In [92]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```
In [ ]:
```
# 1 importing data
```
In [4]:
```
data = pd.read_csv("C:/Users/USER/OneDrive/Jonah Mbugua/Mbugua/Hex Softwares Internship 2024/Data sets/MBA admission.csv")
```
In [ ]:
```
# 2 understanding the data
```
In [5]:
```
data.head()
```
Out[5]:

	application_id	gender	international	gpa	major	race	gmat	work_exp	work
0	1	Female	False	3.30	Business	Asian	620.0	3.0	Financ Servic
1	2	Male	False	3.28	Humanities	Black	680.0	5.0	Invest Manag
2	3	Female	True	3.30	Business	NaN	710.0	5.0	Techn
3	4	Male	False	3.47	STEM	Black	690.0	6.0	Techn
4	5	Male	False	3.35	STEM	Hispanic	590.0	5.0	Consu

In [6]:
```
data.tail()
```
Out[6]:

	application_id	gender	international	gpa	major	race	gmat	work_exp	wor
6189	6190	Male	False	3.49	Business	White	640.0	5.0	Othe
6190	6191	Male	False	3.18	STEM	Black	670.0	4.0	Cons
6191	6192	Female	True	3.22	Business	NaN	680.0	5.0	Heal
6192	6193	Male	True	3.36	Business	NaN	590.0	5.0	Othe
6193	6194	Male	False	3.23	STEM	Hispanic	650.0	4.0	Cons

In [ ]:
```
# 3 descriptive analysis
```
In [15]:
```
data.describe()
```
Out[15]:

	application_id	gpa	gmat	work_exp
count	6194.000000	6194.000000	6194.000000	6194.000000

mean	3097.500000	3.250714	651.092993	5.016952
std	1788.198115	0.151541	49.294883	1.032432
min	1.000000	2.650000	570.000000	1.000000
25%	1549.250000	3.150000	610.000000	4.000000
50%	3097.500000	3.250000	650.000000	5.000000
75%	4645.750000	3.350000	680.000000	6.000000
max	6194.000000	3.770000	780.000000	9.000000

In [7]:
```
data.shape
```
Out[7]:
```
(6194, 10)
```
In [10]:
```
data.columns
```
Out[10]:
```
Index(['application_id', 'gender', 'international', 'gpa', 'major', 'race',
       'gmat', 'work_exp', 'work_industry', 'admission'],
      dtype='object')
```
In [ ]:
```
# 4 uniqueness of the dataset
```
In [25]:
```
 data.nunique()
```
Out[25]:
```
application_id    6194
gender               2
international        2
gpa                101
major                3
race                 5
gmat                22
work_exp             9
work_industry       14
admission            2
dtype: int64
```
In [26]:
```
 data['race'].unique ()
```
Out[26]:
```
array(['Asian', 'Black', nan, 'Hispanic', 'White', 'Other'], dtype=object)
```
In [27]:
```
data['gpa'].unique ()
```
Out[27]:
```
array([3.3 , 3.28, 3.47, 3.35, 3.18, 2.93, 3.02, 3.24, 3.27, 3.05, 2.85,
       3.39, 3.03, 3.32, 3.23, 3.13, 3.09, 3.46, 3.64, 3.42, 3.4 , 3.26,
       2.99, 3.08, 3.65, 3.04, 3.19, 3.33, 3.53, 3.5 , 3.22, 3.16, 3.45,
       3.12, 3.41, 3.38, 3.43, 2.96, 3.44, 3.01, 3.  , 3.36, 3.31, 3.07,
       3.49, 3.34, 2.89, 3.2 , 3.17, 3.1 , 3.52, 3.15, 3.21, 3.48, 3.14,
       2.97, 3.11, 3.29, 3.25, 3.51, 3.06, 2.95, 3.37, 3.55, 3.54, 3.6 ,
       3.61, 3.71, 3.77, 3.58, 2.98, 3.56, 3.69, 2.79, 2.87, 2.88, 3.63,
       2.9 , 3.74, 2.91, 2.92, 2.78, 3.57, 3.66, 2.81, 3.59, 2.82, 3.62,
```

```
       2.73, 3.68, 2.84, 2.83, 2.86, 3.67, 2.94, 2.72, 2.8 , 3.76, 3.7 ,
       3.73, 2.65])
```

In [29]:

```
data['work_industry'].unique ()
```

Out[29]:

```
array(['Financial Services', 'Investment Management', 'Technology',
       'Consulting', 'Nonprofit/Gov', 'PE/VC', 'Health Care',
       'Investment Banking', 'Other', 'Retail', 'Energy', 'CPG',
       'Real Estate', 'Media/Entertainment'], dtype=object)
```

In [97]:

```
print(df)
```

```
      application_id  international   gpa   gmat  work_exp
0                  1          False  3.30  620.0       3.0
1                  2          False  3.28  680.0       5.0
2                  3           True  3.30  710.0       5.0
3                  4          False  3.47  690.0       6.0
4                  5          False  3.35  590.0       5.0
...              ...            ...   ...    ...       ...
6189            6190          False  3.49  640.0       5.0
6190            6191          False  3.18  670.0       4.0
6191            6192           True  3.22  680.0       5.0
6192            6193           True  3.36  590.0       5.0
6193            6194          False  3.23  650.0       4.0

[6194 rows x 5 columns]
```

In [ ]:

 # cleaning the data

In [23]:

```
data.isnull().sum()
```

Out[23]:

```
application_id       0
gender               0
international        0
gpa                  0
major                0
race              1842
gmat                 0
work_exp             0
work_industry        0
admission         5194
dtype: int64
```

In [131]:

```
df= data.drop(['major','gender','race','work_industry','admission'],axis=1)
```

In [132]:

```
df.head()
```

Out[132]:

	application_id	international	gpa	gmat	work_exp
0	1	False	3.30	620.0	3.0
1	2	False	3.28	680.0	5.0
2	3	True	3.30	710.0	5.0
3	4	False	3.47	690.0	6.0

4	5	False	3.35	590.0	5.0

In [133]:
```
print(df)
```

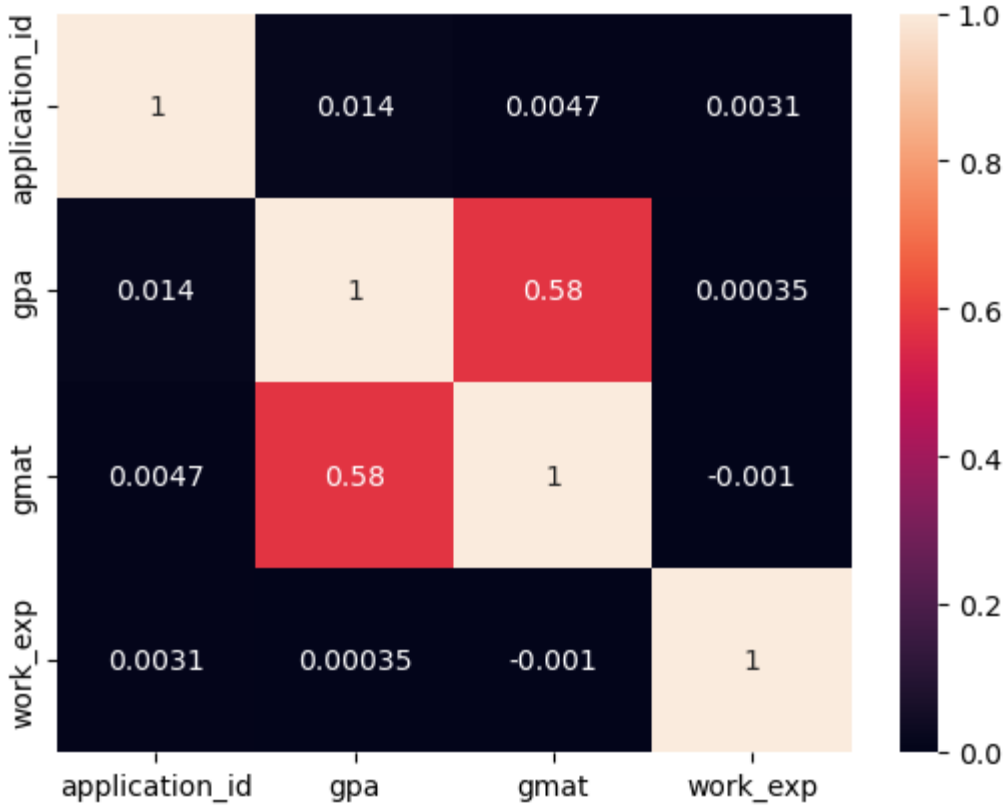
```
      application_id  international   gpa   gmat  work_exp
0                  1          False  3.30  620.0       3.0
1                  2          False  3.28  680.0       5.0
2                  3           True  3.30  710.0       5.0
3                  4          False  3.47  690.0       6.0
4                  5          False  3.35  590.0       5.0
...              ...            ...   ...   ...       ...
6189            6190          False  3.49  640.0       5.0
6190            6191          False  3.18  670.0       4.0
6191            6192           True  3.22  680.0       5.0
6192            6193           True  3.36  590.0       5.0
6193            6194          False  3.23  650.0       4.0

[6194 rows x 5 columns]
```

In [ ]:
```
#relationship analysis
```

In [134]:
```
correlation = df.corr()
```

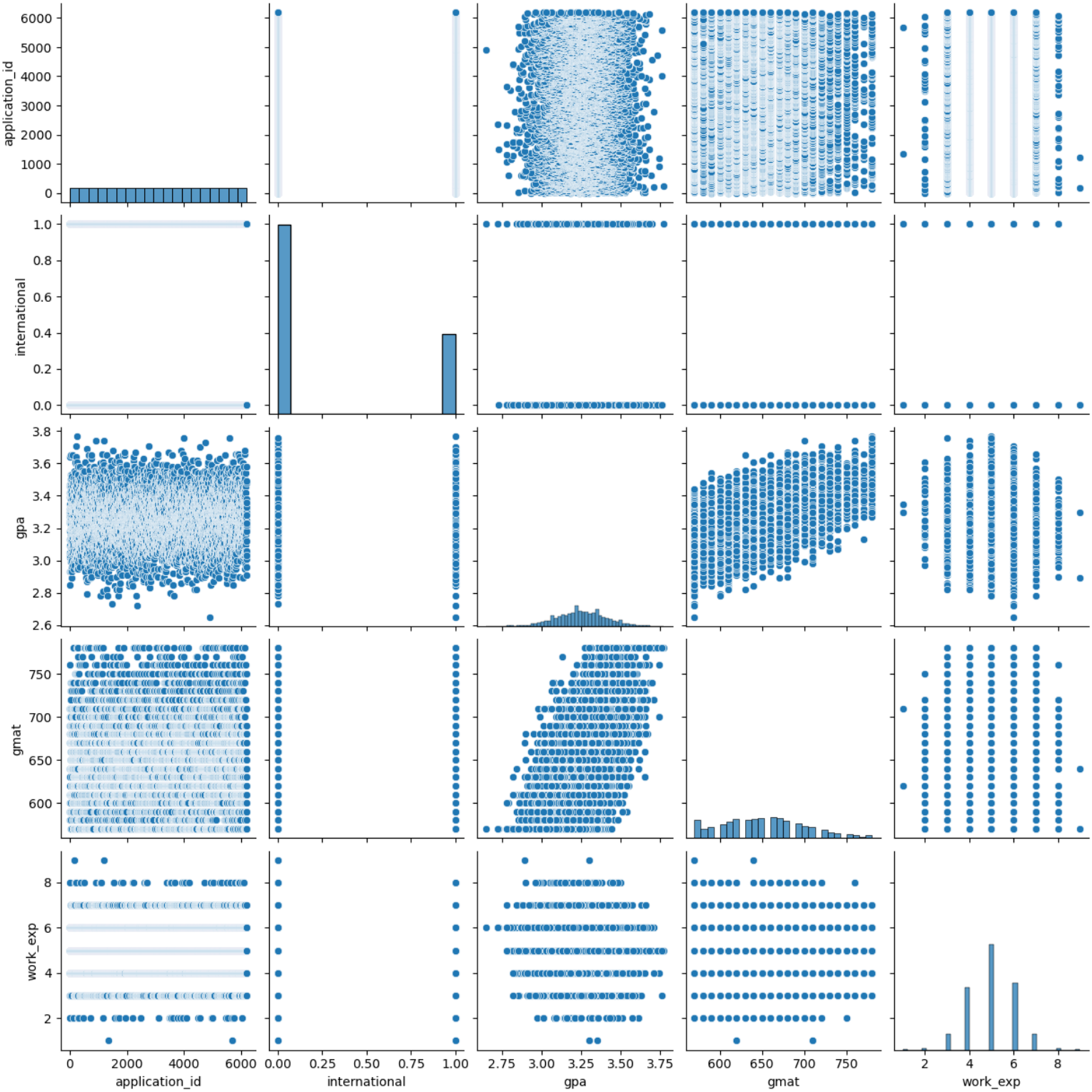
In [108]:
```
sns.heatmap(correlation, xticklabels=correlation.columns, yticklabels=correlation.columns, annot=True)
```

Out[108]:
<Axes: >



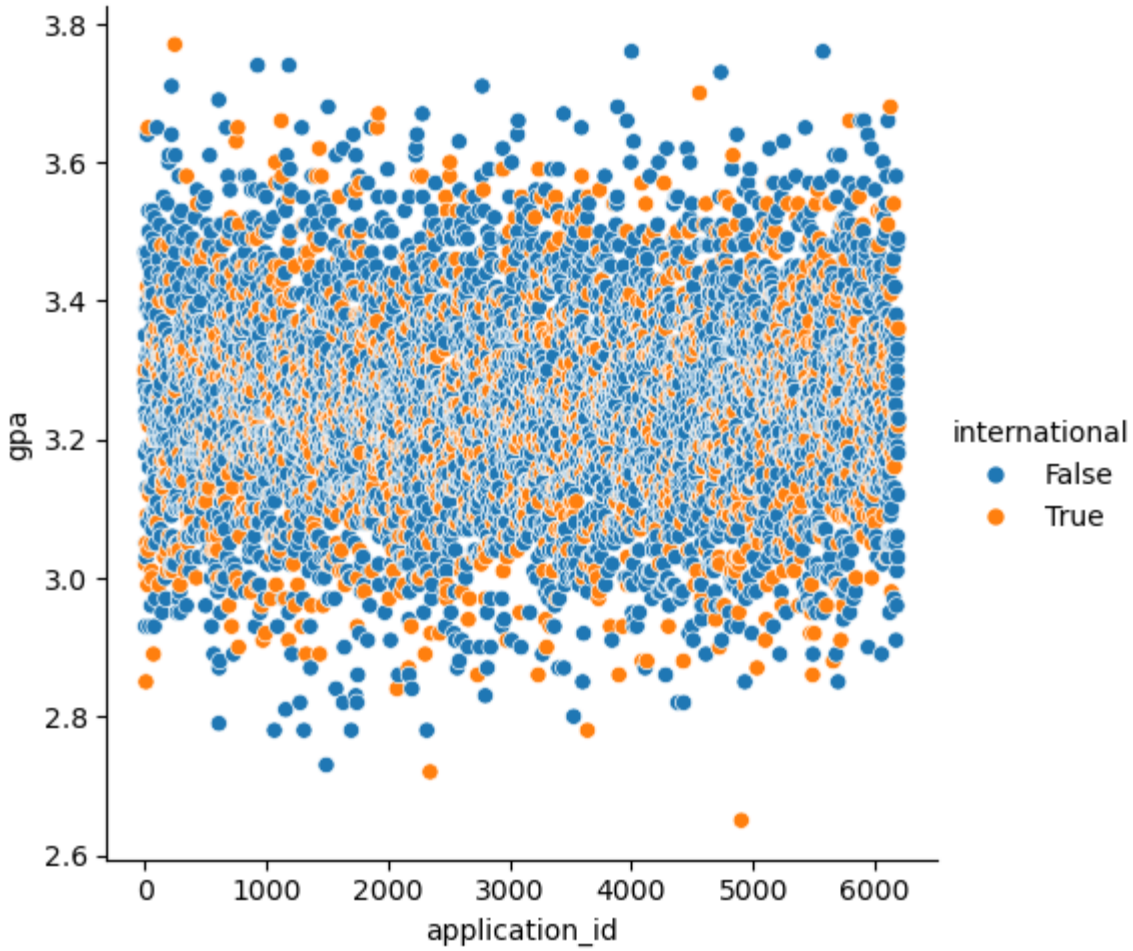
In [135]:
```
sns.pairplot(df)
```

Out[135]:
<seaborn.axisgrid.PairGrid at 0x1216e55b2f0>

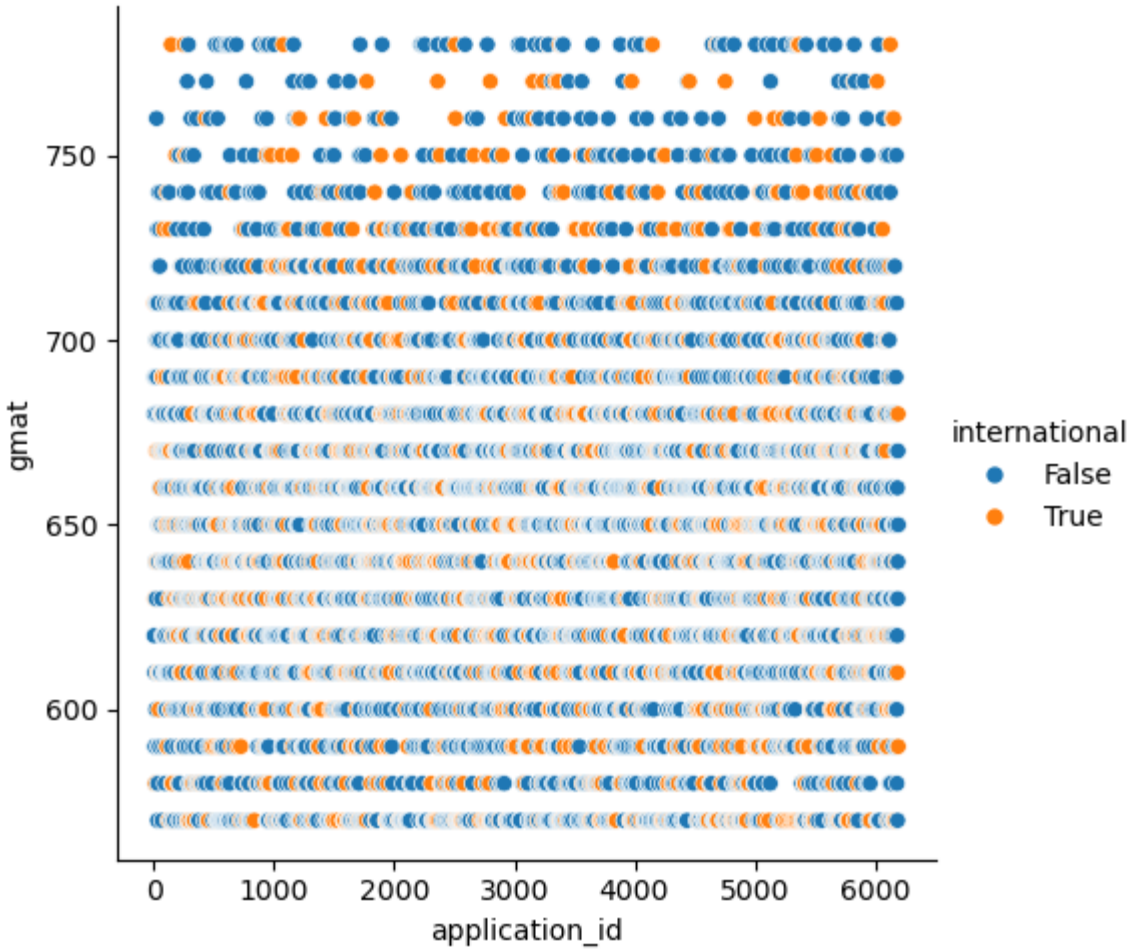


In [138]:
```
sns.relplot(x='application_id', y='gpa', hue='international', data=df)
```

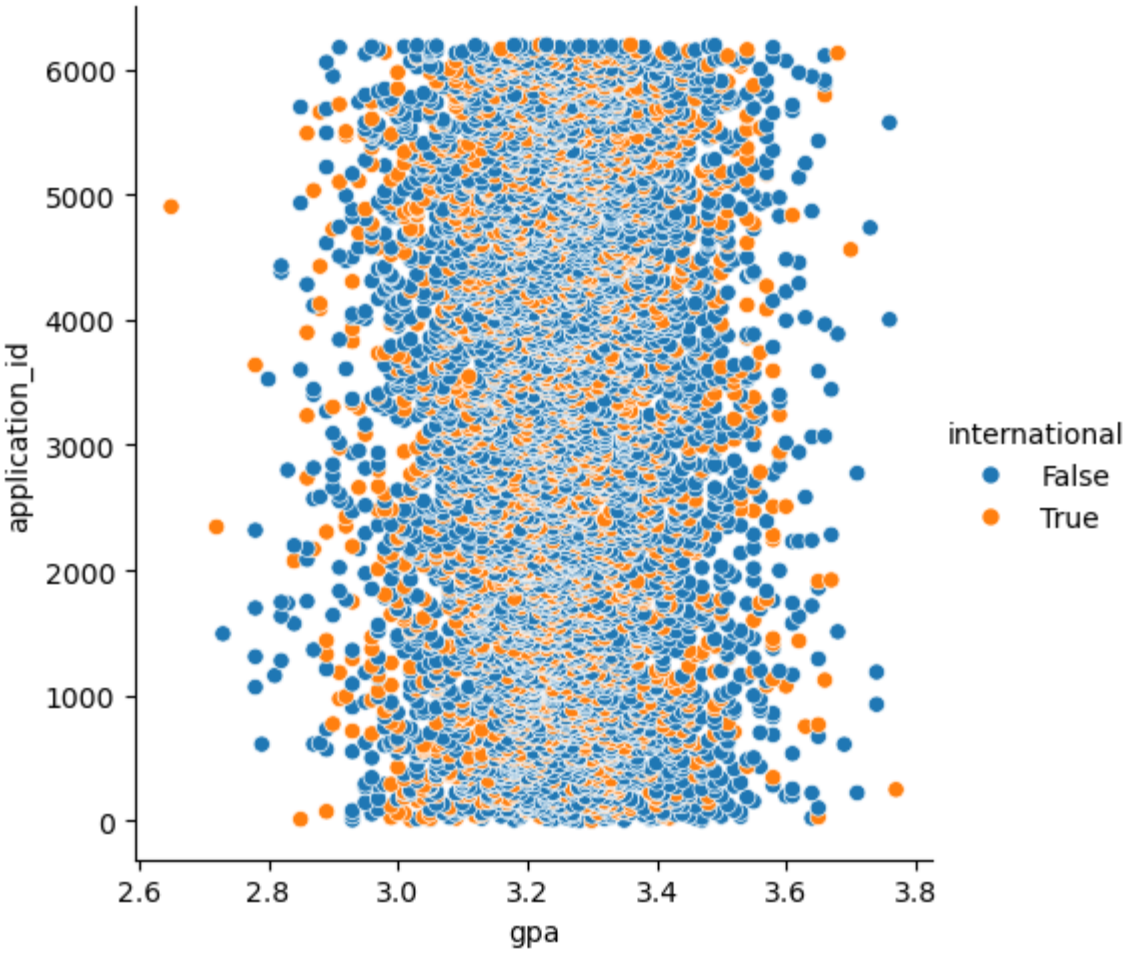
Out[138]:
<seaborn.axisgrid.FacetGrid at 0x121713b5460>



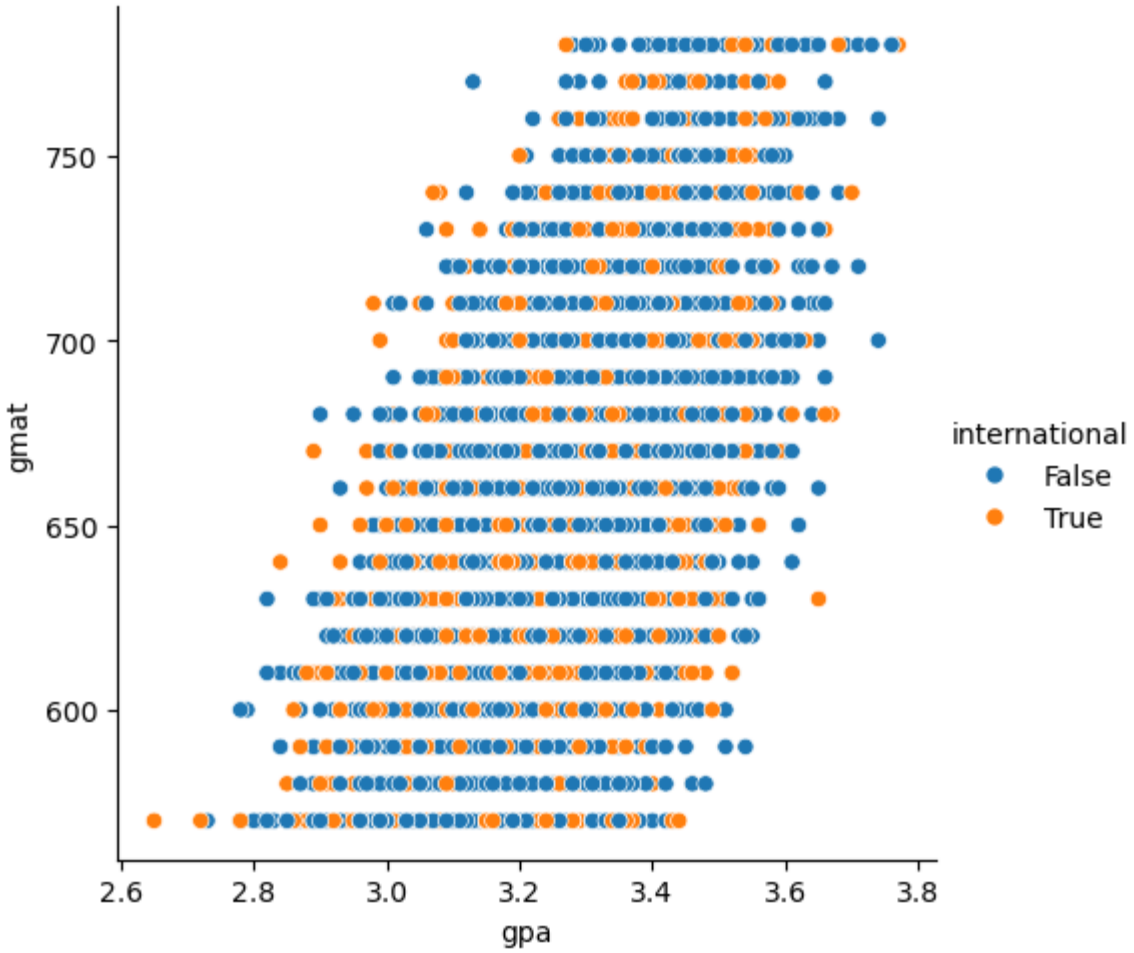
In [139]:
```
sns.relplot(x='application_id', y='gmat', hue='international', data=df)
```
Out[139]:
<seaborn.axisgrid.FacetGrid at 0x121715fb8c0>



In [141]:
```
sns.relplot(x='gpa',y='application_id',  hue='international', data=df)
```
Out[141]:
<seaborn.axisgrid.FacetGrid at 0x12171c4f770>



In [142]:
```
sns.relplot(x='gpa',y='gmat',  hue='international', data=df)
```
Out[142]:
<seaborn.axisgrid.FacetGrid at 0x12171f55460>



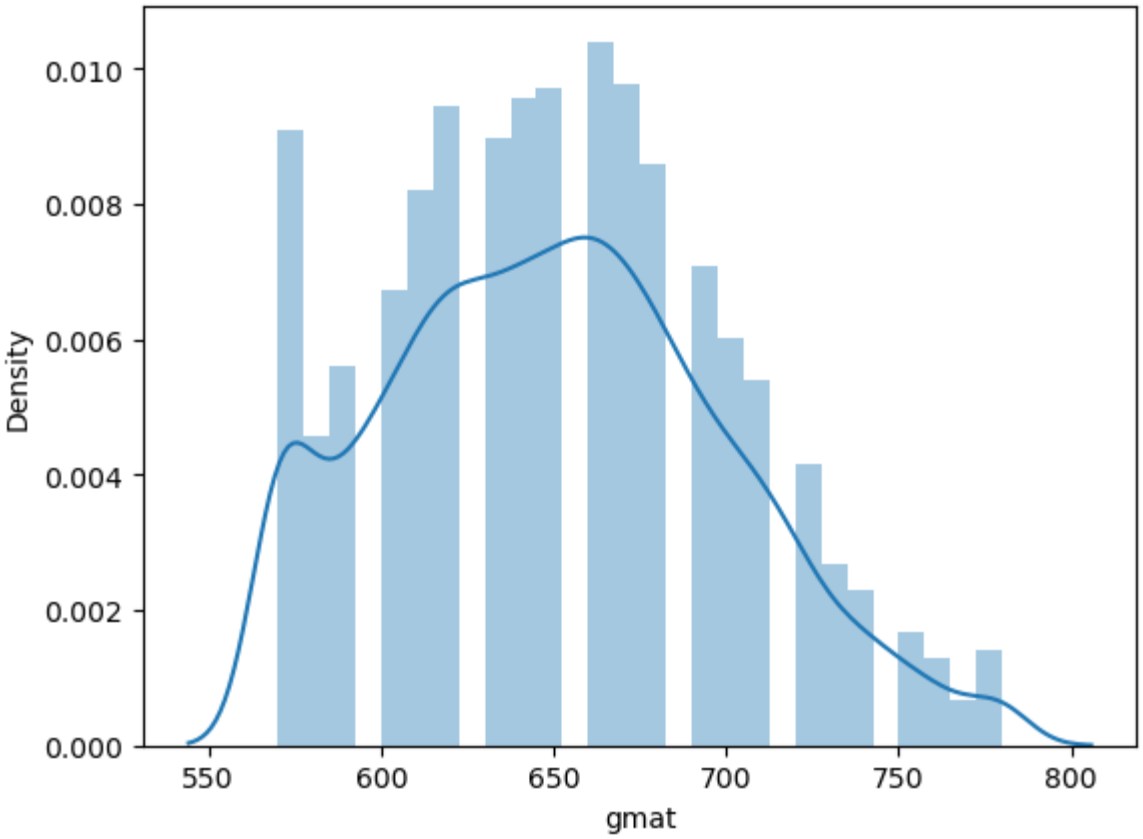
In [153]:
```
sns.distplot(df['gmat'])
```
C:\Users\USER\AppData\Local\Temp\ipykernel_16232\180379012.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

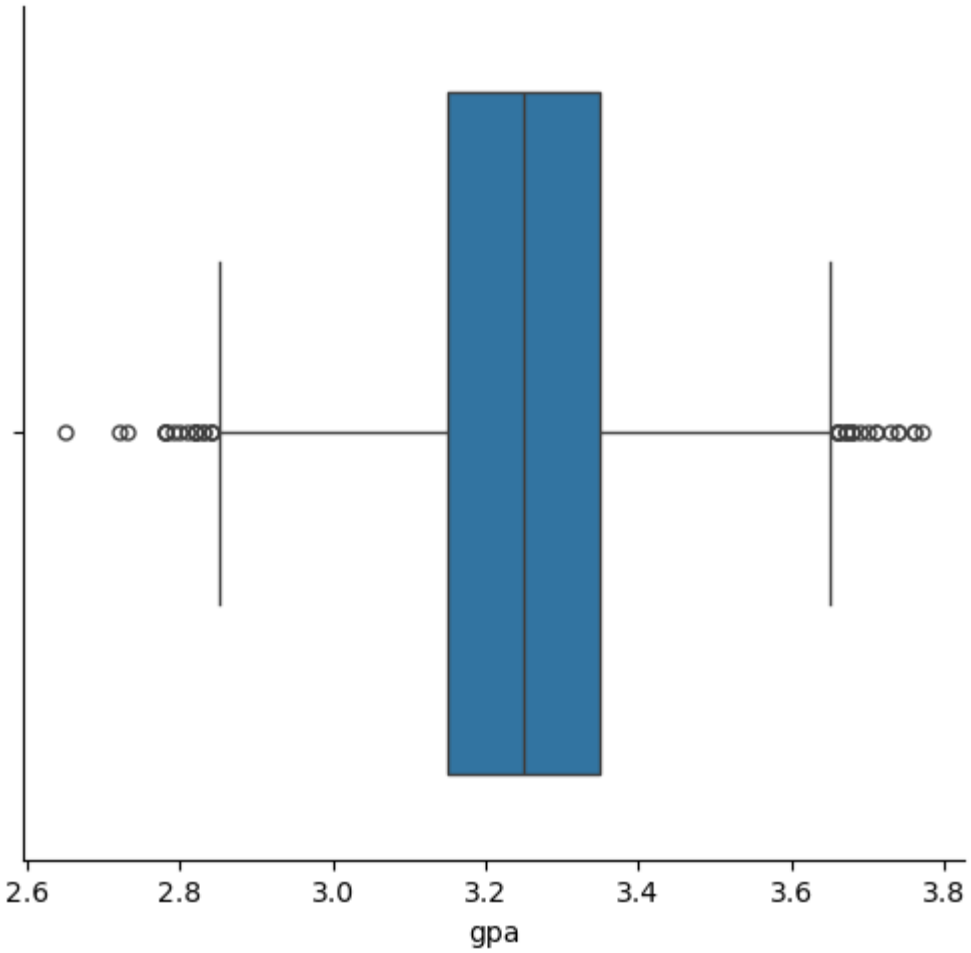
Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

```
  sns.distplot(df['gmat'])
```
Out[153]:
<Axes: xlabel='gmat', ylabel='Density'>



In [154]:
```
sns.catplot(x='gpa',kind='box',data=df)
```
Out[154]:
<seaborn.axisgrid.FacetGrid at 0x12174c70050>



In [ ]: