998
education            3
item1ratevalue       5
item2ratevalue       5
internetSkill       75
dtype: int64
```

In [138]:
```
data01[['birthYear','education','item1ratevalue','item2ratevalue','
```

Out[138]:
```
birthYear         29
education          1
item1ratevalue     3
item2ratevalue     3
internetSkill     33
dtype: int64
```

In [139]:
```
data01[['internetSkill','internetExperience','internetFrequency','I
```

Out[139]:

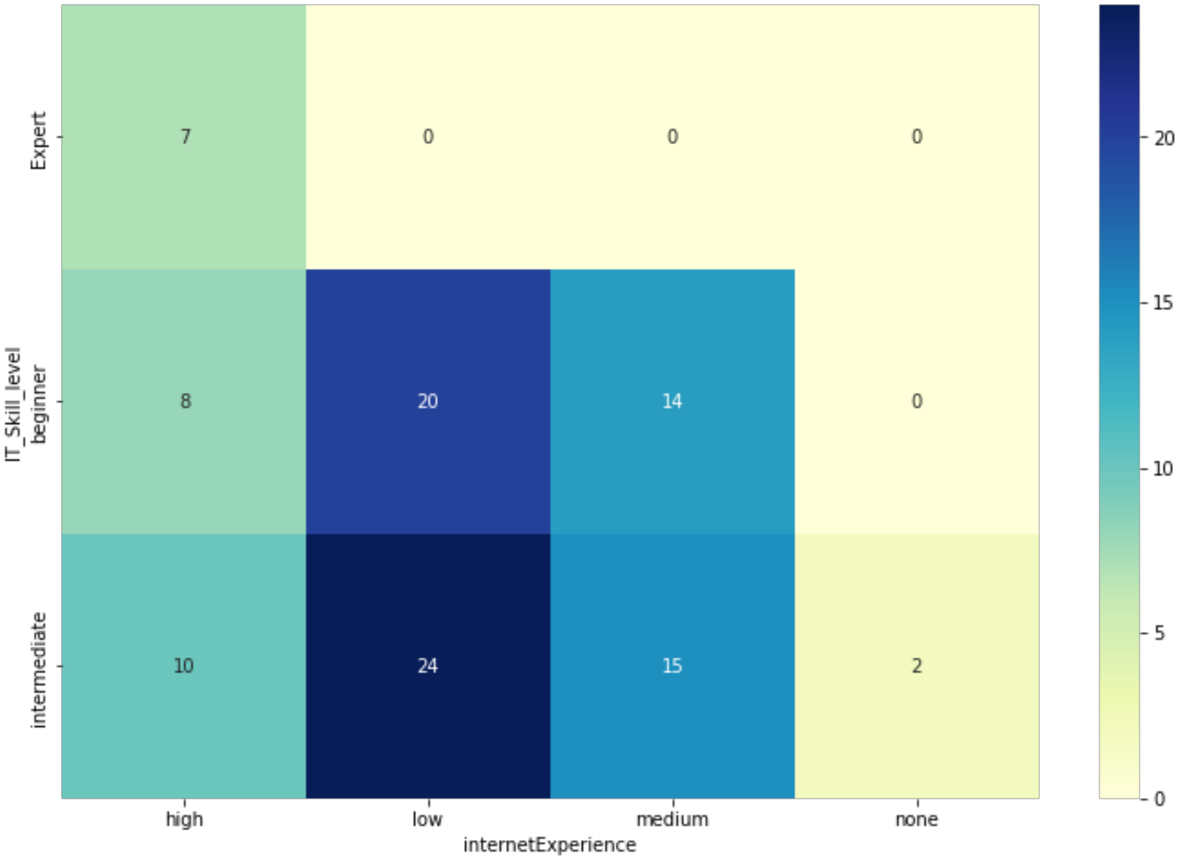
	internetSkill	internetExperience	internetFrequency	IT_Skill_level
0	67	high	daily	Expert
1	47	low	occasionally	beginner
2	52	low	monthly	intermediate
3	46	low	daily	beginner
4	54	medium	monthly	intermediate
...	...	...	...	...
95	36	low	occasionally	beginner
96	51	low	monthly	intermediate
97	57	medium	occasionally	intermediate
98	49	medium	daily	beginner
99	55	medium	daily	intermediate

100 rows × 4 columns

In [140]:

```
contigency_1= pd.crosstab(data01['IT_Skill_level'], data01['interne
print(contigency_1)
plt.figure(figsize=(12,8))
sns.heatmap(contigency_1, annot=True, cmap="YlGnBu")
plt.show()
```

```
internetExperience  high  low  medium  none
IT_Skill_level
Expert                 7    0       0     0
beginner               8   20      14     0
intermediate          10   24      15     2
```



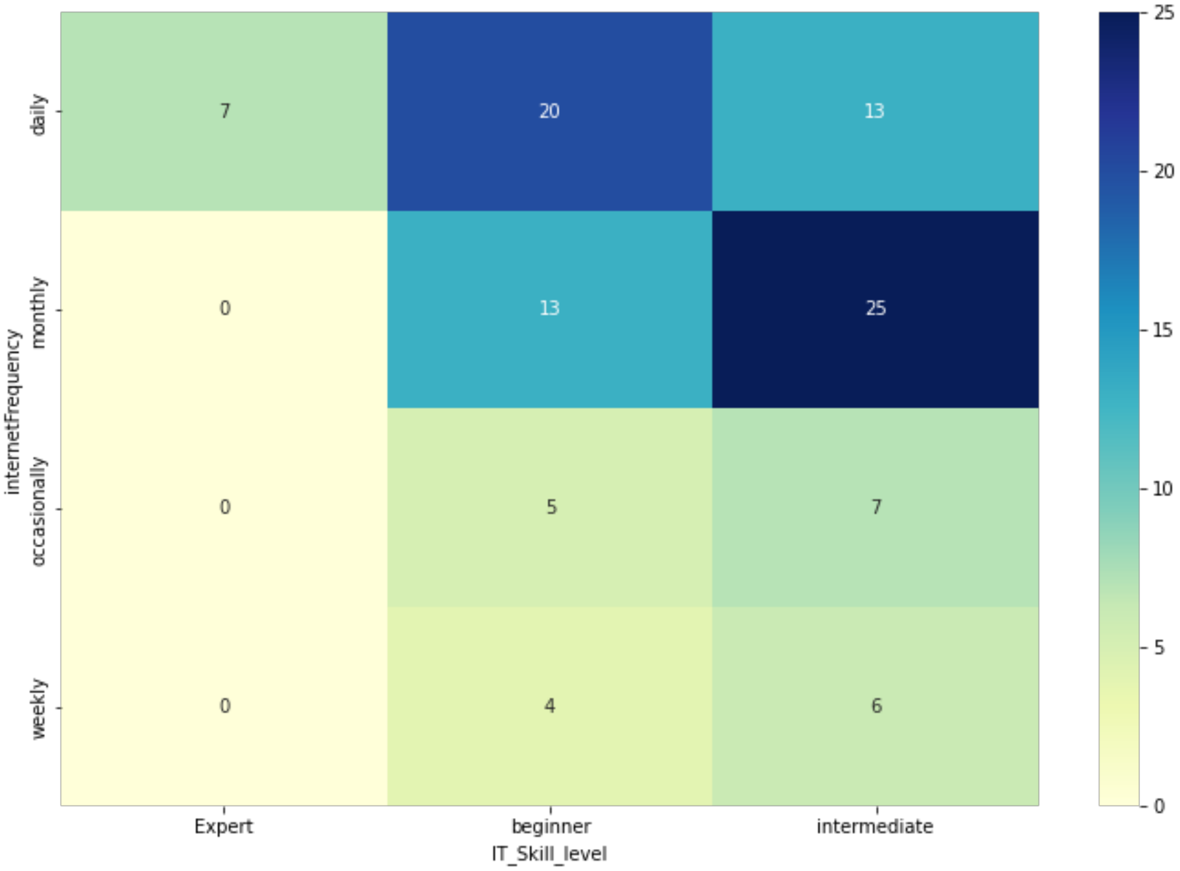
In [141]:

```
c1, p1, dof1, expected1 = chi2_contingency(contigency_1)
print('Null hypothesis is that IT_Skill_level and internetExperienc
print(c1)
print('This is the P-value')
print(round(p1,2))
print(dof1)
print(expected1)
#stats.chi2_contingency(contigency_1)
```

```
Null hypothesis is that IT_Skill_level and internetExperience is u
nrelated/independent. Confidence level 95%
24.478350587884055
This is the P-value
0.0
6
[[ 1.75  3.08  2.03  0.14]
 [10.5  18.48 12.18  0.84]
 [12.75 22.44 14.79  1.02]]
```

In [142]:
```
contigency_2= pd.crosstab(data01['internetFrequency'], data01['IT_S
print(contigency_2)
plt.figure(figsize=(12,8))
sns.heatmap(contigency_2, annot=True, cmap="YlGnBu")
plt.show()
```

```
IT_Skill_level     Expert  beginner  intermediate
internetFrequency
daily                   7        20            13
monthly                 0        13            25
occasionally            0         5             7
weekly                  0         4             6
```



In [143]:

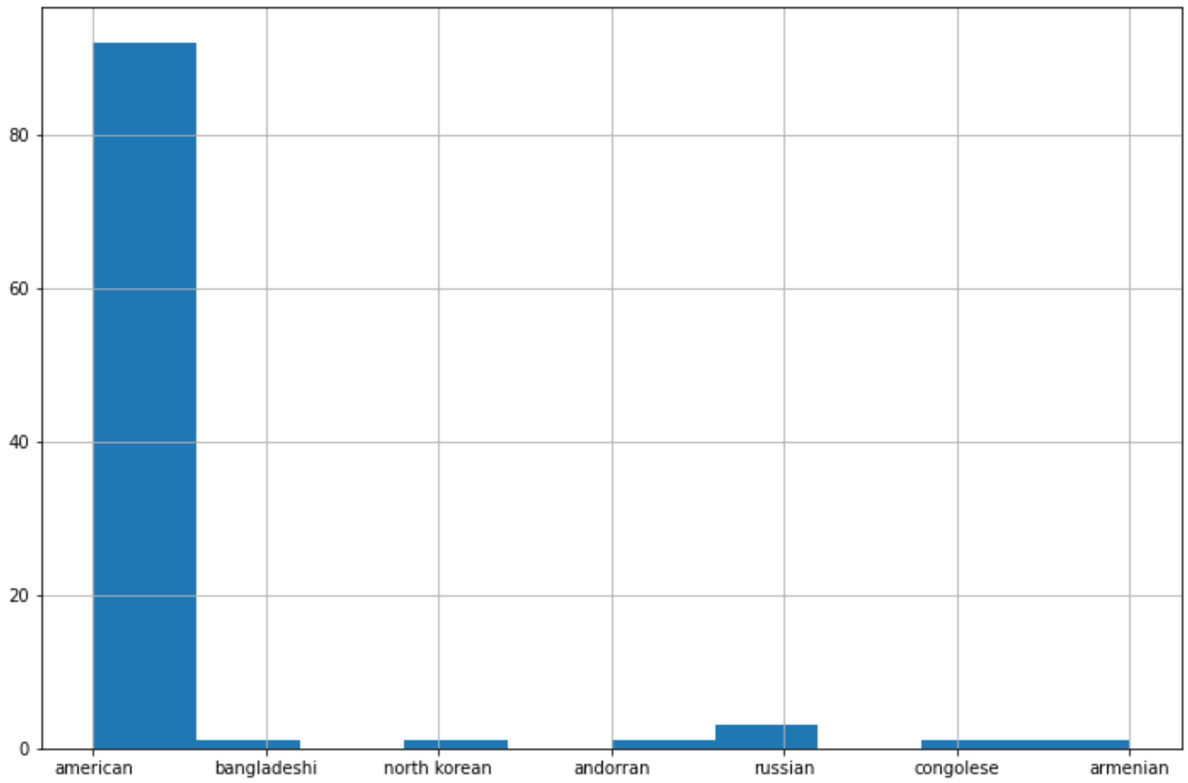
```
c2, p2, dof2, expected2 = chi2_contingency(contigency_2)
print('Null hypothesis is that IT_Skill_level and internetFrequency
print(c2)
print('This is the P-value')
print(round(p2,2))
print(dof2)
print(expected2)
```

```
Null hypothesis is that IT_Skill_level and internetFrequency is un
related/independent. Confidence level 95%
16.267752715121137
This is the P-value
0.01
6
[[ 2.8   16.8   20.4 ]
 [ 2.66 15.96 19.38]
 [ 0.84  5.04  6.12]
 [ 0.7   4.2   5.1 ]]
```

Nationality

In [144]:

```
fig01 = data01['nationality'].hist(figsize=(12,8))
```



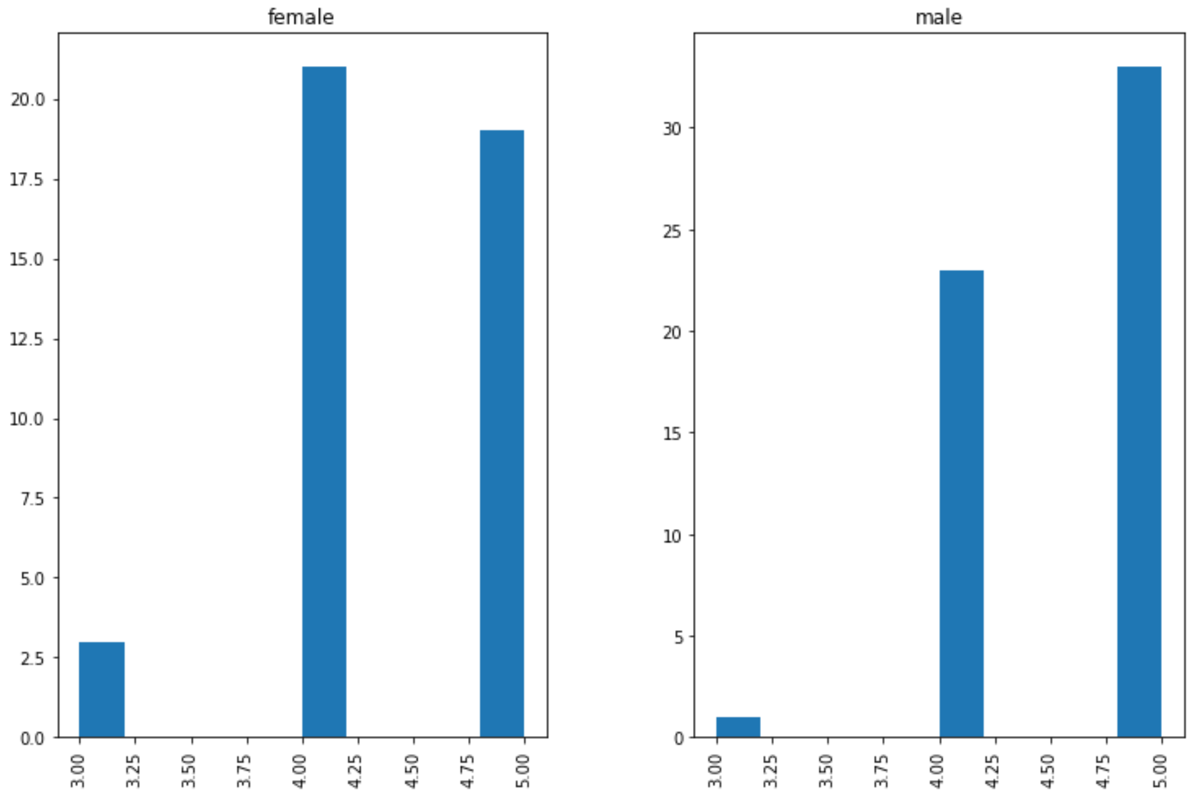
Rating vs Gender

In [145]:

```
data01['item1ratevalue'].hist(by=data01['gender'], figsize=(12, 8))
```

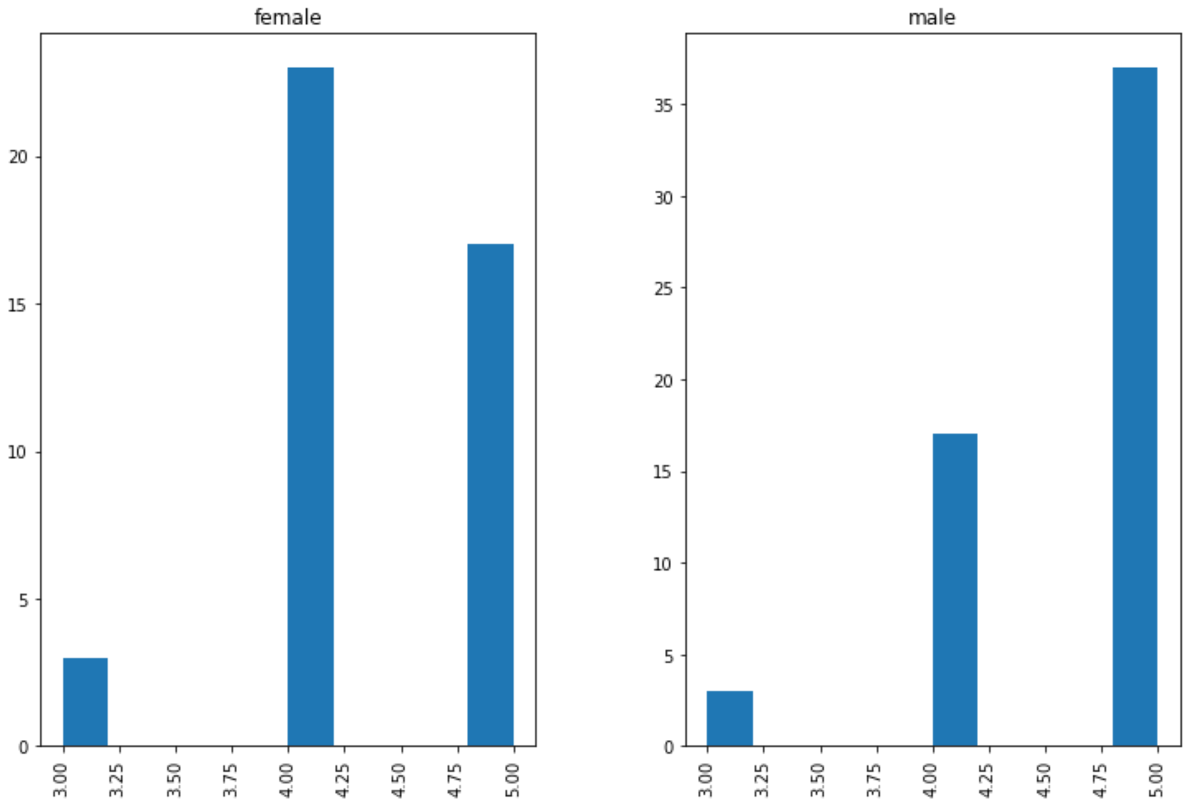
Out[145]:

```
array([<AxesSubplot:title={'center':'female'}>,
       <AxesSubplot:title={'center':'male'}>], dtype=object)
```



```
In [146]: data01['item2ratevalue'].hist(by=data01['gender'], figsize=(12, 8))
```

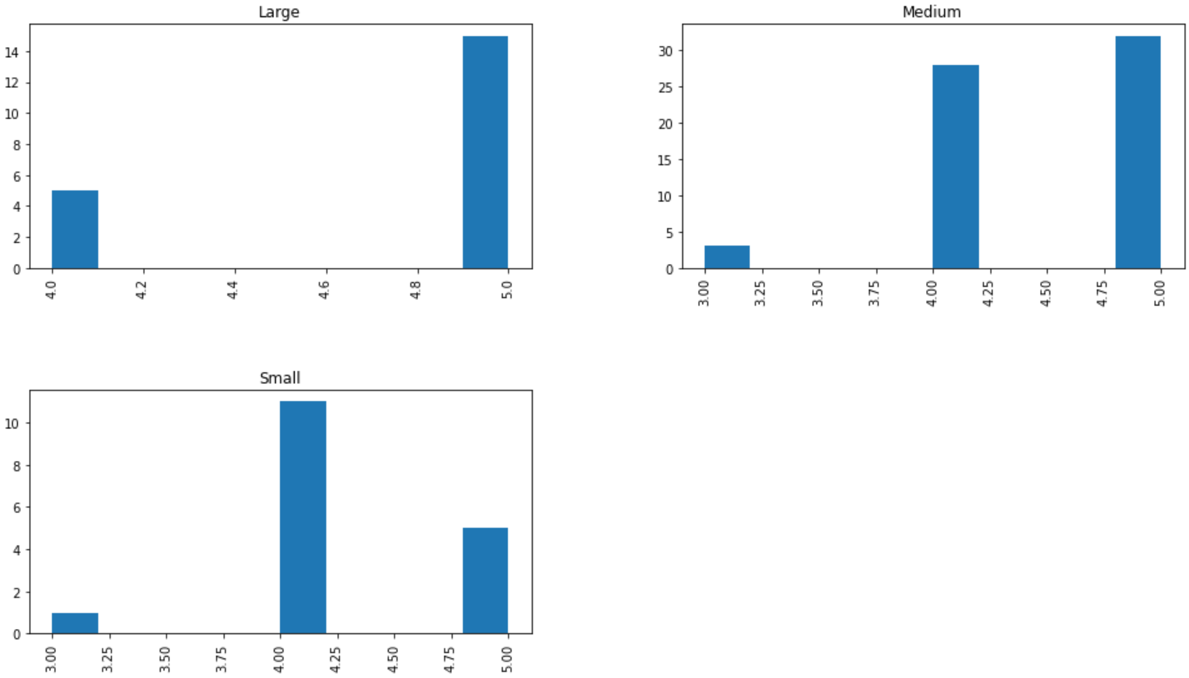
```
Out[146]: array([<AxesSubplot:title={'center':'female'}>,
                 <AxesSubplot:title={'center':'male'}>], dtype=object)
```



Ratings Vs CitySize

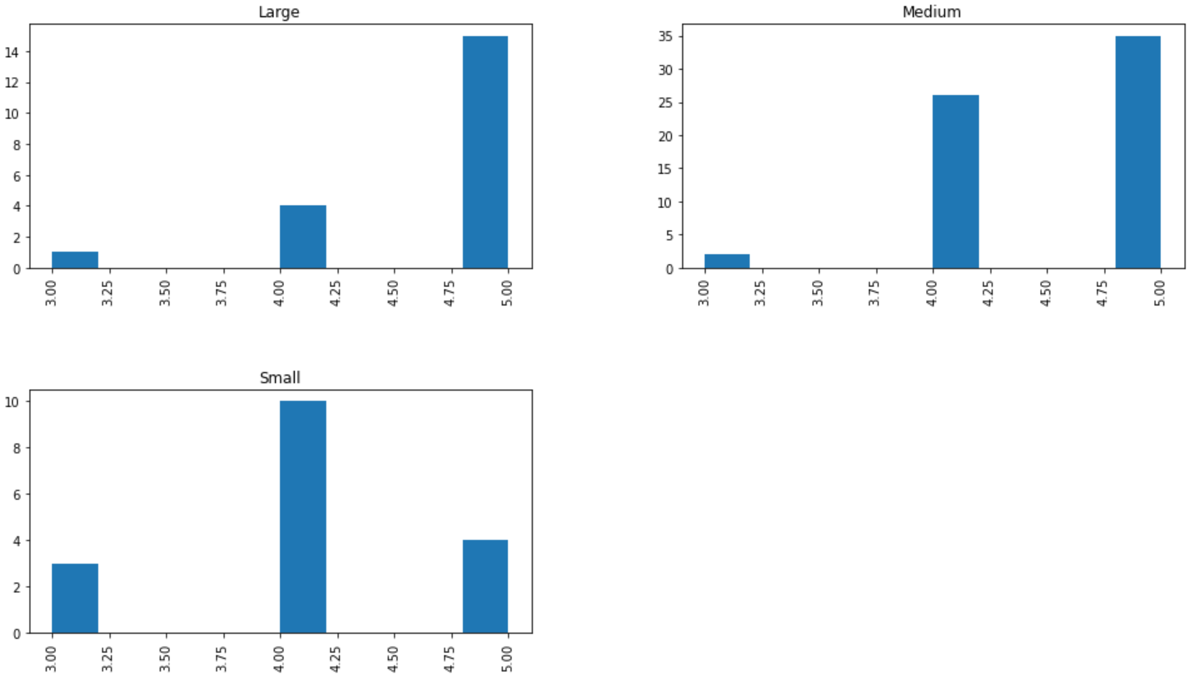
In [147]:
```
data01['item1ratevalue'].hist(by=data01['citySize'], figsize=(16, 9
```

Out[147]:
```
array([[<AxesSubplot:title={'center':'Large'}>,
        <AxesSubplot:title={'center':'Medium'}>],
       [<AxesSubplot:title={'center':'Small'}>, <AxesSubplot:>]],
      dtype=object)
```



In [148]:
```
data01['item2ratevalue'].hist(by=data01['citySize'], figsize=(16, 9
```

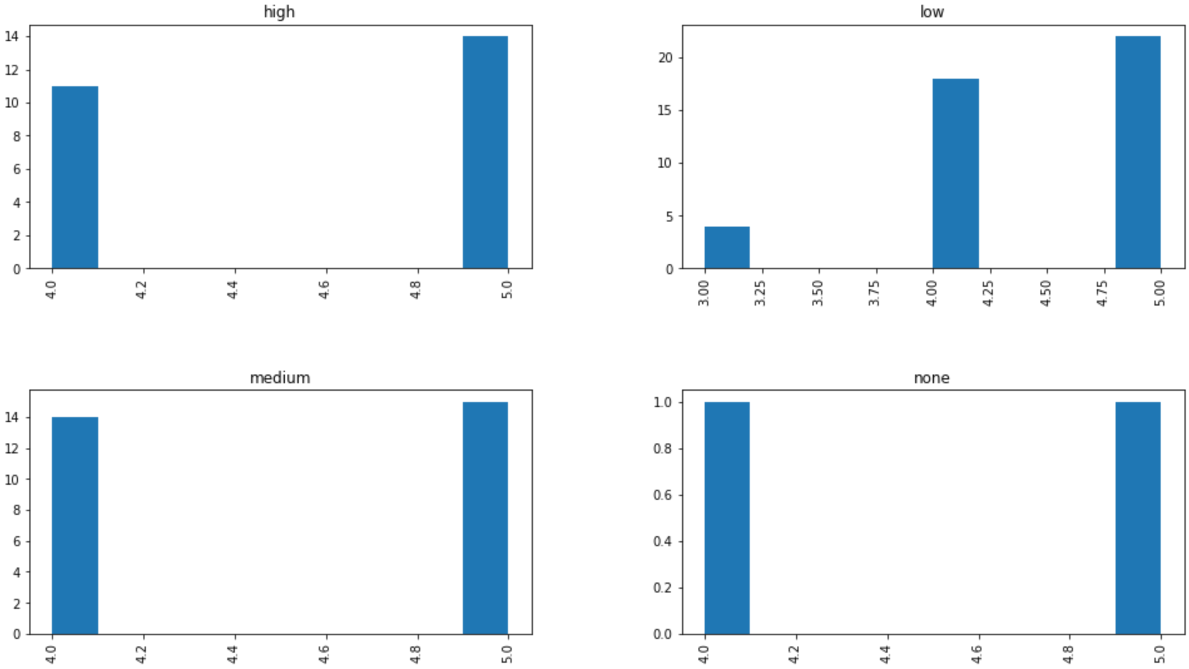
Out[148]:
```
array([[<AxesSubplot:title={'center':'Large'}>,
        <AxesSubplot:title={'center':'Medium'}>],
       [<AxesSubplot:title={'center':'Small'}>, <AxesSubplot:>]],
      dtype=object)
```



Ratings Vs Internet Experience

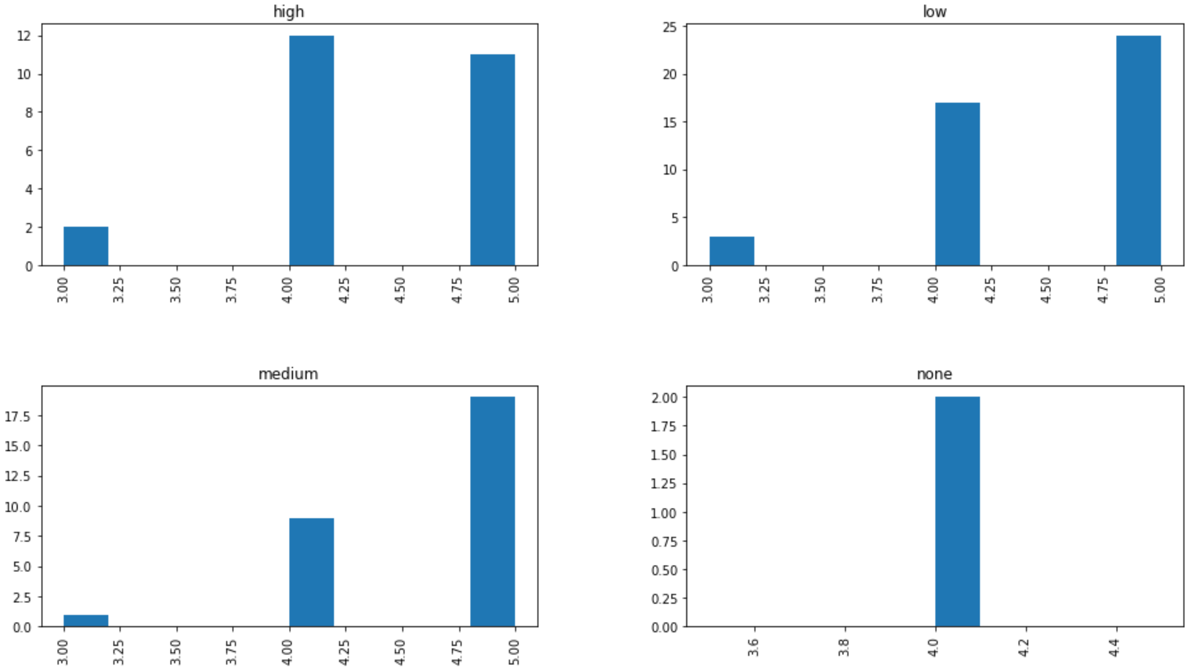
In [149]:
```
data01['item1ratevalue'].hist(by=data01['internetExperience'], figs
```

Out[149]:
```
array([[<AxesSubplot:title={'center':'high'}>,
        <AxesSubplot:title={'center':'low'}>],
       [<AxesSubplot:title={'center':'medium'}>,
        <AxesSubplot:title={'center':'none'}>]], dtype=object)
```



```
In [150]: data01['item2ratevalue'].hist(by=data01['internetExperience'], figs
```

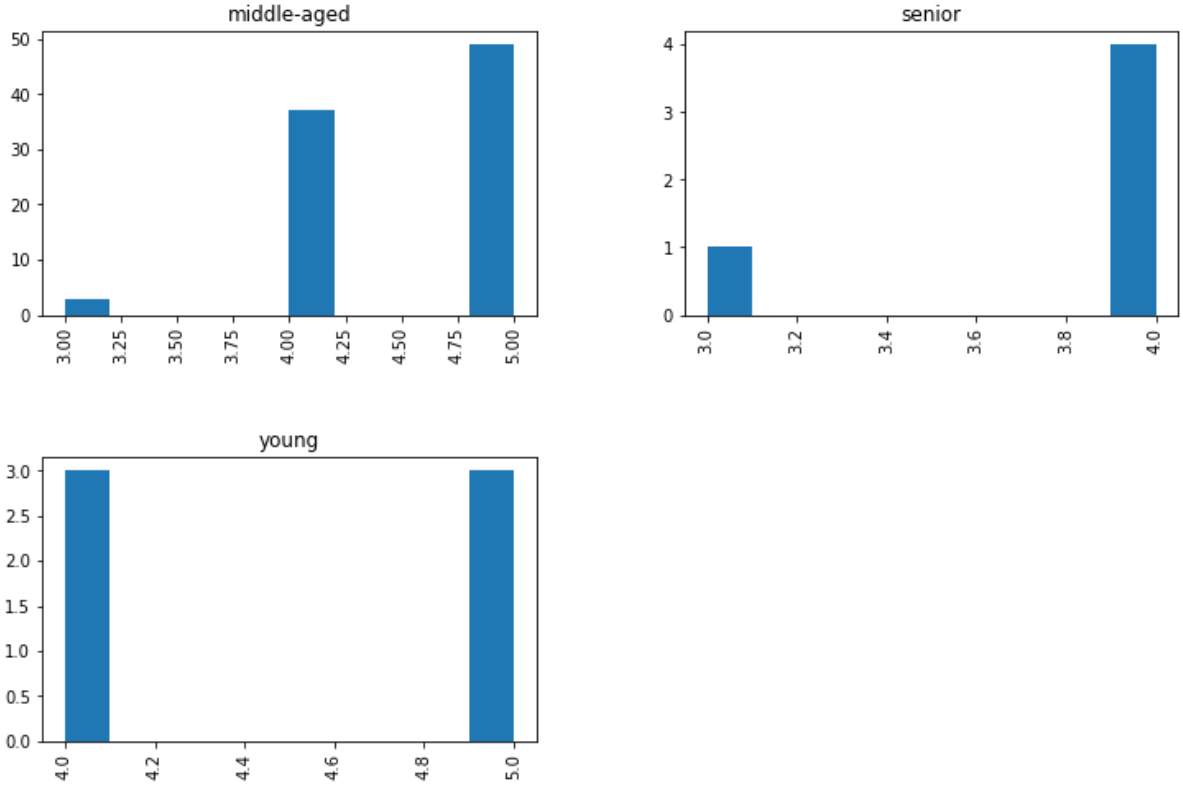
```
Out[150]: array([[<AxesSubplot:title={'center':'high'}>,
                  <AxesSubplot:title={'center':'low'}>],
                 [<AxesSubplot:title={'center':'medium'}>,
                  <AxesSubplot:title={'center':'none'}>]], dtype=object)
```



Ratings Vs Age Group

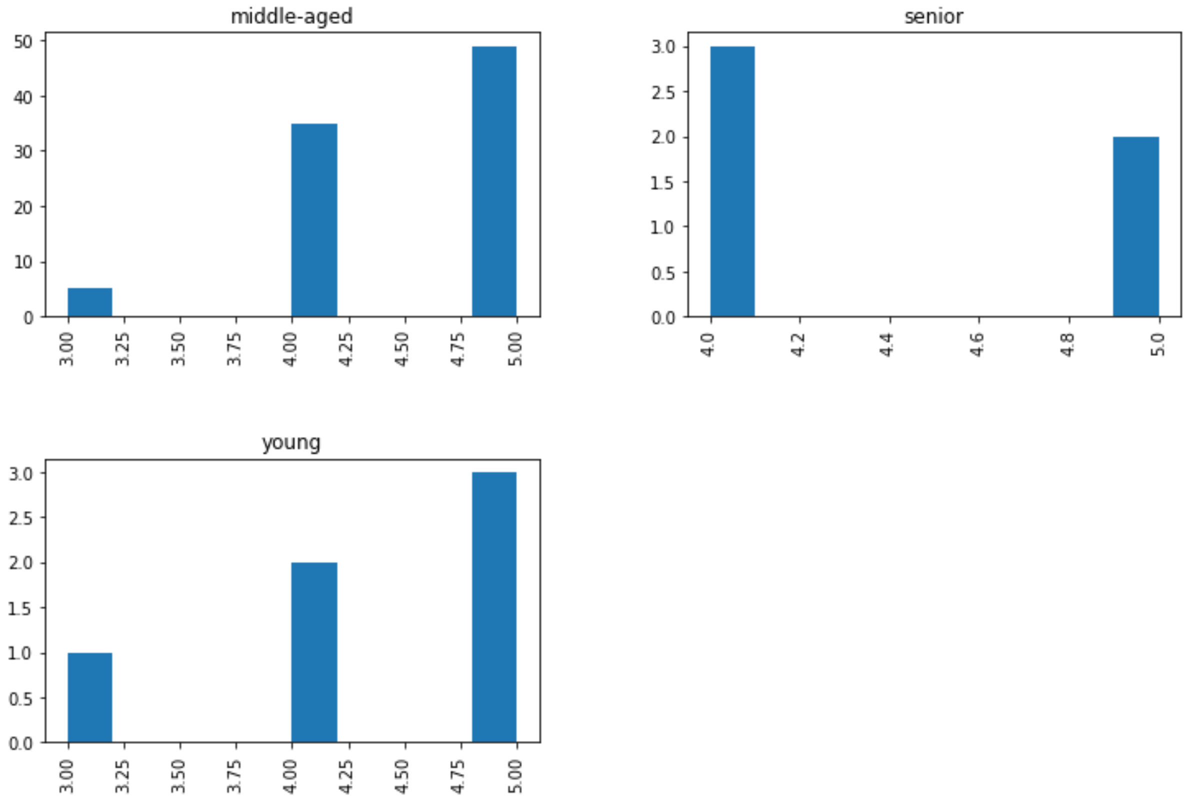
```
In [151]: data01['item1ratevalue'].hist(by=data01['ageGroup'], figsize=(12, 8
```

```
Out[151]: array([[<AxesSubplot:title={'center':'middle-aged'}>,
                  <AxesSubplot:title={'center':'senior'}>],
                 [<AxesSubplot:title={'center':'young'}>, <AxesSubplot:>]],
                dtype=object)
```



```
In [152]: data01['item2ratevalue'].hist(by=data01['ageGroup'], figsize=(12, 8
```

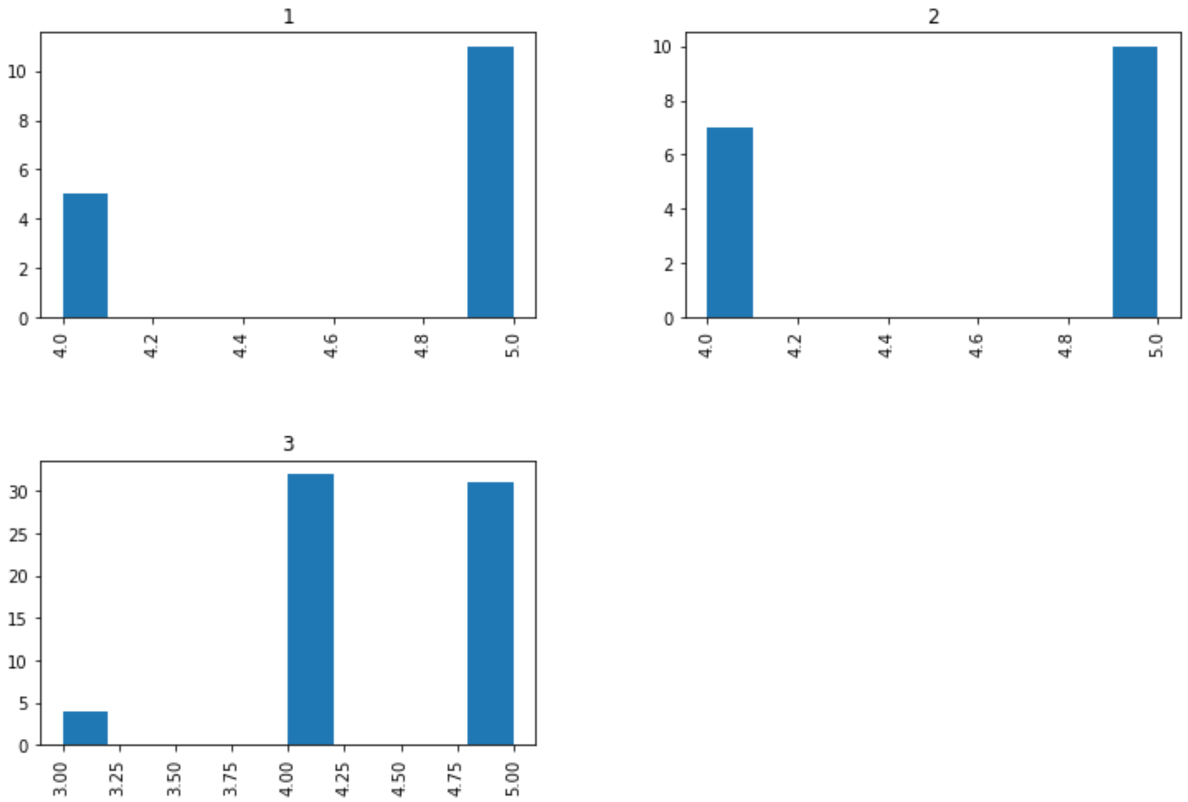
```
Out[152]: array([[<AxesSubplot:title={'center':'middle-aged'}>,
                  <AxesSubplot:title={'center':'senior'}>],
                 [<AxesSubplot:title={'center':'young'}>, <AxesSubplot:>]],
                dtype=object)
```



Education Vs Rating

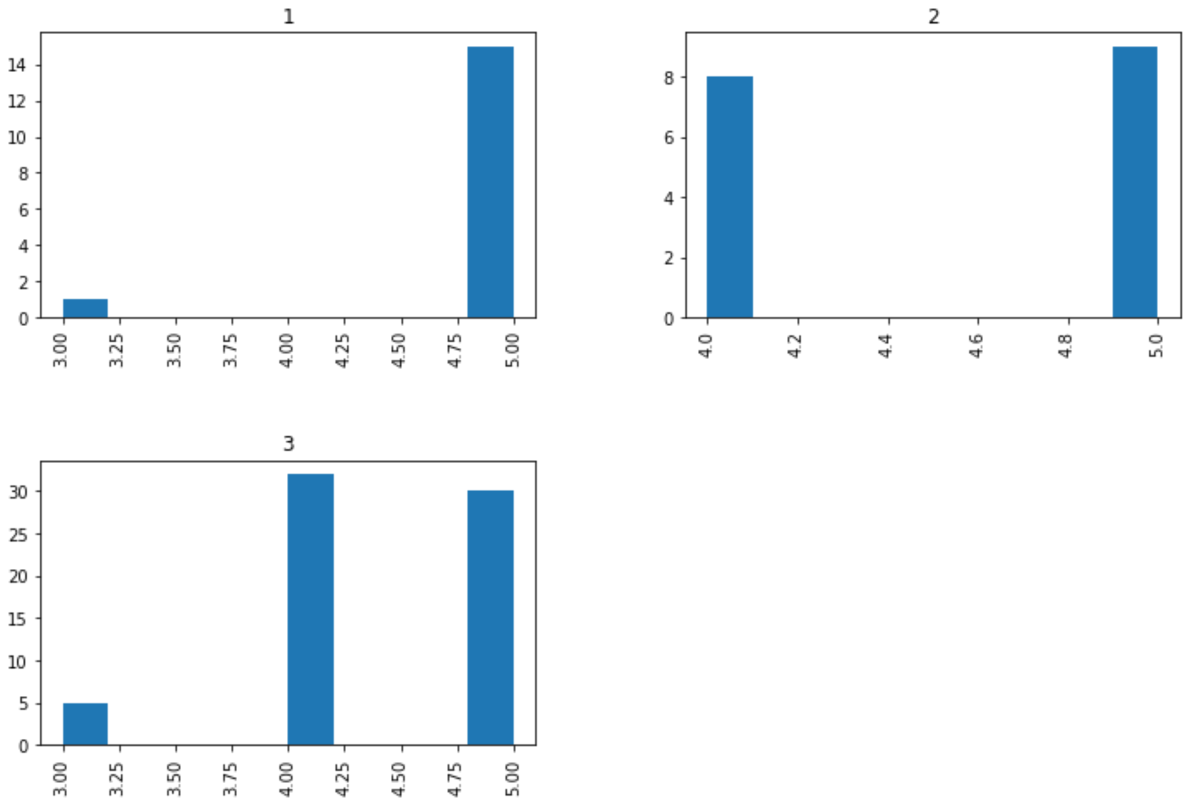
```
In [153]: data01['item1ratevalue'].hist(by=data01['education'], figsize=(12,
```

```
Out[153]: array([[<AxesSubplot:title={'center':'1'}>,
                  <AxesSubplot:title={'center':'2'}>],
                 [<AxesSubplot:title={'center':'3'}>, <AxesSubplot:>]], dtyp
          e=object)
```



```
In [154]: data01['item2ratevalue'].hist(by=data01['education'], figsize=(12,
```

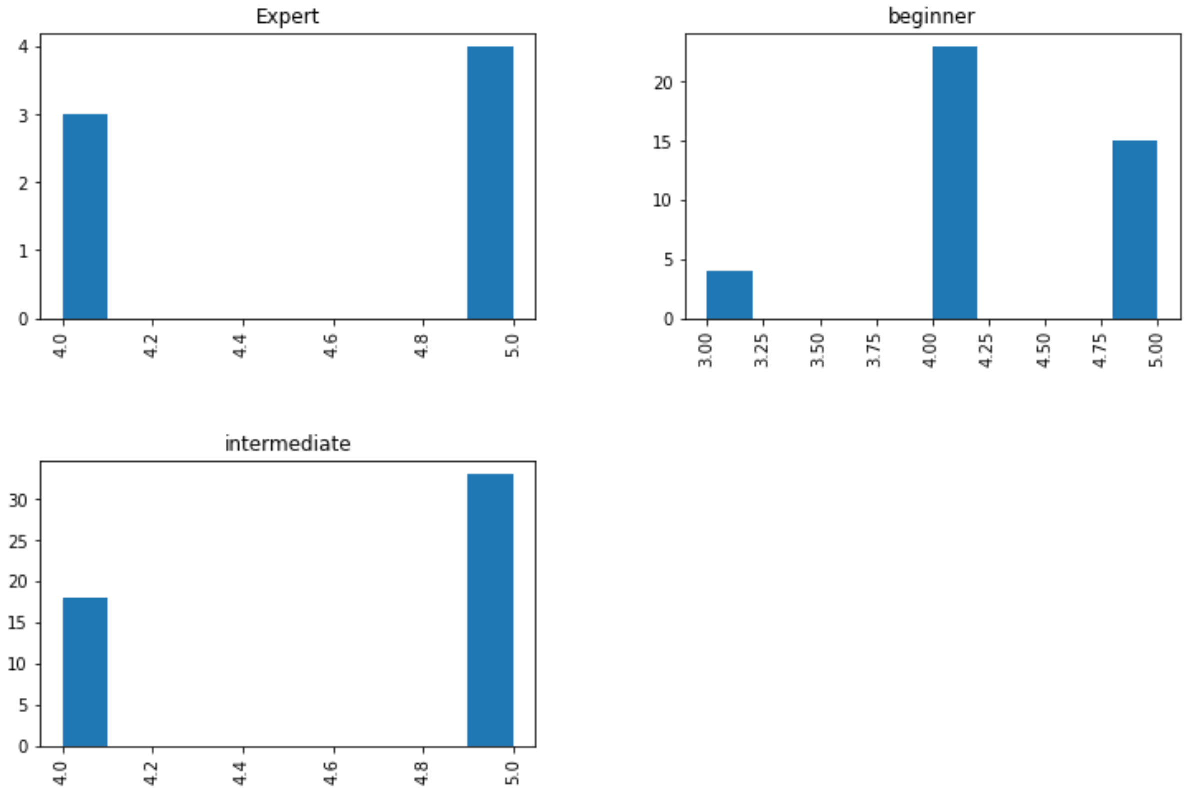
```
Out[154]: array([[<AxesSubplot:title={'center':'1'}>,
                  <AxesSubplot:title={'center':'2'}>],
                 [<AxesSubplot:title={'center':'3'}>, <AxesSubplot:>]], dtyp
          e=object)
```



Rating vs InternetSkill (IT_Skill_level)

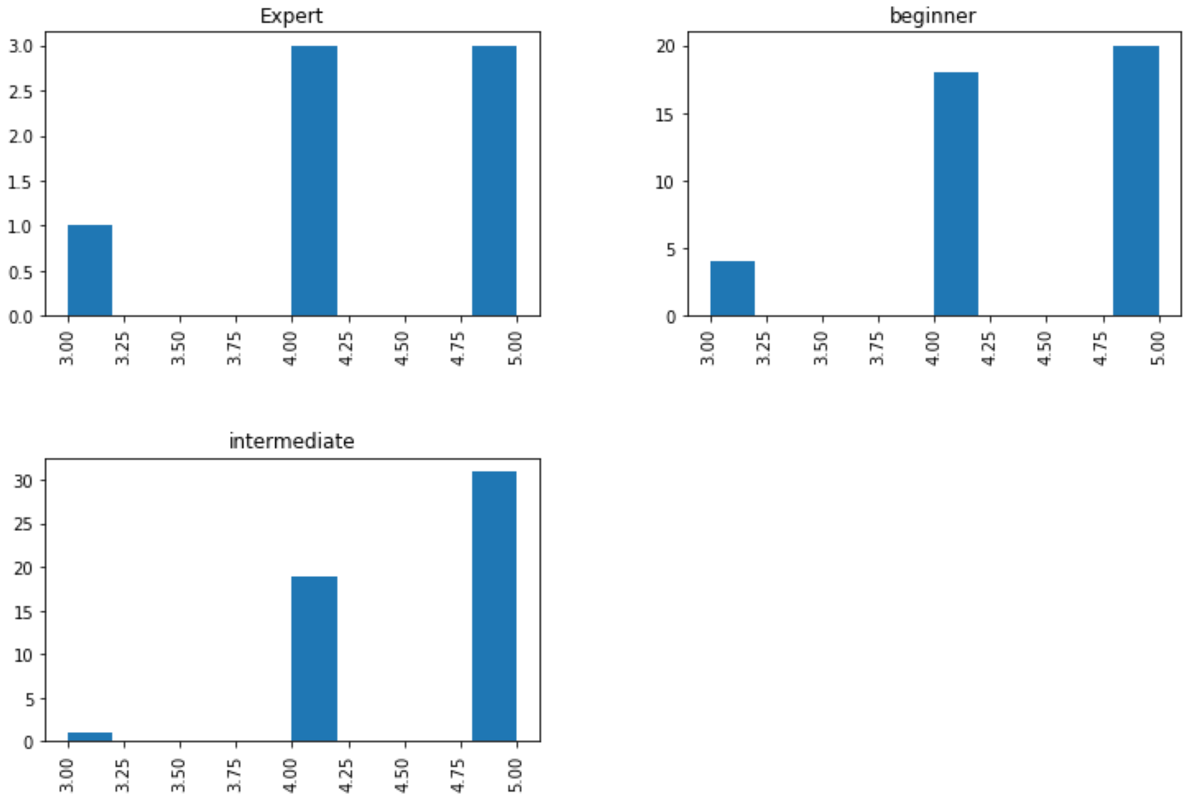
In [155]:
```
data01['item1ratevalue'].hist(by=data01['IT_Skill_level'], figsize=
```

Out[155]:
```
array([[<AxesSubplot:title={'center':'Expert'}>,
        <AxesSubplot:title={'center':'beginner'}>],
       [<AxesSubplot:title={'center':'intermediate'}>, <AxesSubplo
t:>]],
      dtype=object)
```



```
In [156]: data01['item2ratevalue'].hist(by=data01['IT_Skill_level'], figsize=
```

```
Out[156]: array([[<AxesSubplot:title={'center':'Expert'}>,
                  <AxesSubplot:title={'center':'beginner'}>],
                 [<AxesSubplot:title={'center':'intermediate'}>, <AxesSubplo
          t:>]],
                dtype=object)
```

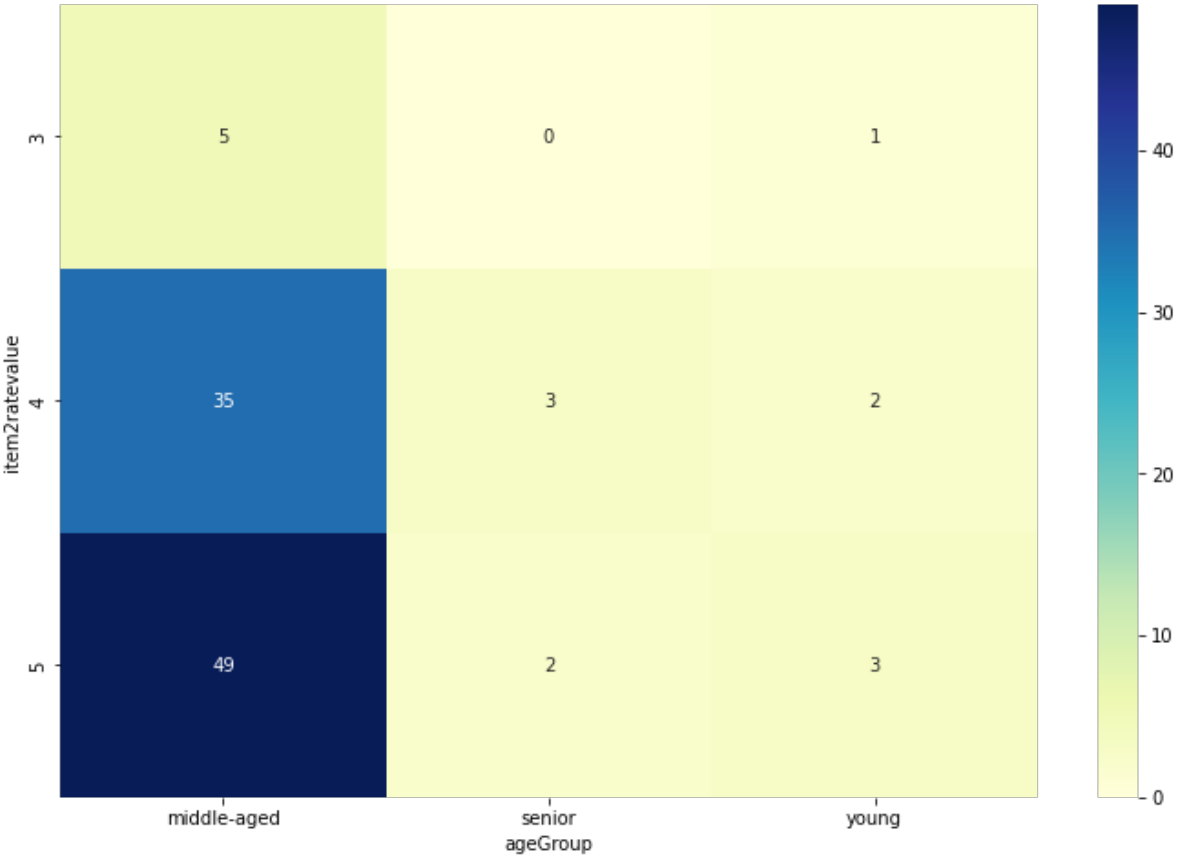


Independence (Chi-square and t-tests)

In [157]:

```
contigency_3= pd.crosstab(data01['item2ratevalue'], data01['ageGrou
print(contigency_3)
plt.figure(figsize=(12,8))
sns.heatmap(contigency_3, annot=True, cmap="YlGnBu")
plt.show()
c3, p3, dof3, expected3 = chi2_contingency(contigency_3)
print(c3)
print('This is the P-value')
print(round(p3,2))
```

```
ageGroup        middle-aged  senior  young
item2ratevalue
3                         5       0      1
4                        35       3      2
5                        49       2      3
```

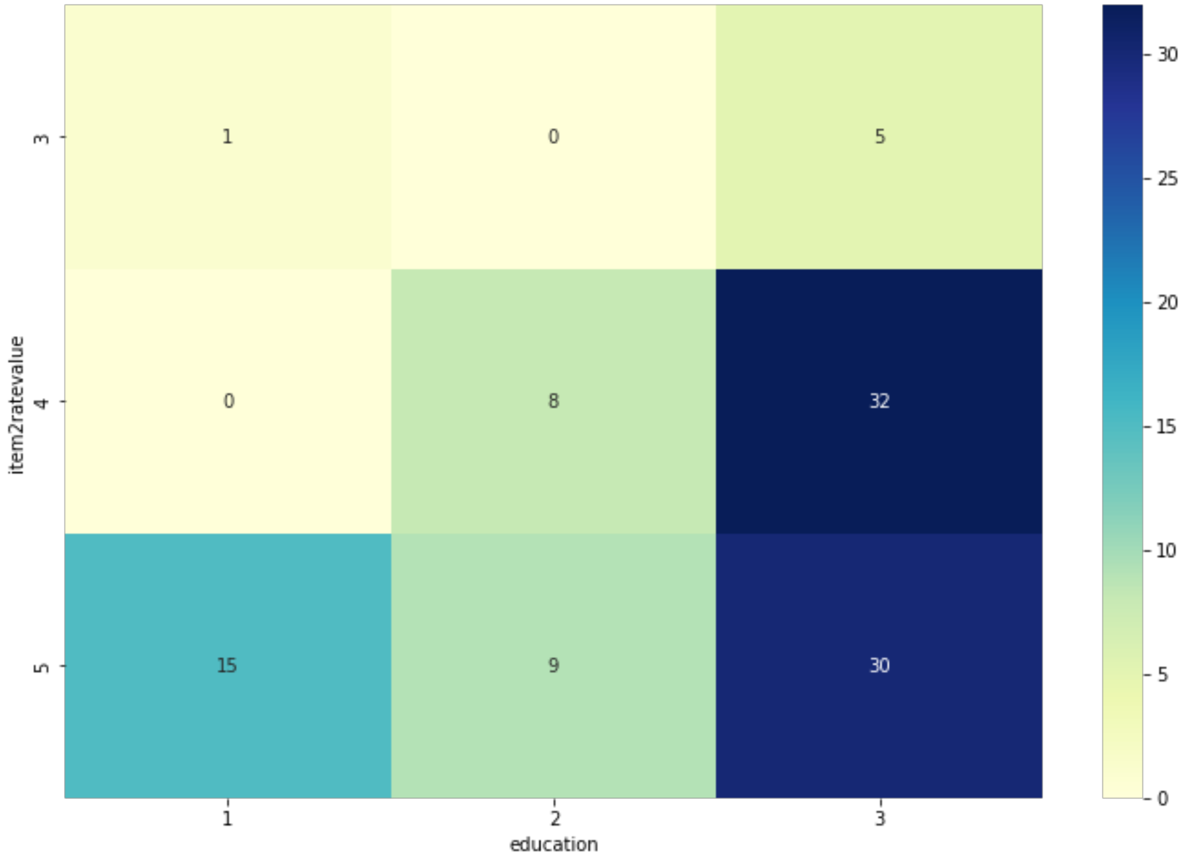


```
2.253849354972951
This is the P-value
0.69
```

In [158]:

```
contigency_4= pd.crosstab(data01['item2ratevalue'], data01['educati
print(contigency_4)
plt.figure(figsize=(12,8))
sns.heatmap(contigency_4, annot=True, cmap="YlGnBu")
plt.show()
c4, p4, dof4, expected4 = chi2_contingency(contigency_4)
print(c4)
print('This is the P-value')
print(round(p4,2))
```

```
education        1  2   3
item2ratevalue
3                1  0   5
4                0  8  32
5               15  9  30
```

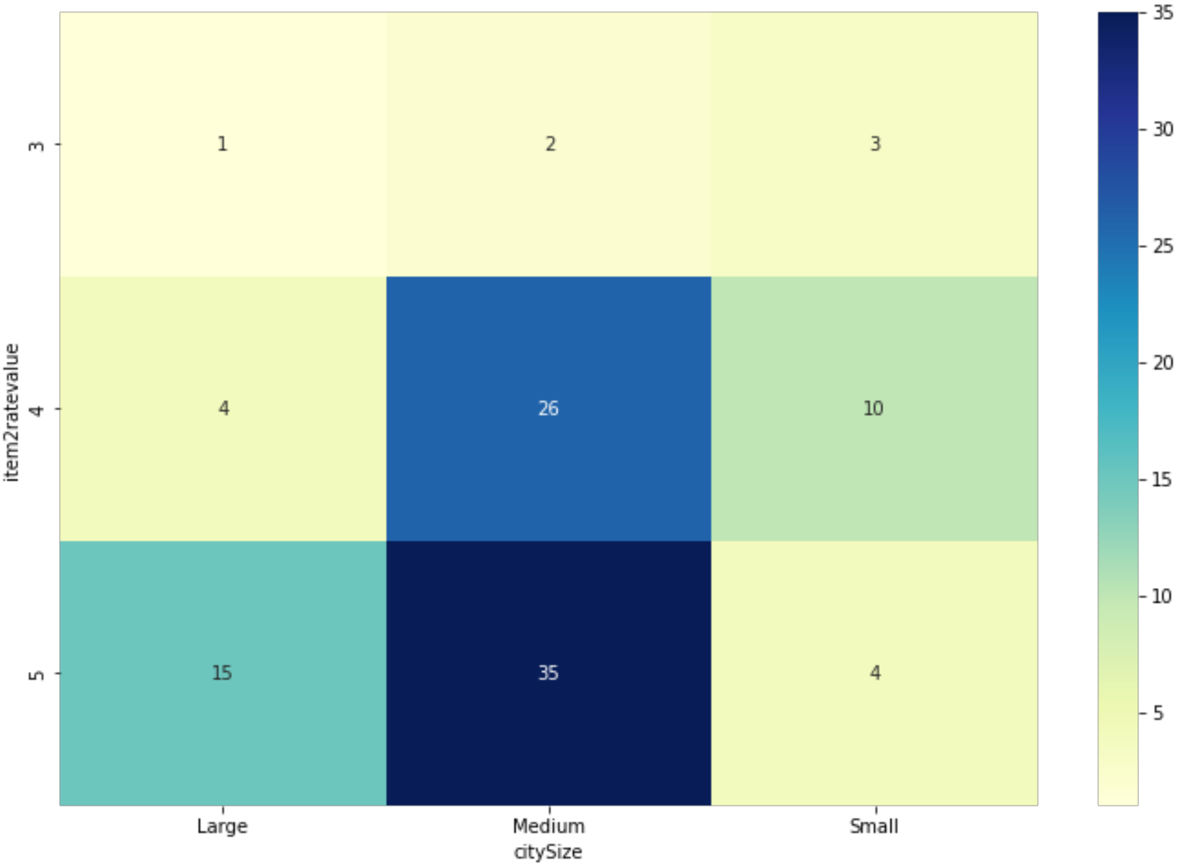


```
14.622110038045067
This is the P-value
0.01
```

In [159]:

```
contigency_5= pd.crosstab(data01['item2ratevalue'], data01['citySiz
print(contigency_5)
plt.figure(figsize=(12,8))
sns.heatmap(contigency_5, annot=True, cmap="YlGnBu")
plt.show()
c5, p5, dof5, expected5 = chi2_contingency(contigency_5)
print(c5)
print('This is the P-value')
print(round(p5,2))
```

```
citySize        Large  Medium  Small
item2ratevalue
3                   1       2      3
4                   4      26     10
5                  15      35      4
```

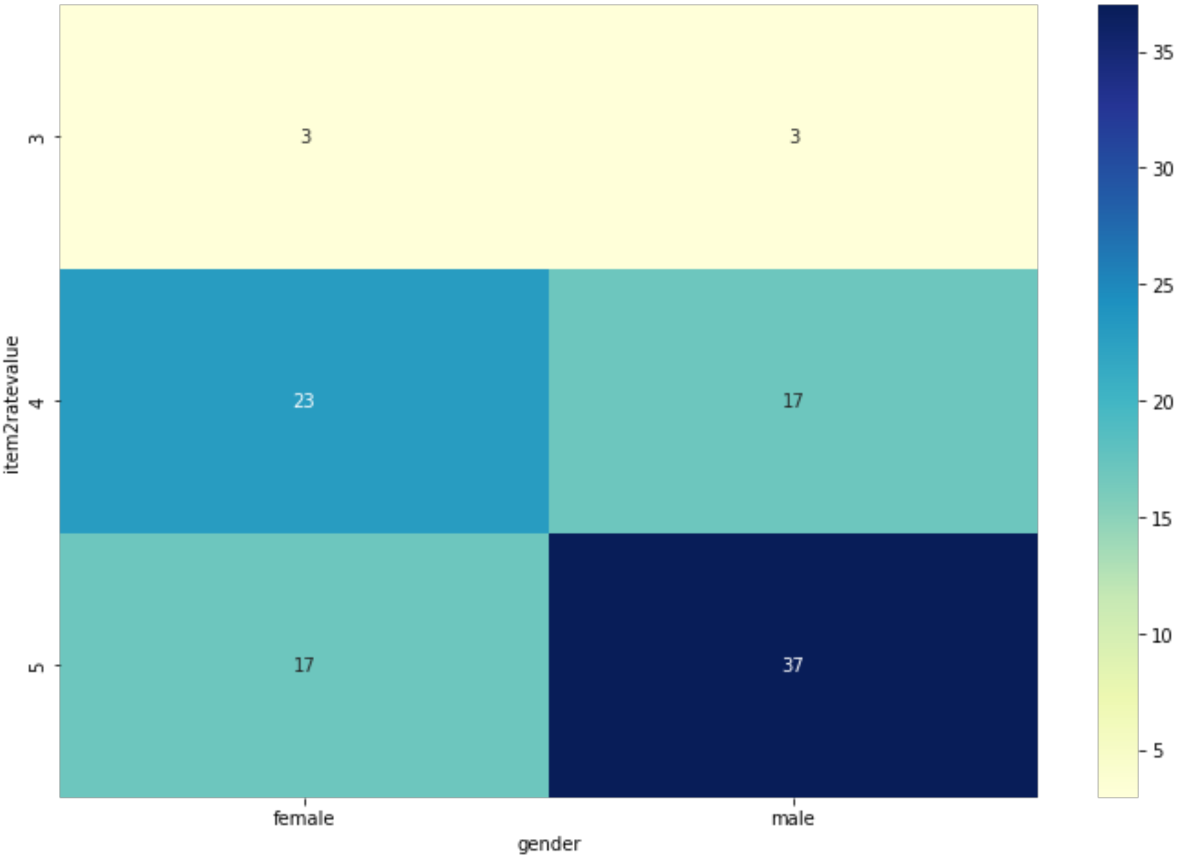


```
12.830826157623543
This is the P-value
0.01
```

In [160]:

```
contigency_6= pd.crosstab(data01['item2ratevalue'], data01['gender'
print(contigency_6)
plt.figure(figsize=(12,8))
sns.heatmap(contigency_6, annot=True, cmap="YlGnBu")
plt.show()
c6, p6, dof6, expected6 = chi2_contingency(contigency_6)
print(c6)
print('This is the P-value')
print(round(p6,2))
```

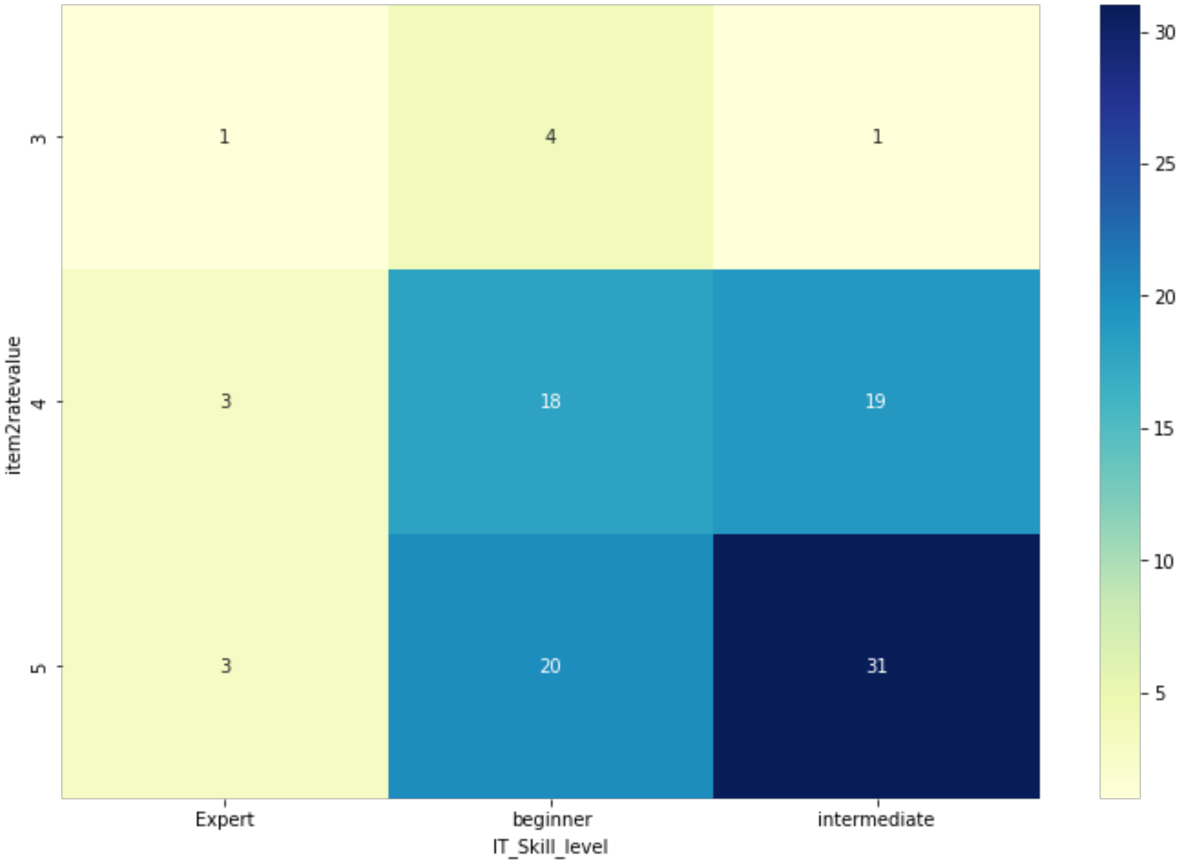
```
gender          female  male
item2ratevalue              
3                    3     3
4                   23    17
5                   17    37
```



```
6.4743037611254675
This is the P-value
0.04
```

In [161]:
```
contigency_7= pd.crosstab(data01['item2ratevalue'], data01['IT_Skil
print(contigency_7)
plt.figure(figsize=(12,8))
sns.heatmap(contigency_7, annot=True, cmap="YlGnBu")
plt.show()
c7, p7, dof7, expected7 = chi2_contingency(contigency_7)
print(c7)
print('This is the P-value')
print(round(p7,2))
stats.chi2_contingency(contigency_7)
```

```
IT_Skill_level  Expert  beginner  intermediate
item2ratevalue
3                    1         4             1
4                    3        18            19
5                    3        20            31
```



```
4.165369851644361
This is the P-value
0.38
```

Out[161]:
```
(4.165369851644361,
 0.38408784888965614,
 4,
 array([[ 0.42,  2.52,  3.06],
        [ 2.8 , 16.8 , 20.4 ],
        [ 3.78, 22.68, 27.54]]))
```

In [162]:

```
summary, results = rp.ttest(group1= data01['item1ratevalue'][data01
         group2= data01['item1ratevalue'][data01['gender'] == 'fema
print(results)
print(summary)
```

```
              Independent t-test  results
0  Difference (Male - Female) =    0.1893
1          Degrees of freedom =   98.0000
2                           t =    1.6381
3       Two side test p value =    0.1046
4      Difference < 0 p value =    0.9477
5      Difference > 0 p value =    0.0523
6                   Cohen's d =    0.3309
7                   Hedge's g =    0.3283
8              Glass's delta1 =    0.3538
9            Point-Biserial r =    0.1632
   Variable      N      Mean        SD        SE  95% Conf.  Inter
val
0      Male   57.0  4.561404  0.535108  0.070877   4.419420  4.703
387
1    Female   43.0  4.372093  0.618110  0.094261   4.181867  4.562
319
2  combined  100.0  4.480000  0.577000  0.057700   4.365511  4.594
489

/Users/manaswimondol/opt/anaconda3/lib/python3.9/site-packages/res
earchpy/ttest.py:38: FutureWarning: The series.append method is de
precated and will be removed from pandas in a future version. Use
pandas.concat instead.
  groups = group1.append(group2, ignore_index= True)
```

In [163]:

```
summary, results = rp.ttest(group1= data01['item2ratevalue'][data01
         group2= data01['item2ratevalue'][data01['gender'] == 'fema
print(results)
print(summary)
```

```
              Independent t-test  results
0  Difference (Male - Female) =    0.2709
1          Degrees of freedom =   98.0000
2                           t =    2.2391
3       Two side test p value =    0.0274
4      Difference < 0 p value =    0.9863
5      Difference > 0 p value =    0.0137
6                   Cohen's d =    0.4523
7                   Hedge's g =    0.4488
8              Glass's delta1 =    0.4565
9            Point-Biserial r =    0.2206
   Variable      N      Mean        SD        SE  95% Conf.  Inter
val
0      Male   57.0  4.596491  0.593406  0.078599   4.439040  4.753
943
1    Female   43.0  4.325581  0.606352  0.092468   4.138974  4.512
189
2  combined  100.0  4.480000  0.611010  0.061101   4.358762  4.601
238

/Users/manaswimondol/opt/anaconda3/lib/python3.9/site-packages/res
earchpy/ttest.py:38: FutureWarning: The series.append method is de
precated and will be removed from pandas in a future version. Use
pandas.concat instead.
  groups = group1.append(group2, ignore_index= True)
```

Internetskill, internetfrequency and internetexperience are correlated to each other so comparing skill with rating is enough to show dependence. InternetSkill has be categorised into 3 categories so that it makes it easier to run t-test and Chi-square tests

Item 1 Category 1

Dependent - Yes if p value < 0.1 (90% confidence level)
Item rating and age group - Yes
Item rating and education - No
Item rating and ciy size - Yes
Item rating and gender - No
Item rating and IT_Skill_level - Yes

Item 2 Category 2

Dependent - Yes if p value < 0.1 (90% confidence level)
Item rating and age group - No
Item rating and education - Yes
Item rating and ciy size - Yes
Item rating and gender - Yes
Item rating and IT_Skill_level - No