

In [1]:

```
%matplotlib inline

from pathlib import Path

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

from keras.models import Sequential
from keras.layers import Dense
from keras.utils import np_utils
from sklearn.preprocessing import LabelEncoder
from keras.utils.np_utils import to_categorical
from sklearn.utils import shuffle
```

Using TensorFlow backend.

In [2]:

```
DATA_DIR = Path('.', 'data', 'final', 'public')
```

In [3]:

```
train_values = pd.read_csv(DATA_DIR / 'train_values.csv', index_col='building_id')
train_labels = pd.read_csv(DATA_DIR / 'train_labels.csv', index_col='building_id')
```

In [4]:

```
#print(train_values.dtypes)
s = (train_values.dtypes == 'object')
object_cols = list(s[s].index)
s = (train_values.dtypes != 'object')
number_cols = list(s[s].index)
```

In [5]:

```
test_values = pd.read_csv(DATA_DIR / 'test_values.csv', index_col='building_id')
```

In [6]:

```
sns_data=train_values[number_cols].join(train_labels)
```

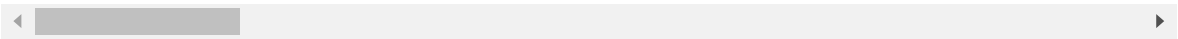
In [7]:

```
sns_data.describe()
```

Out[7]:

	geo_level_1_id	geo_level_2_id	geo_level_3_id	count_floors_pre_eq	age	ai
count	260601.000000	260601.000000	260601.000000	260601.000000	260601.000000	
mean	13.900353	701.074685	6257.876148	2.129723	26.535029	
std	8.033617	412.710734	3646.369645	0.727665	73.565937	
min	0.000000	0.000000	0.000000	1.000000	0.000000	
25%	7.000000	350.000000	3073.000000	2.000000	10.000000	
50%	12.000000	702.000000	6270.000000	2.000000	15.000000	
75%	21.000000	1050.000000	9412.000000	2.000000	30.000000	
max	30.000000	1427.000000	12567.000000	9.000000	995.000000	

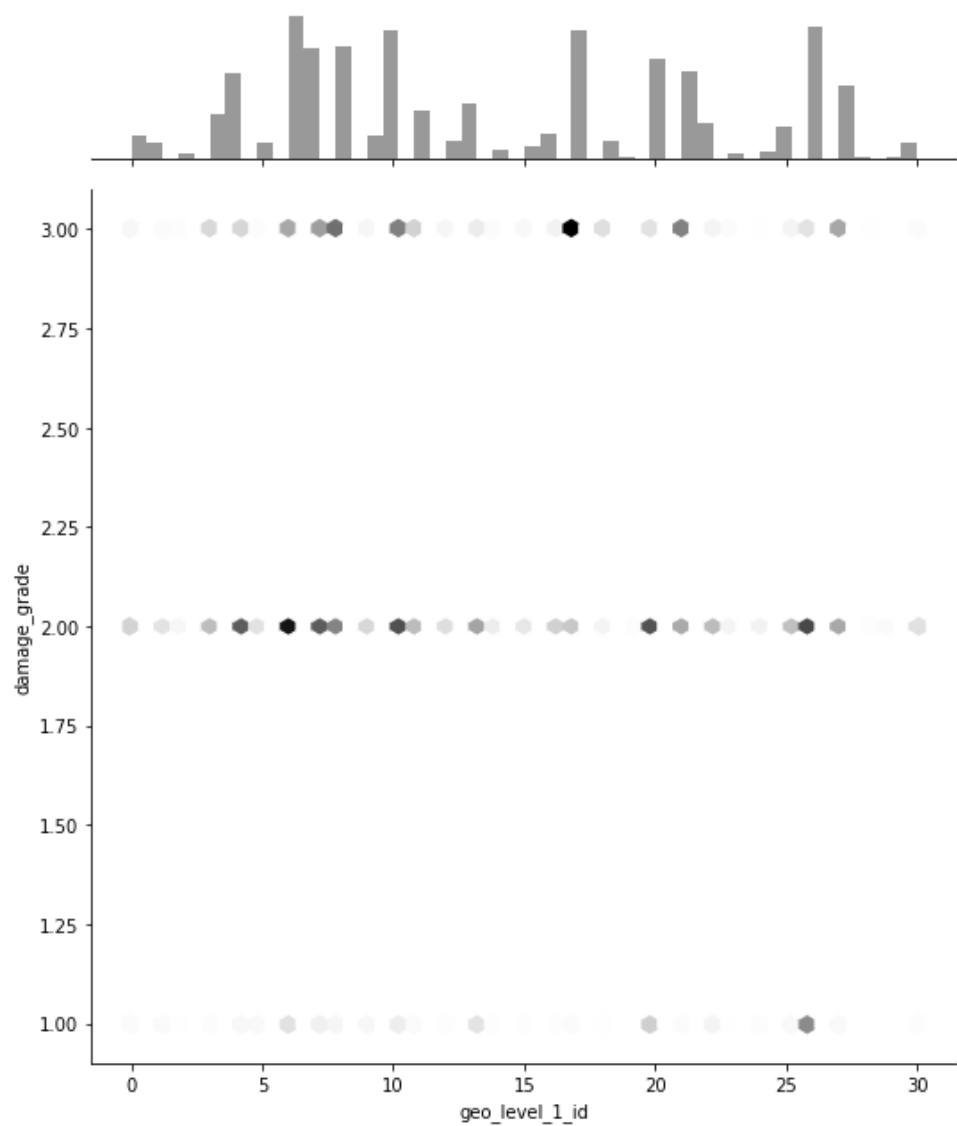
8 rows × 31 columns

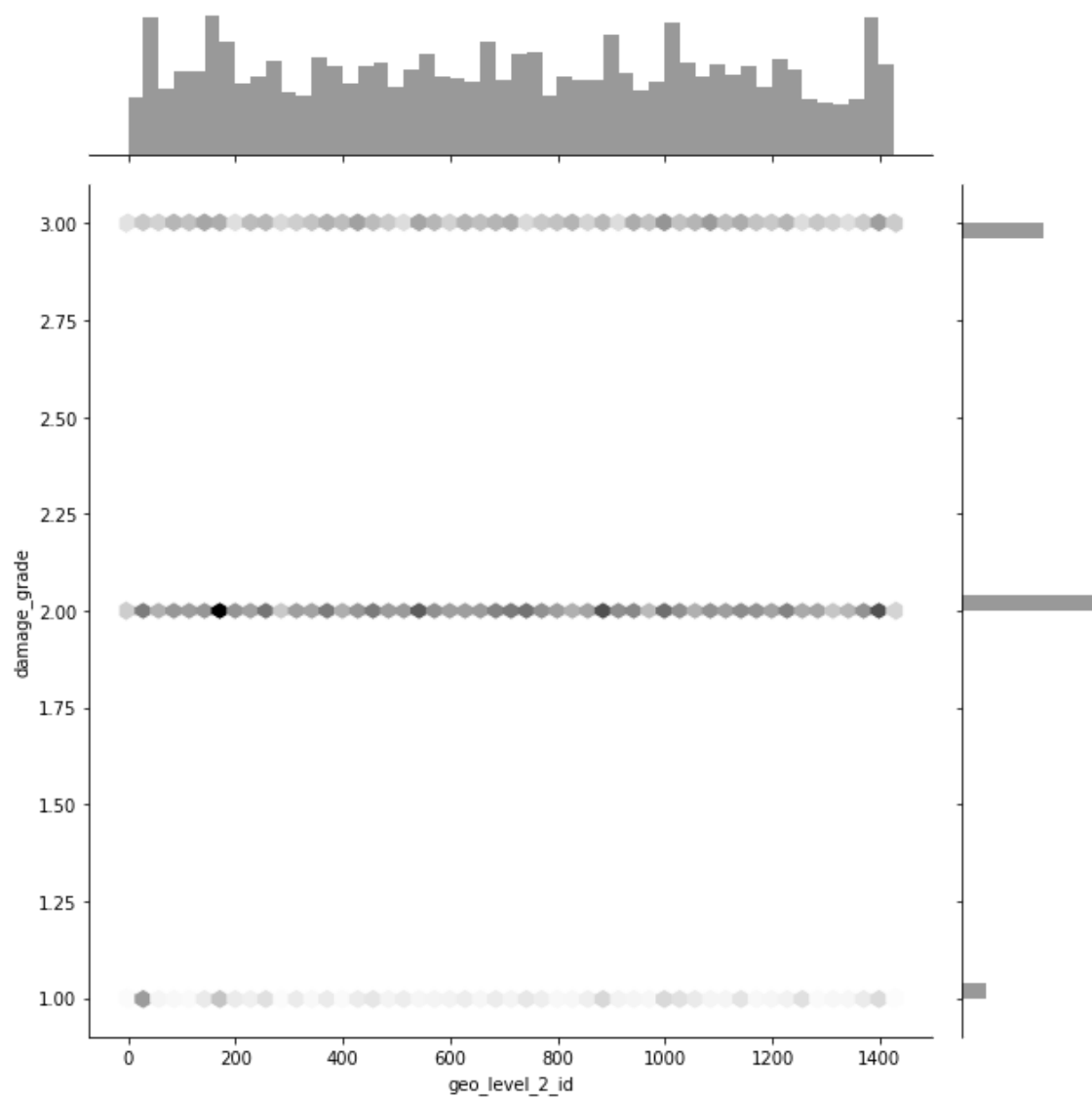


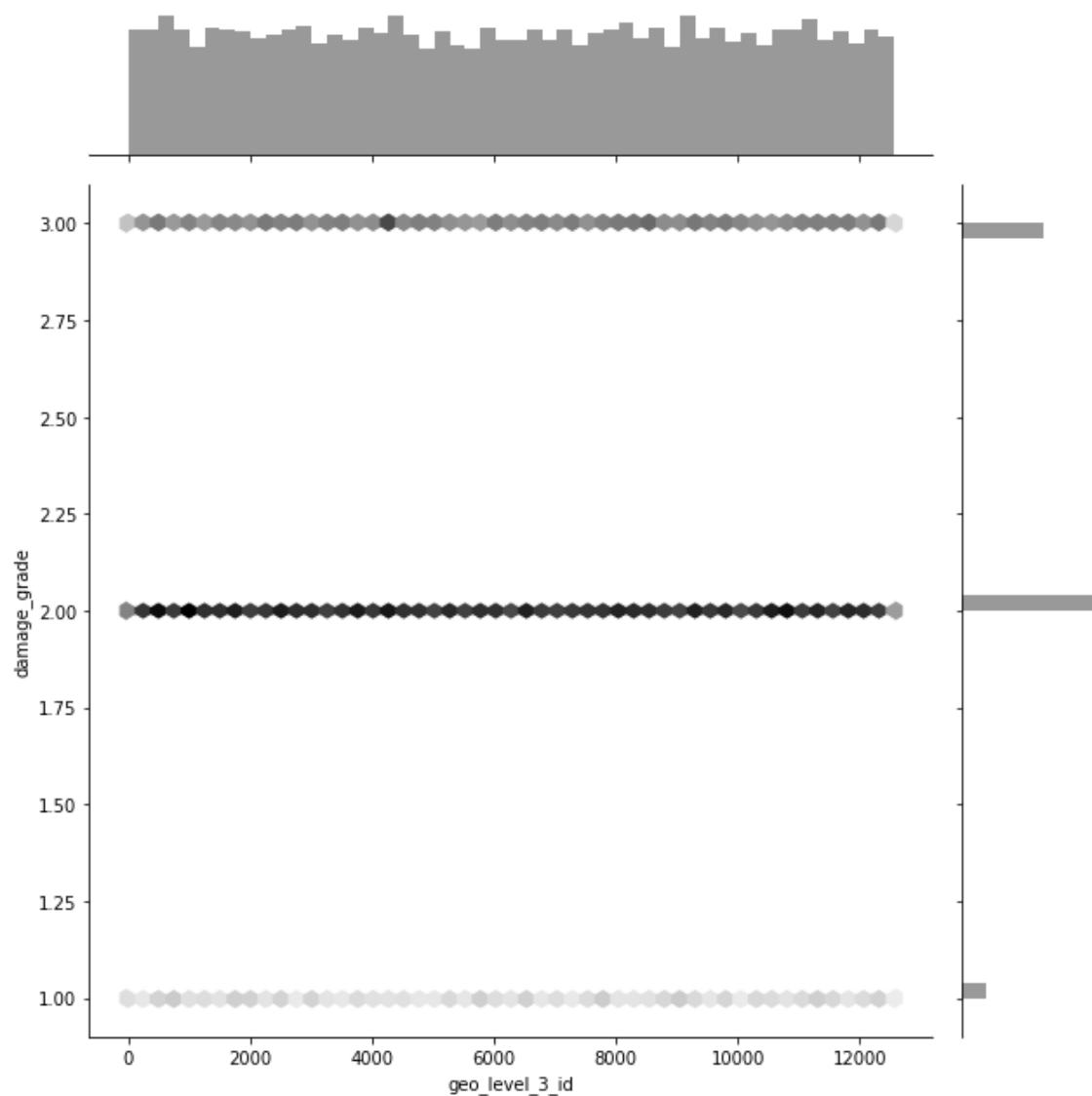
In [8]:

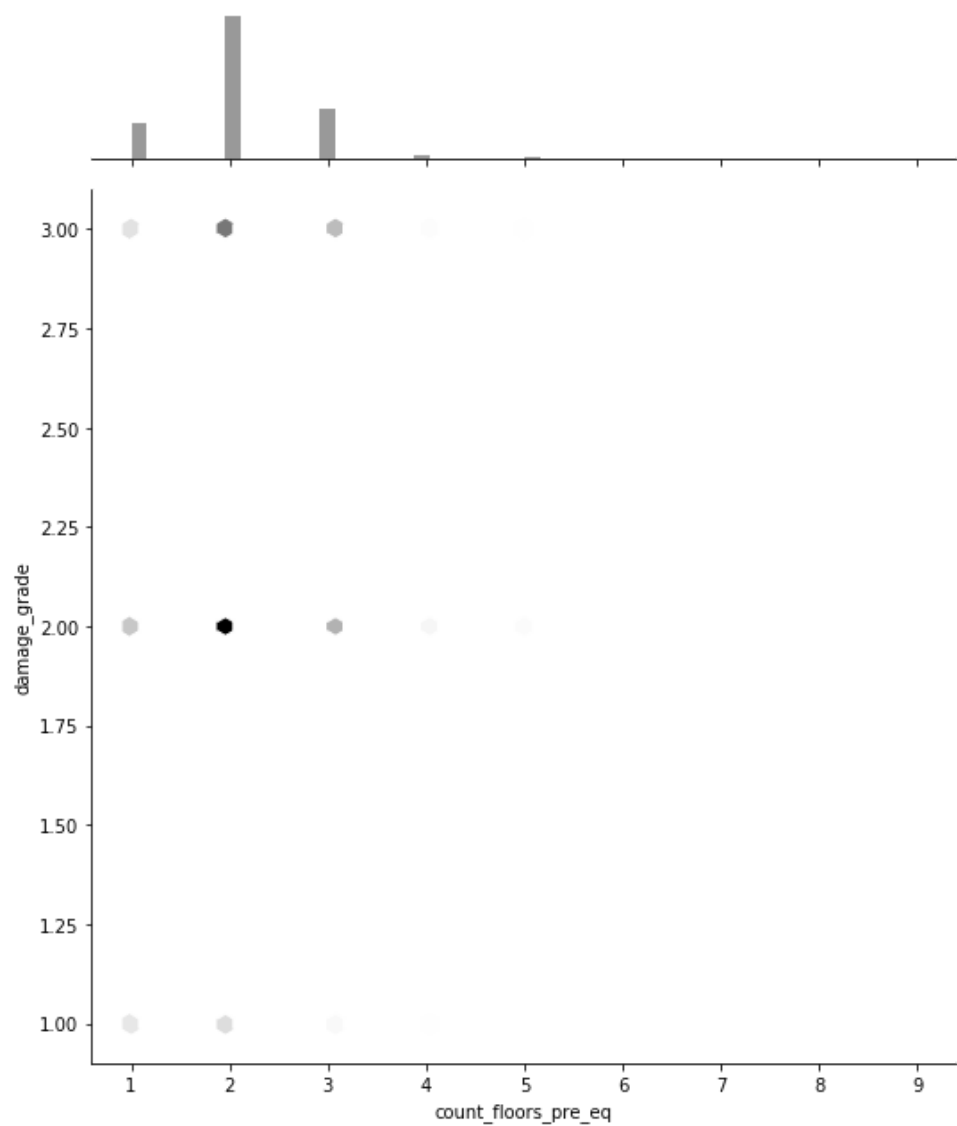
```
for col in sns_data.columns:  
#     print('\nTratando Columna: '+col)  
    sns.jointplot(sns_data[col], sns_data['damage_grade'], kind="hex", color="k", height=9)
```

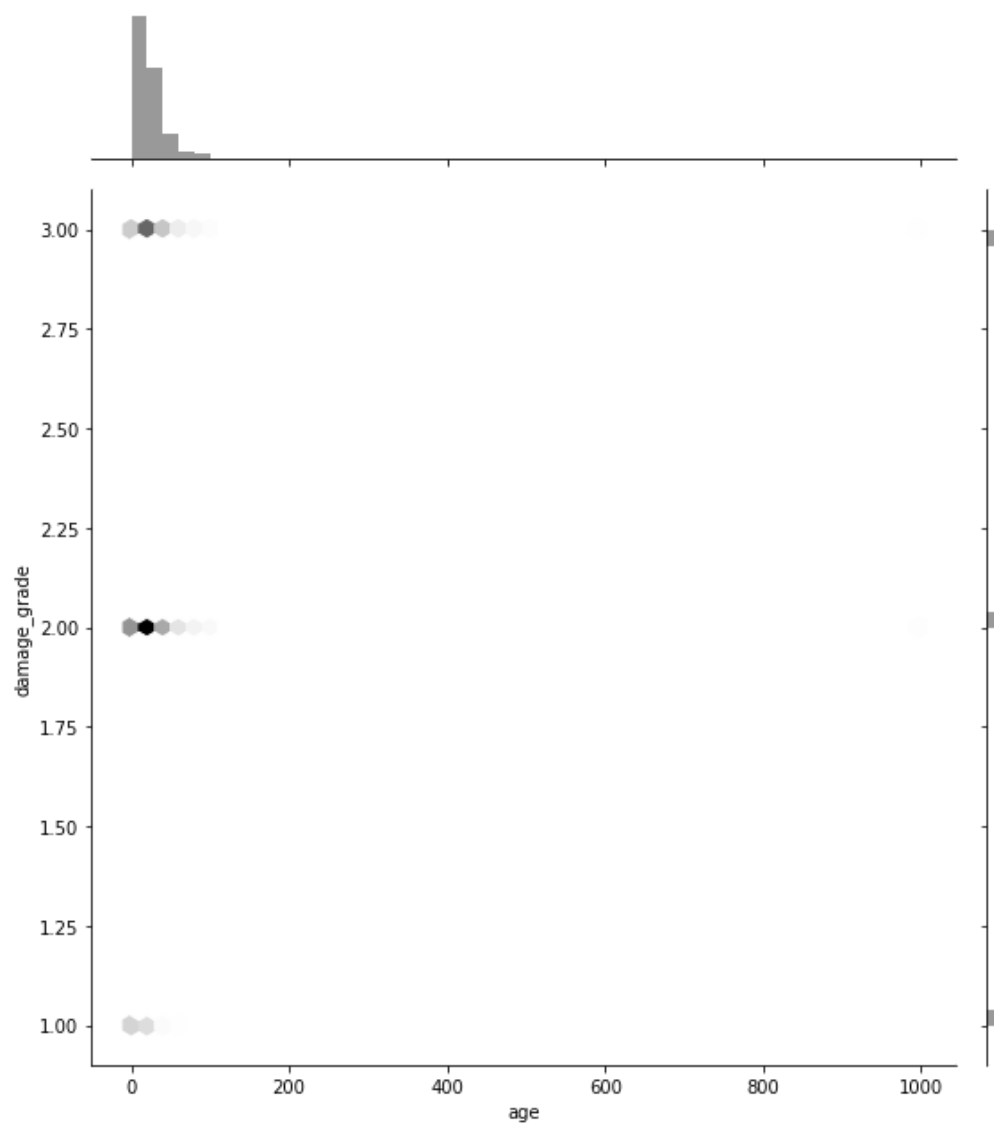
```
/usr/local/lib/python3.5/dist-packages/matplotlib/pyplot.py:514: RuntimeWarning: More than 20 figures have been opened. Figures created through the pyplot interface (`matplotlib.pyplot.figure`) are retained until explicitly closed and may consume too much memory. (To control this warning, see the rcParam `figure.max_open_warning`).  
  max_open_warning, RuntimeWarning)
```

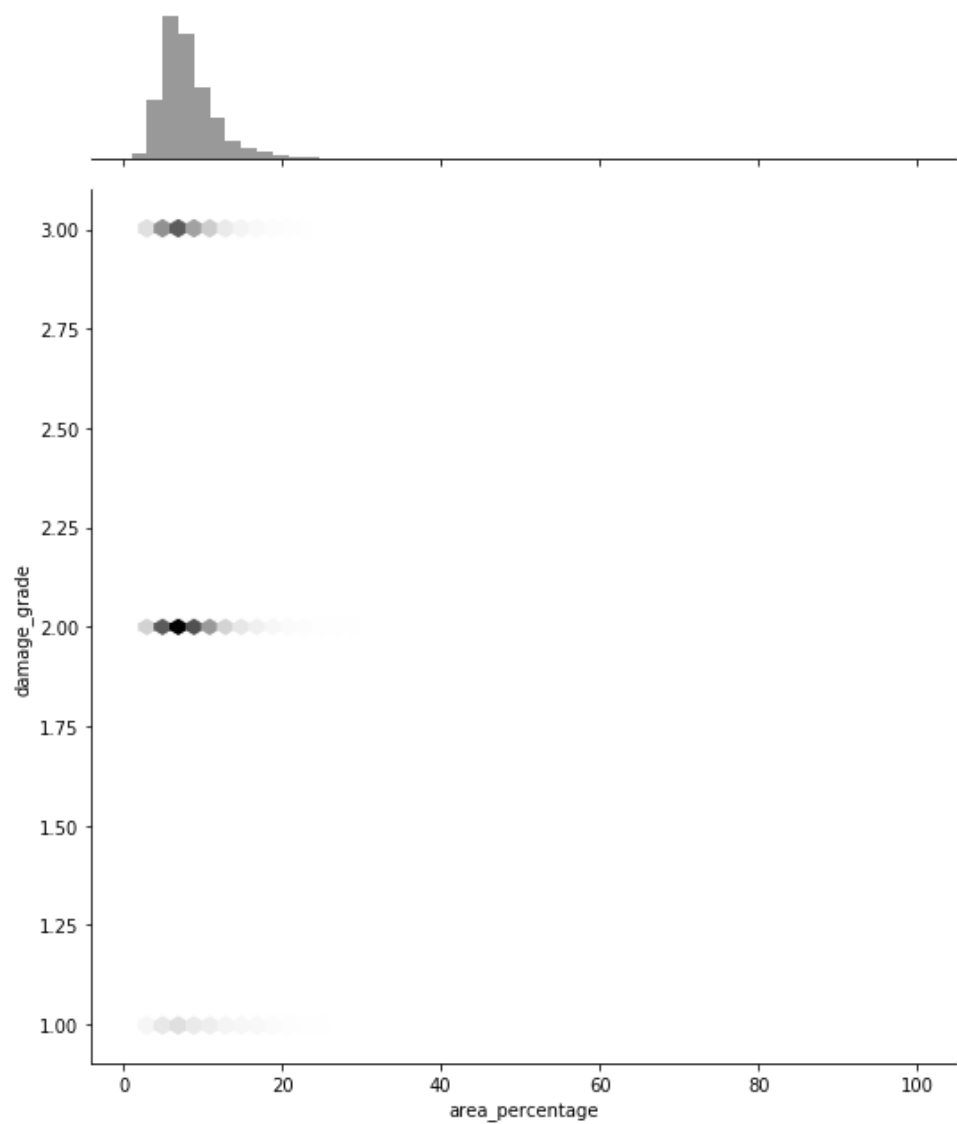


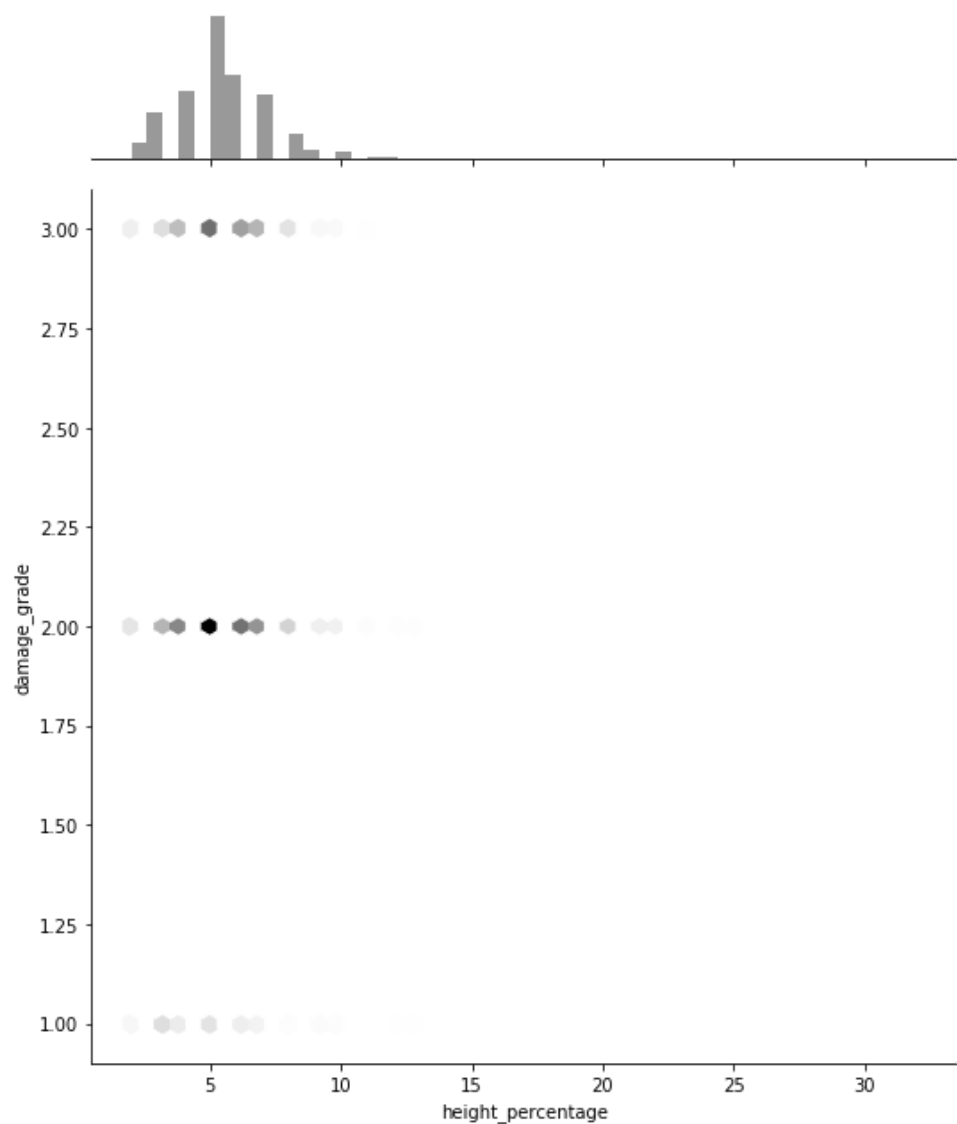


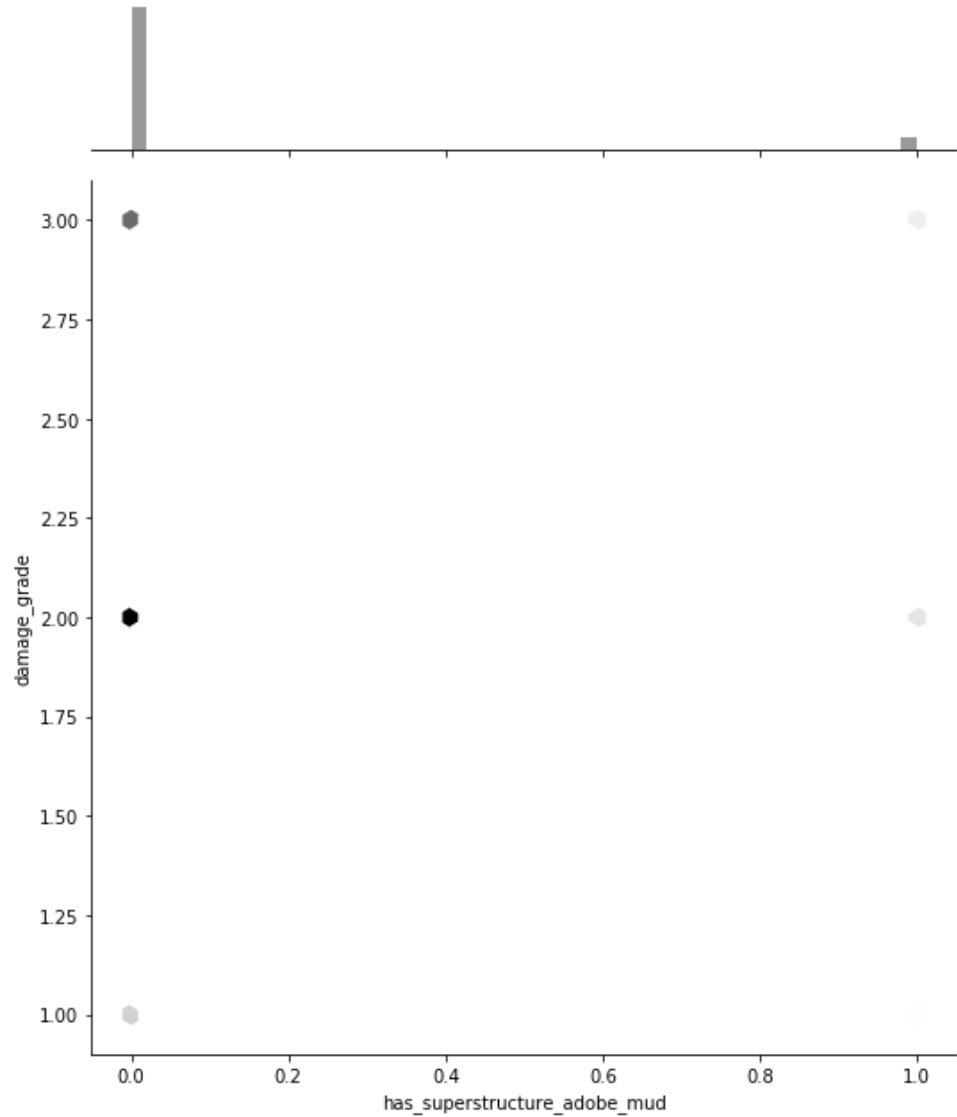


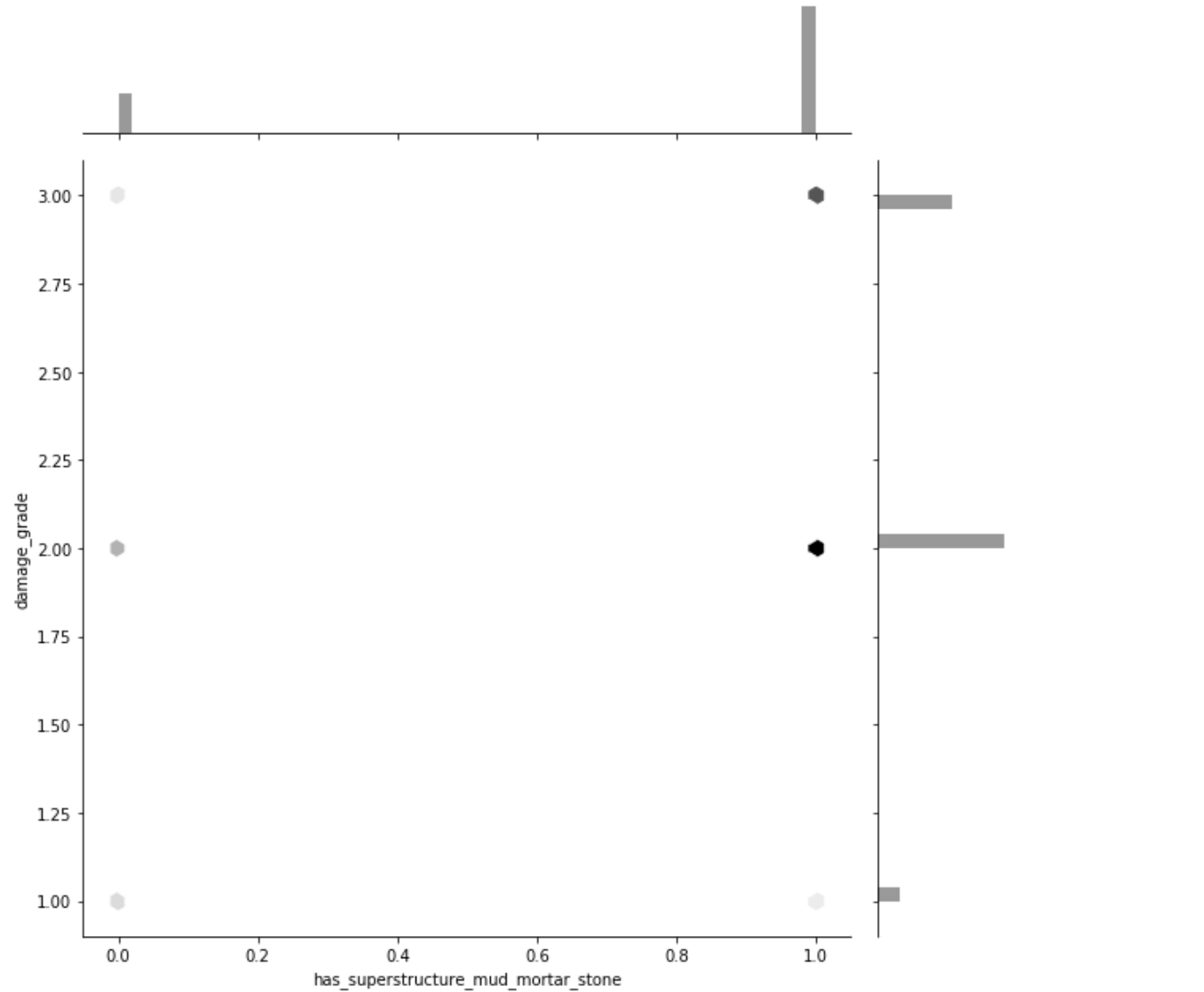


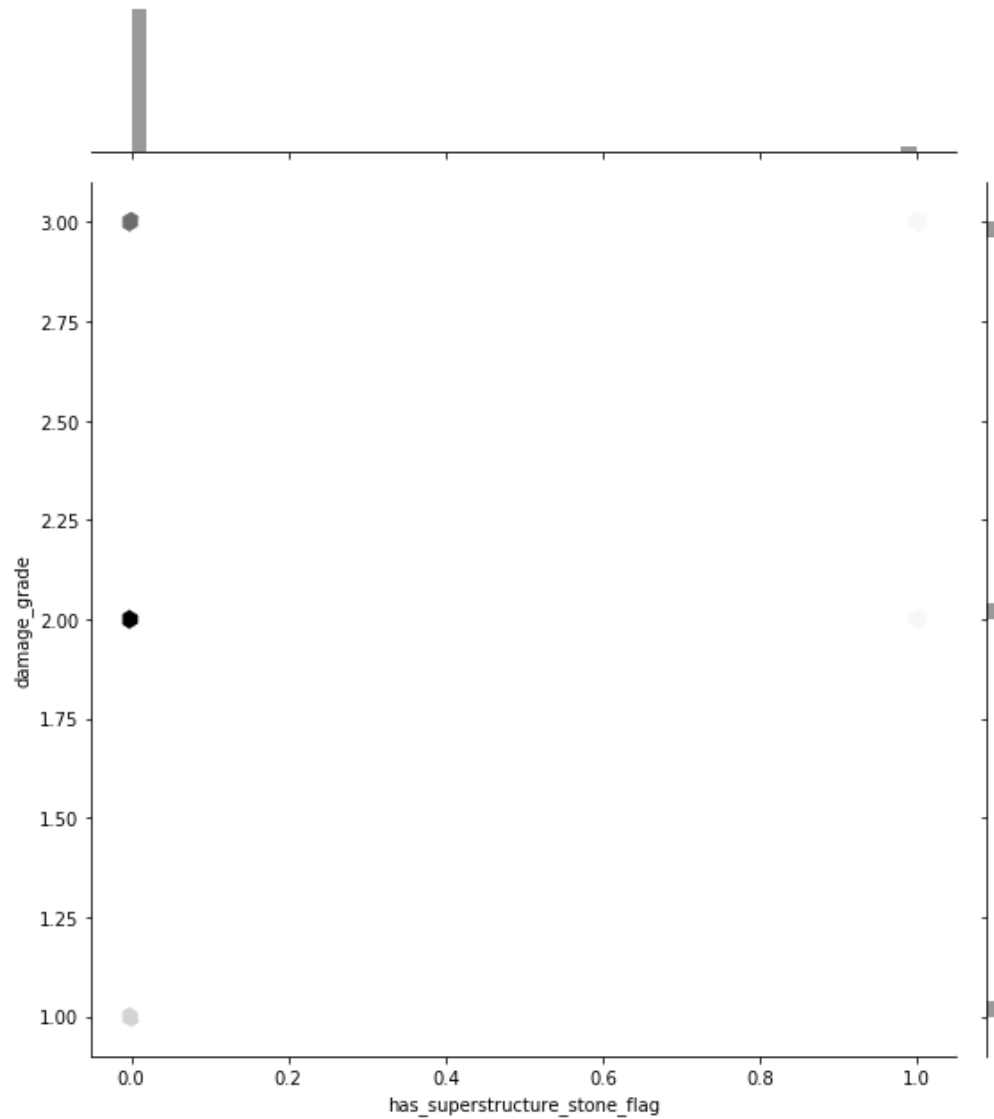


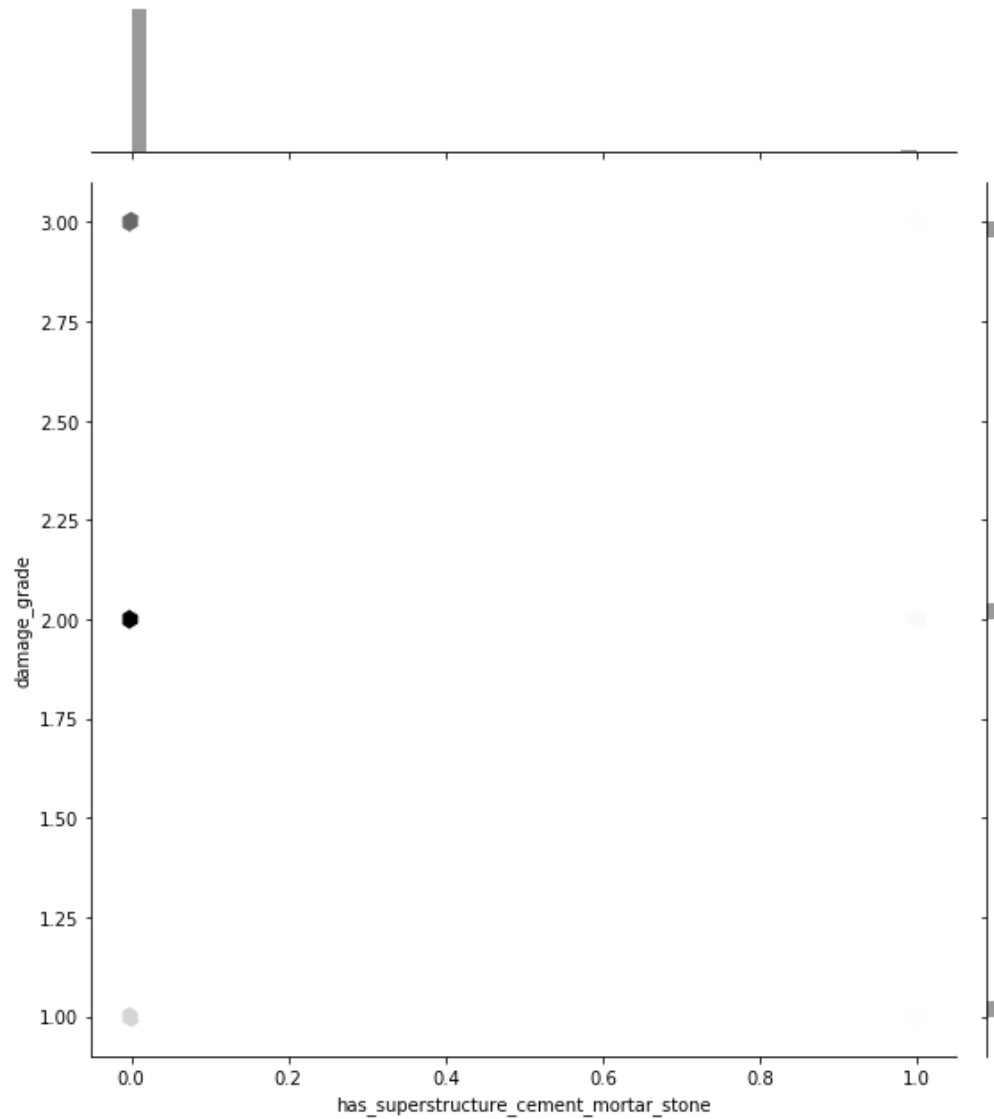


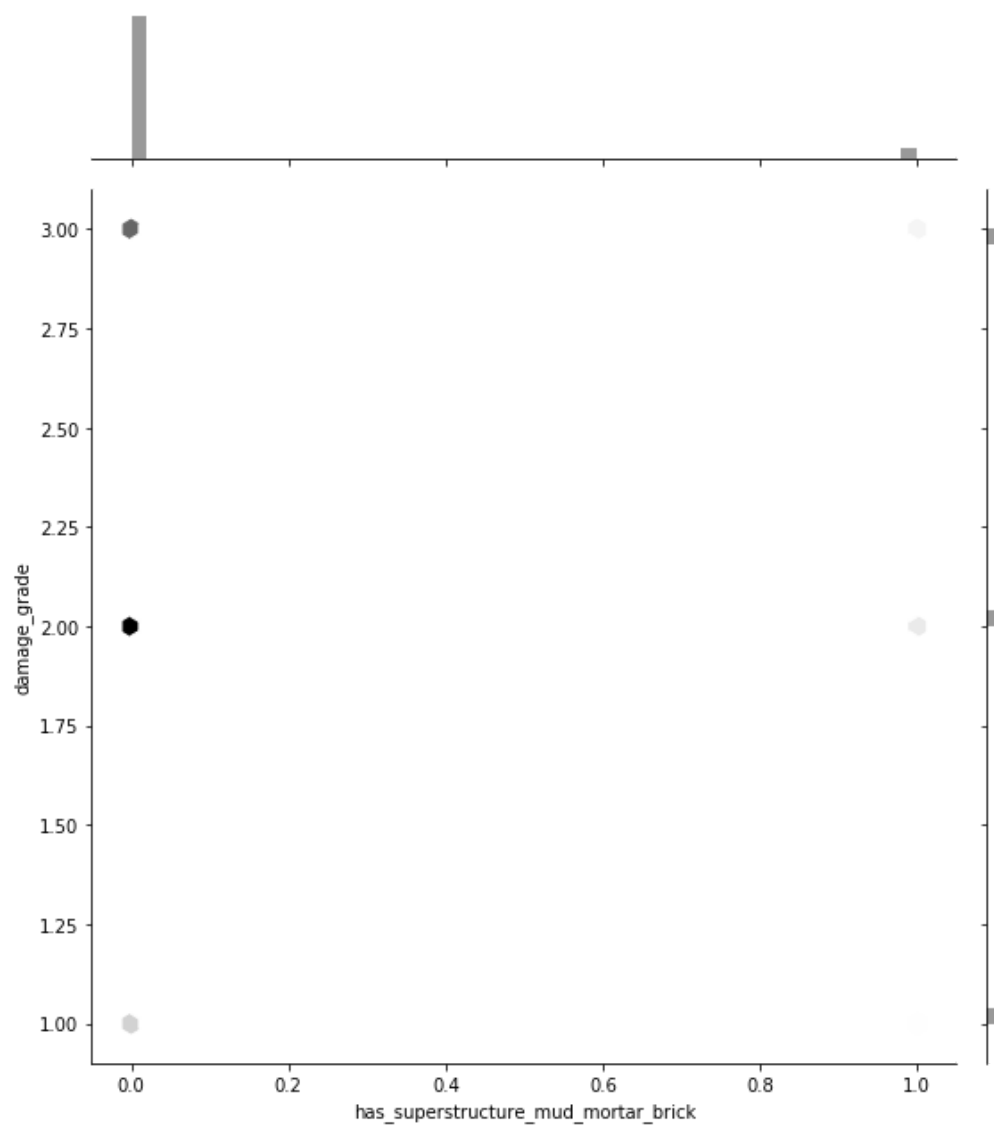


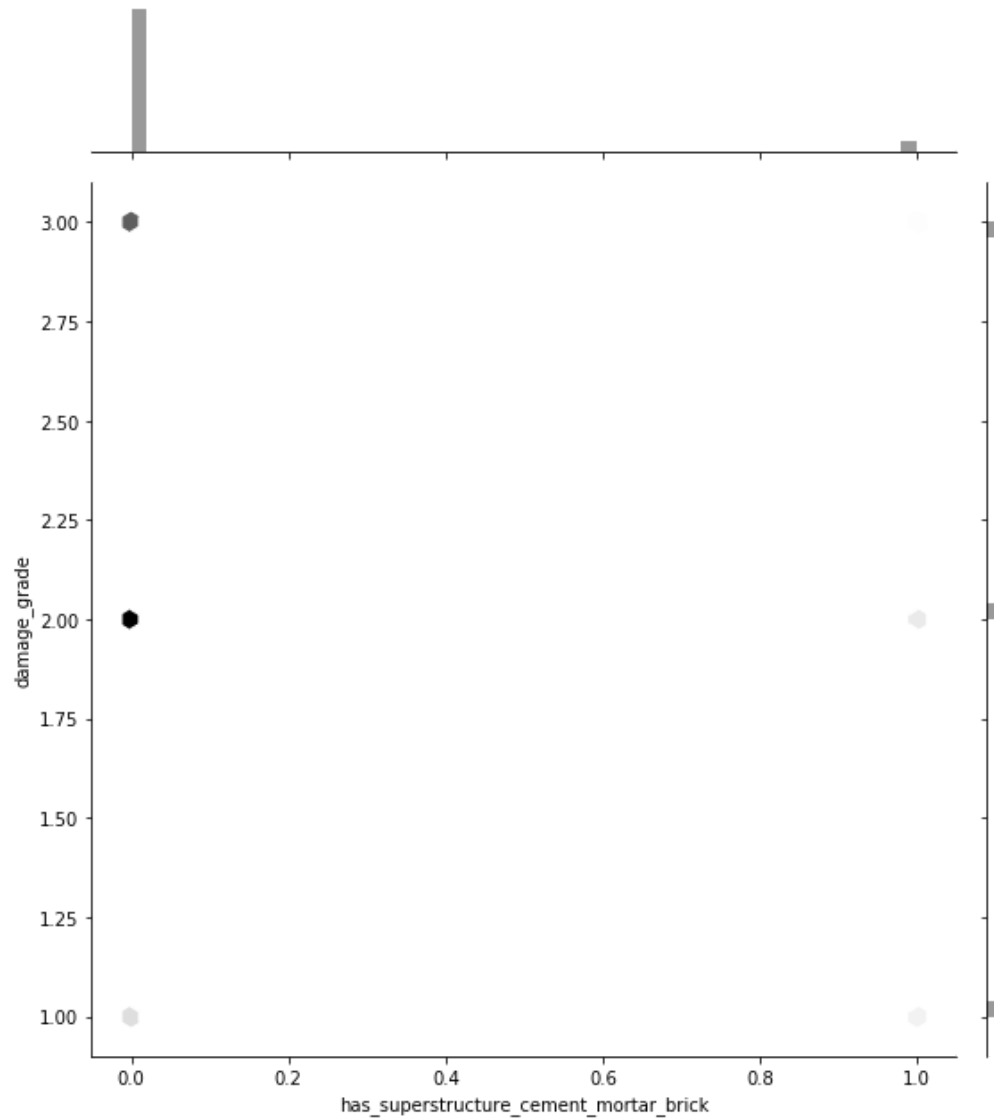


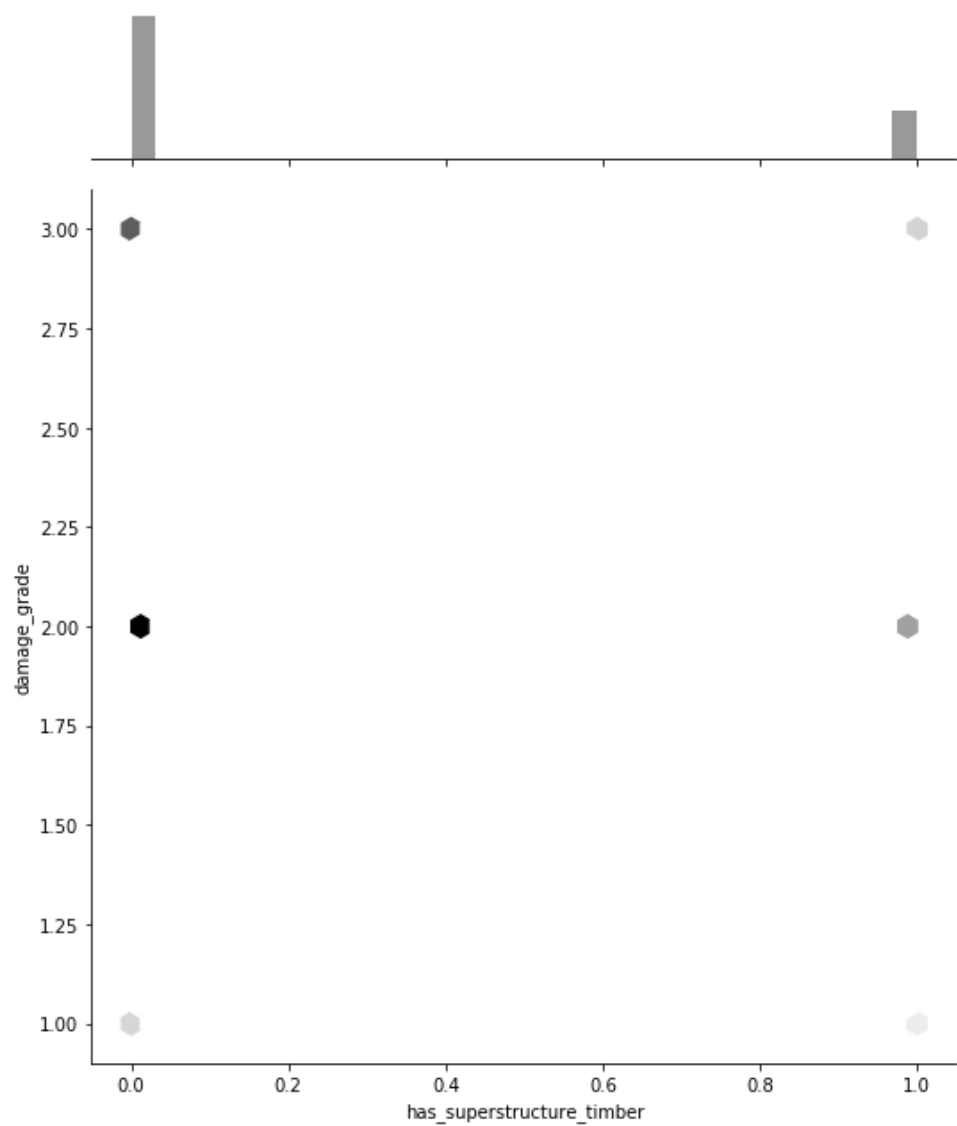


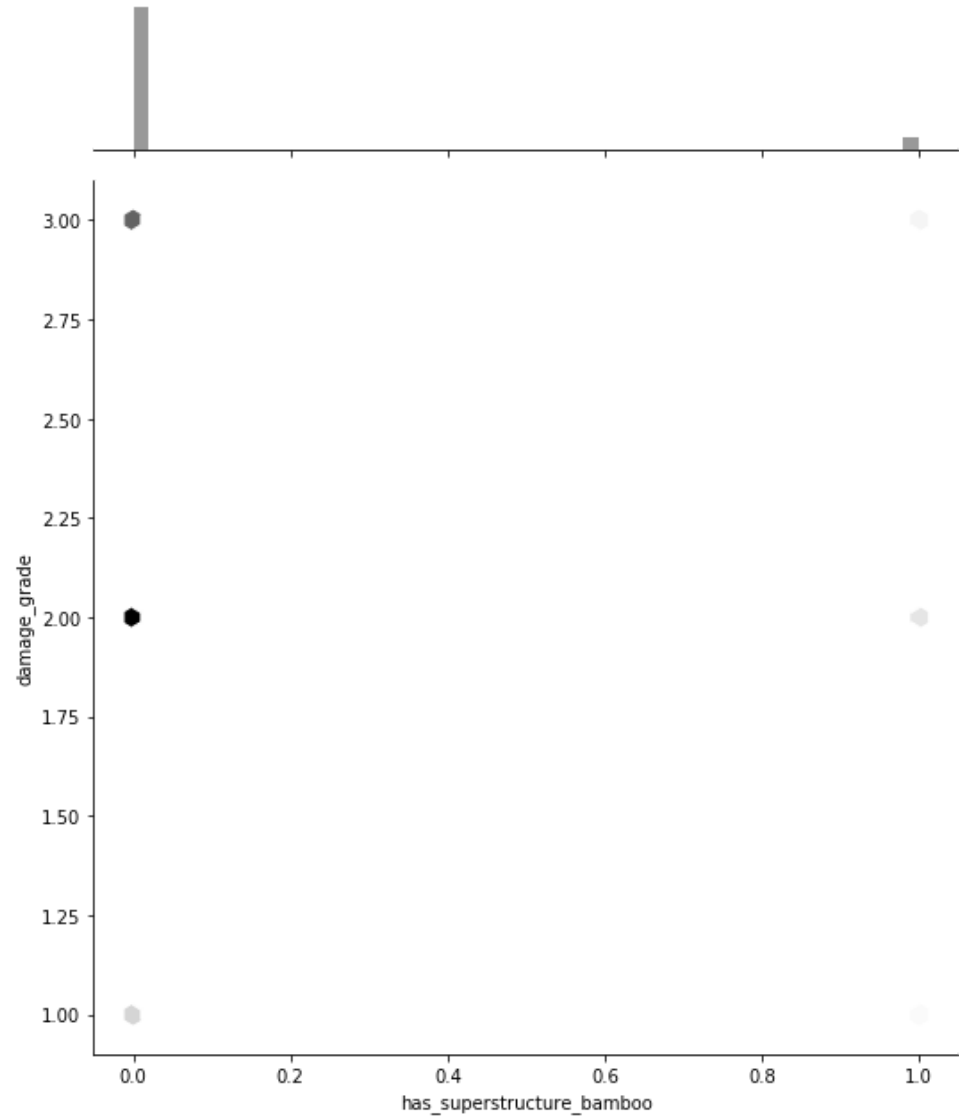


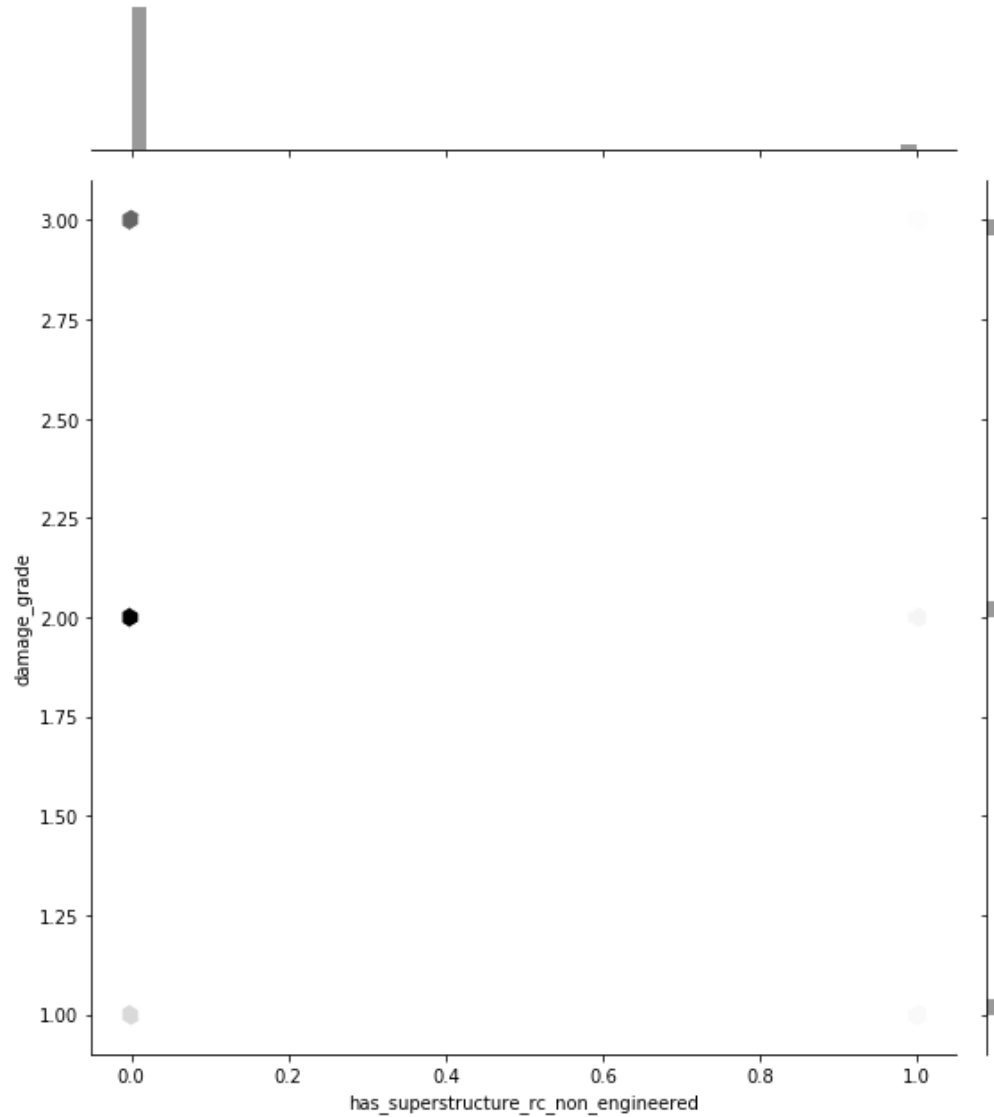


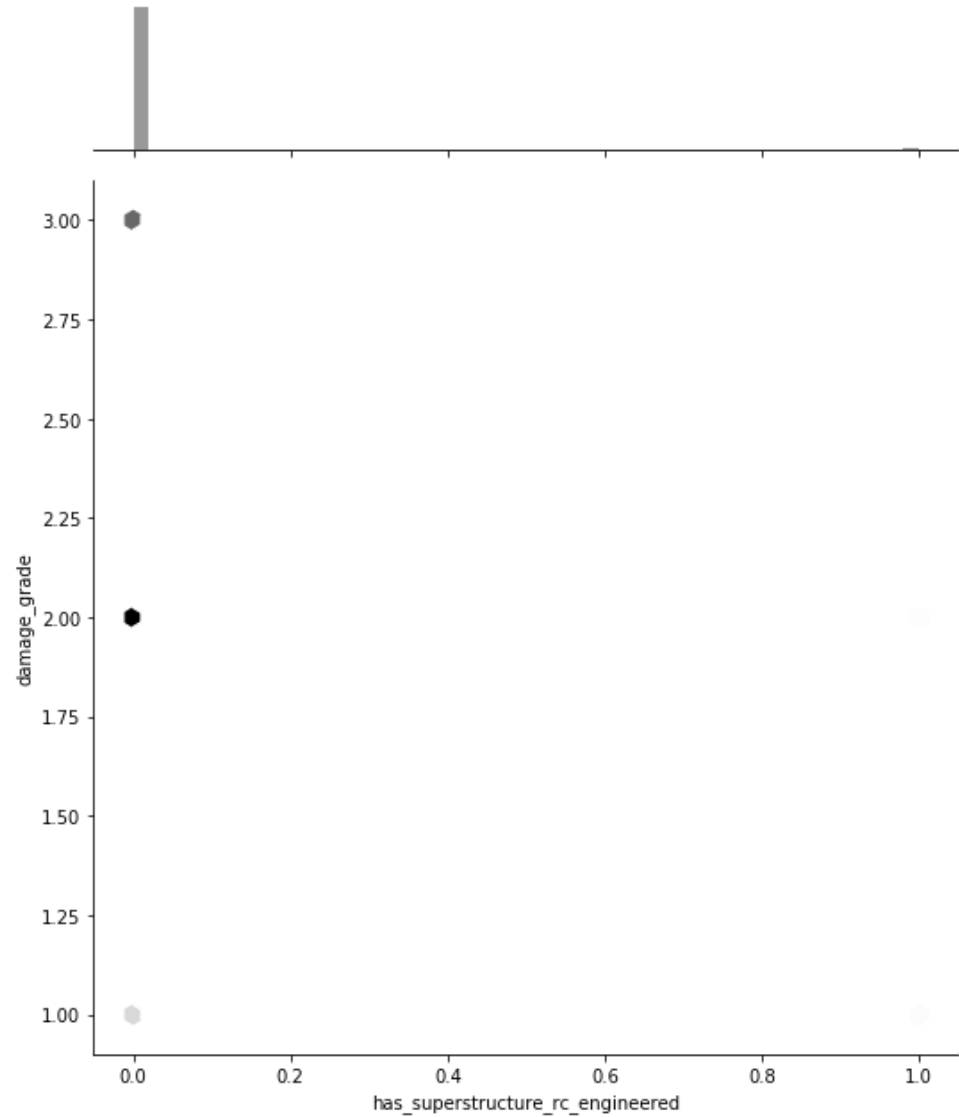


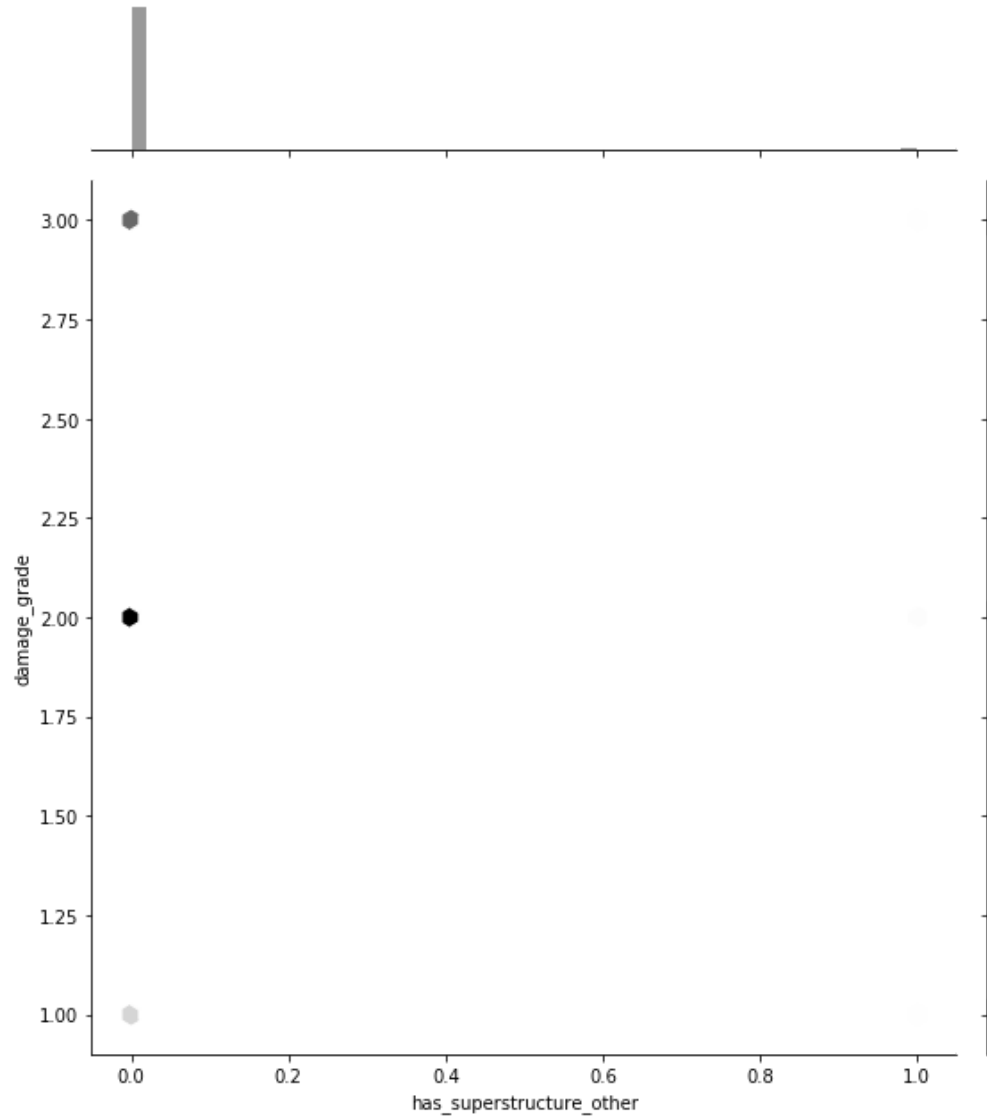


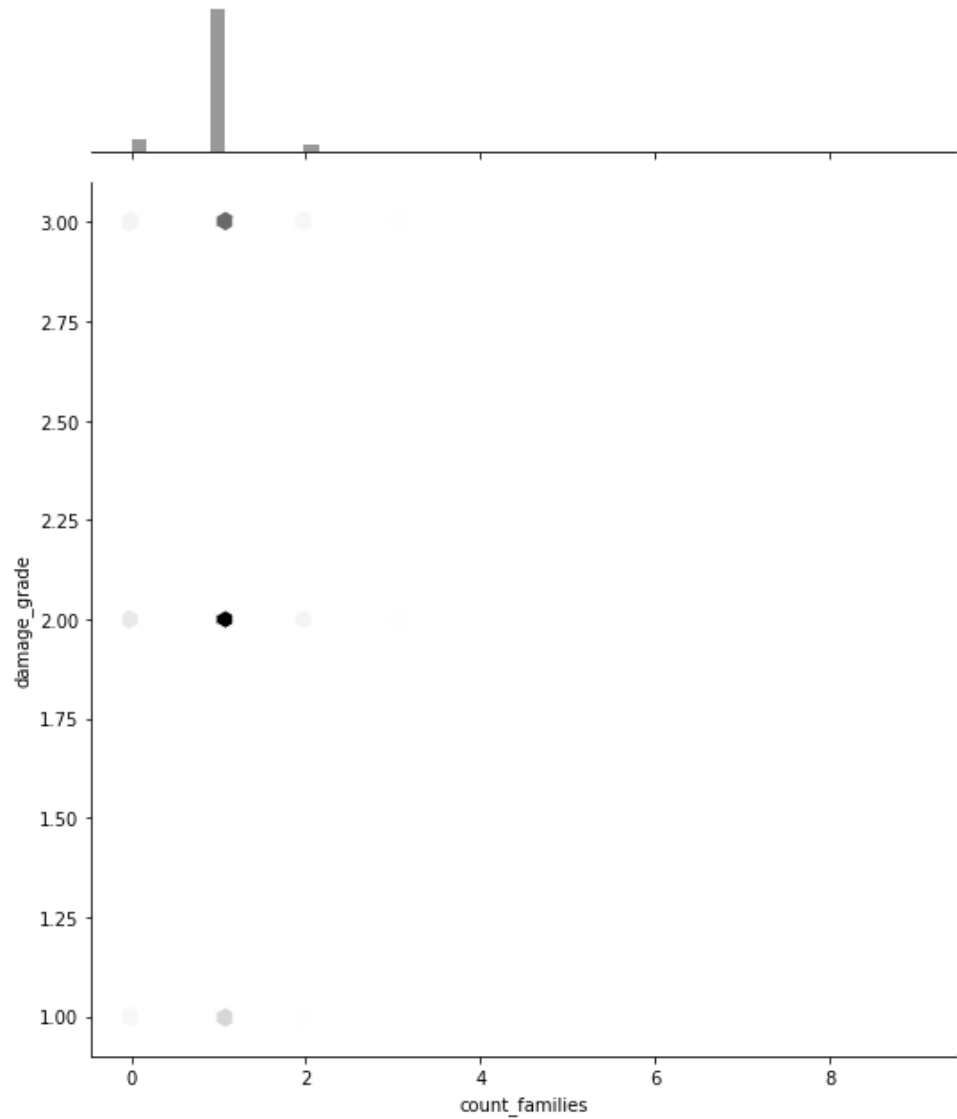


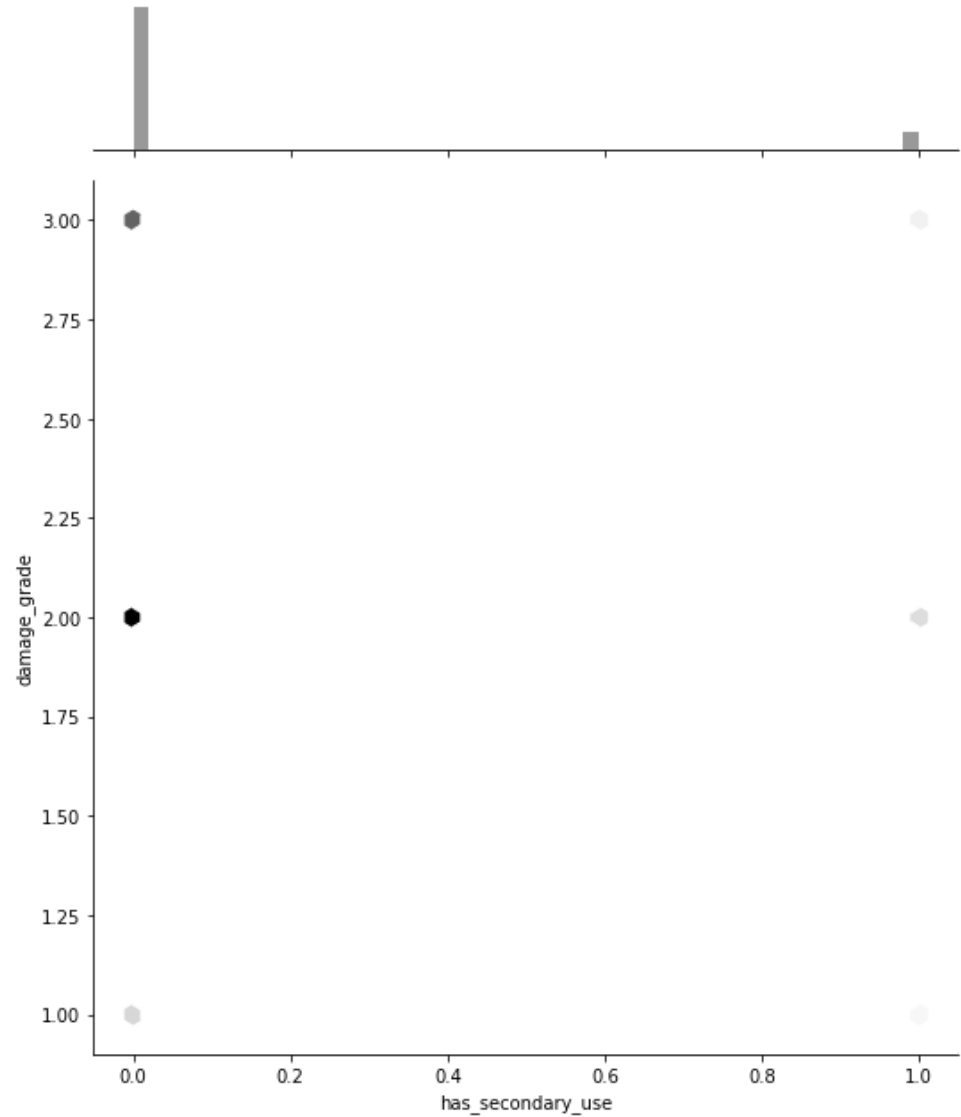


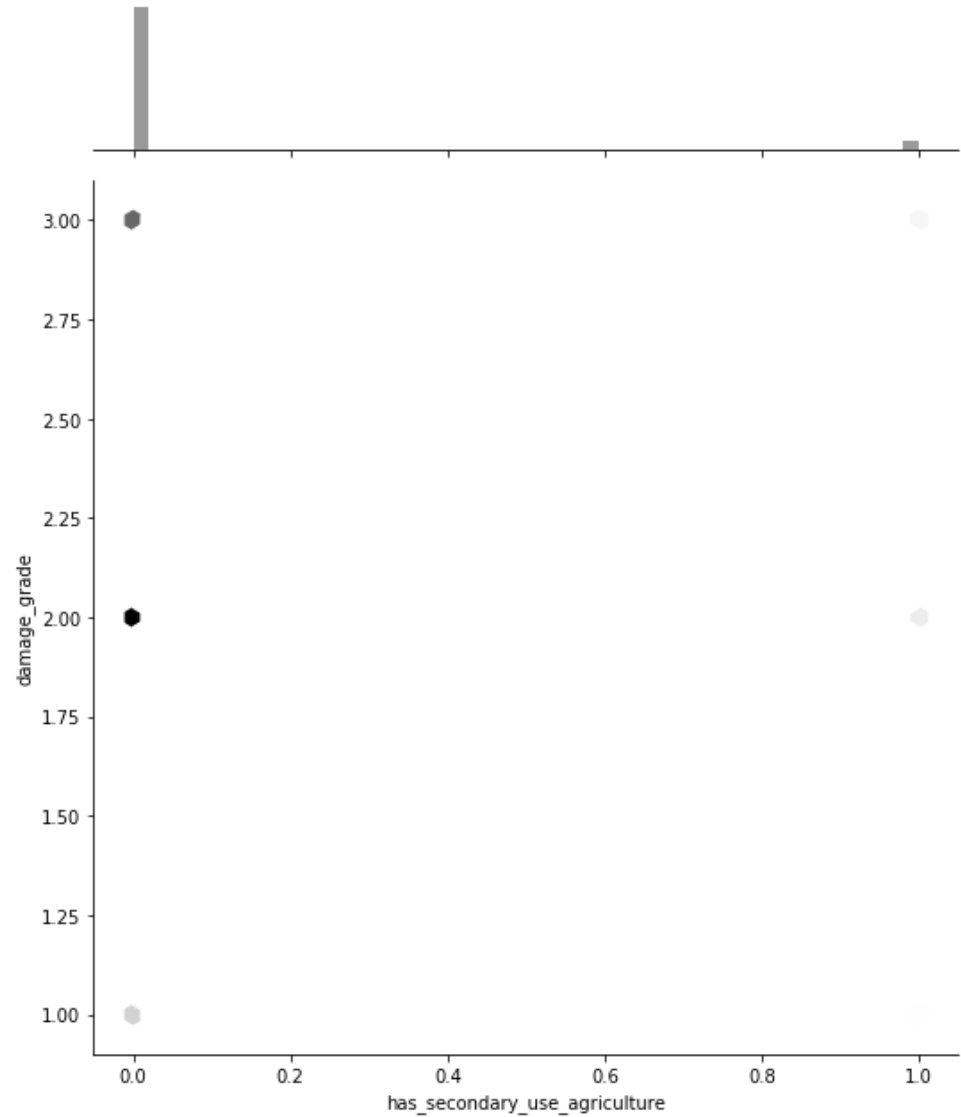


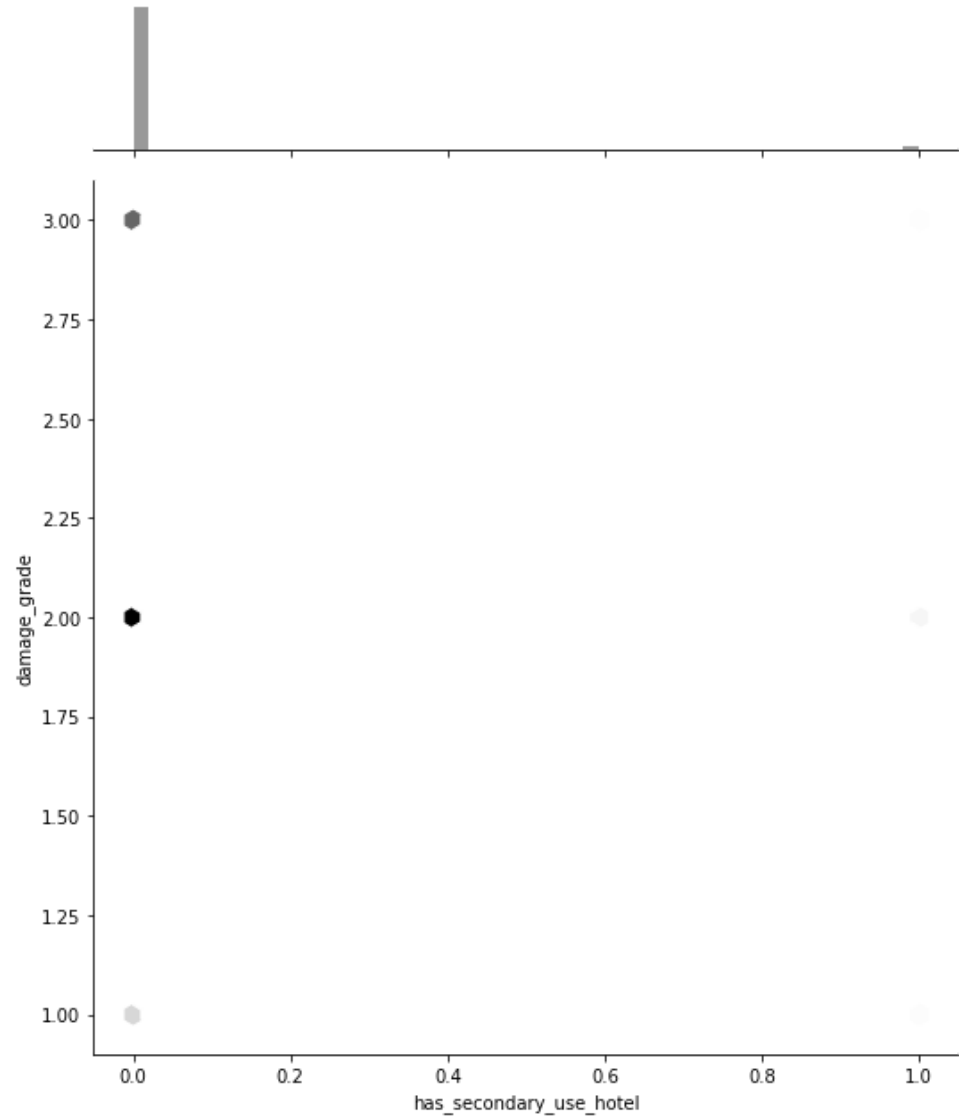


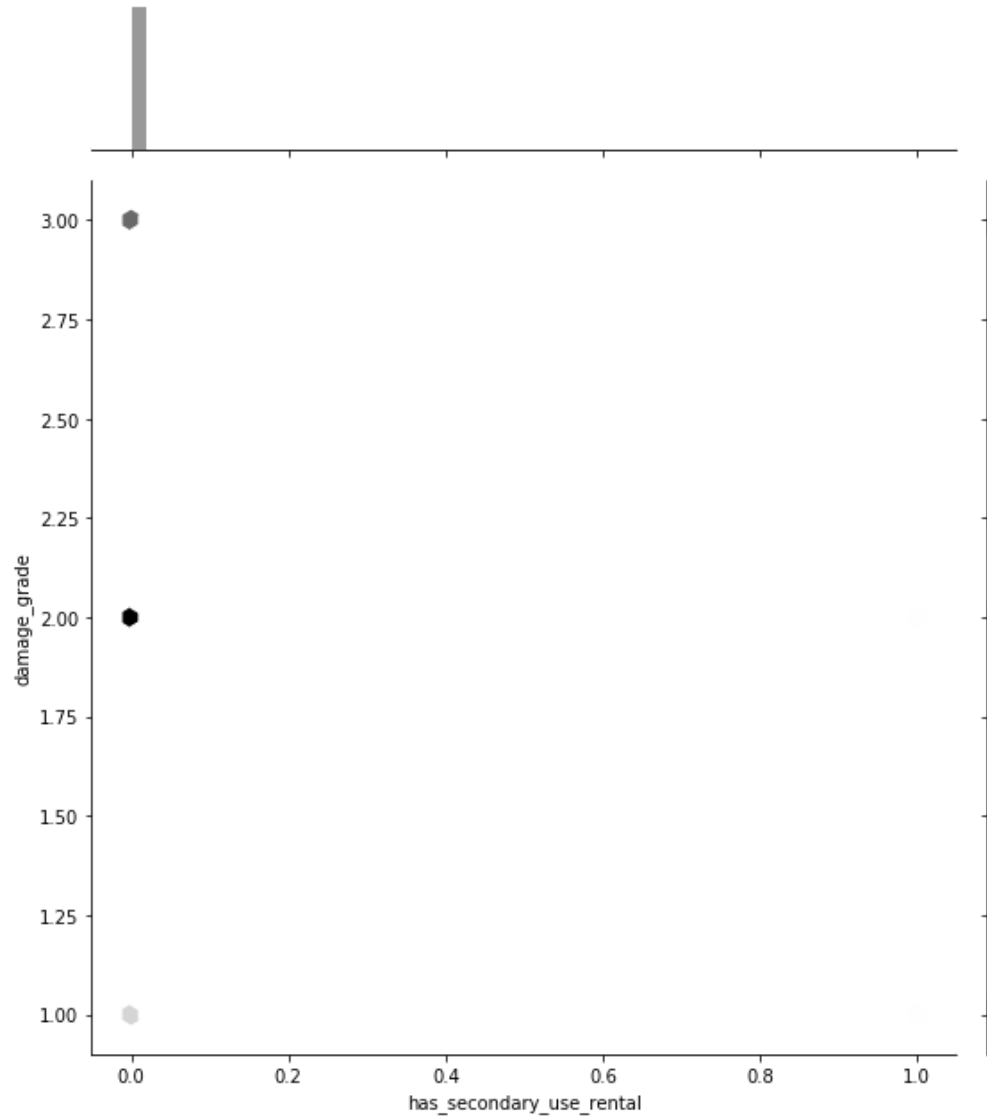


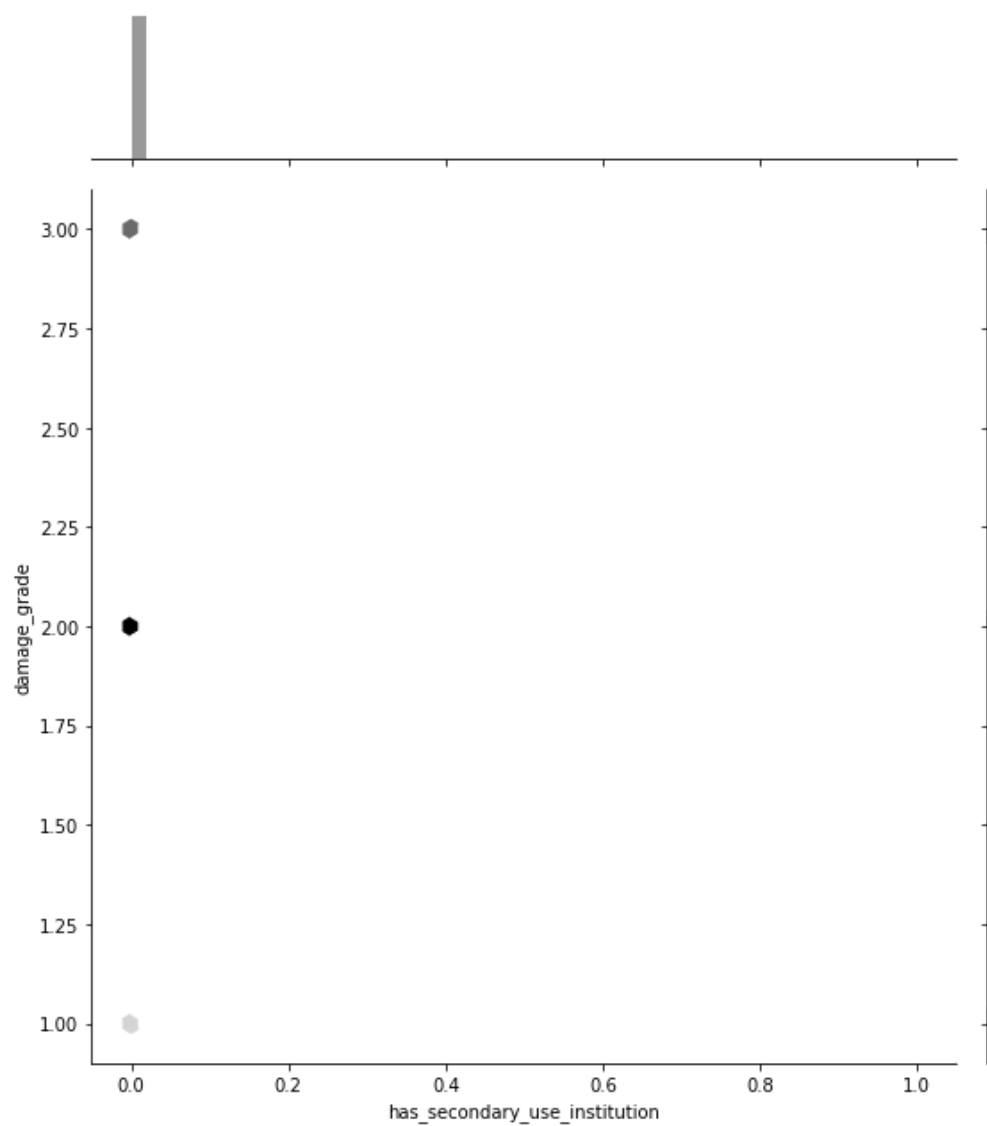


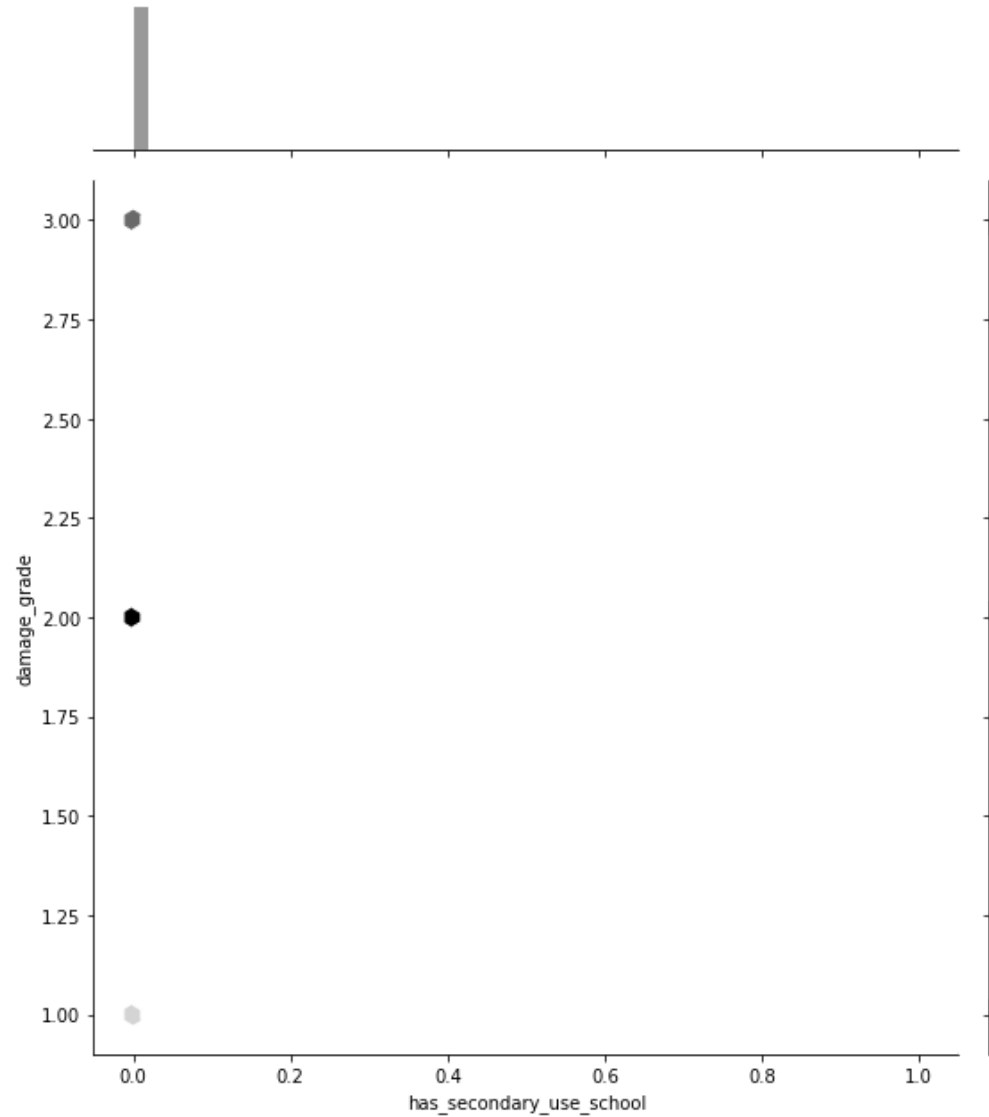


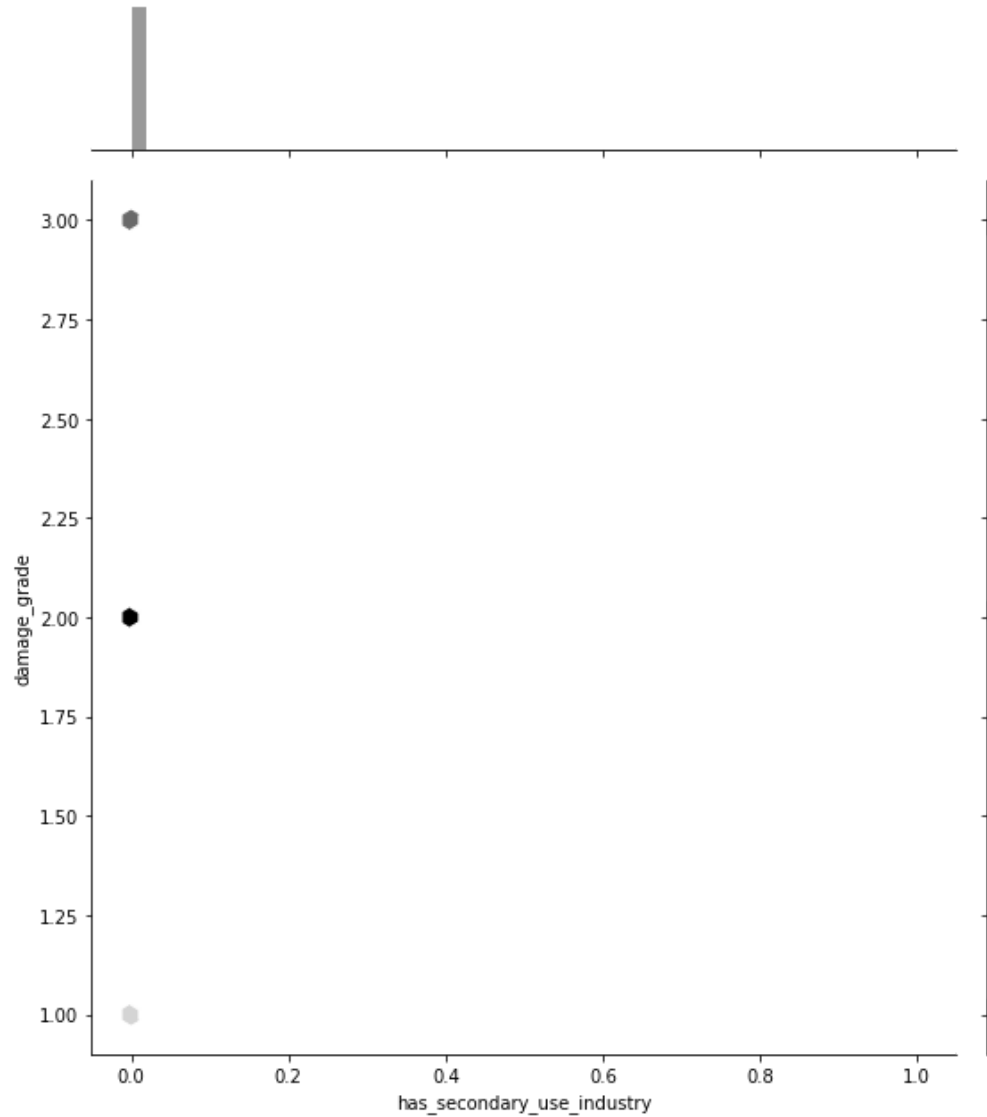


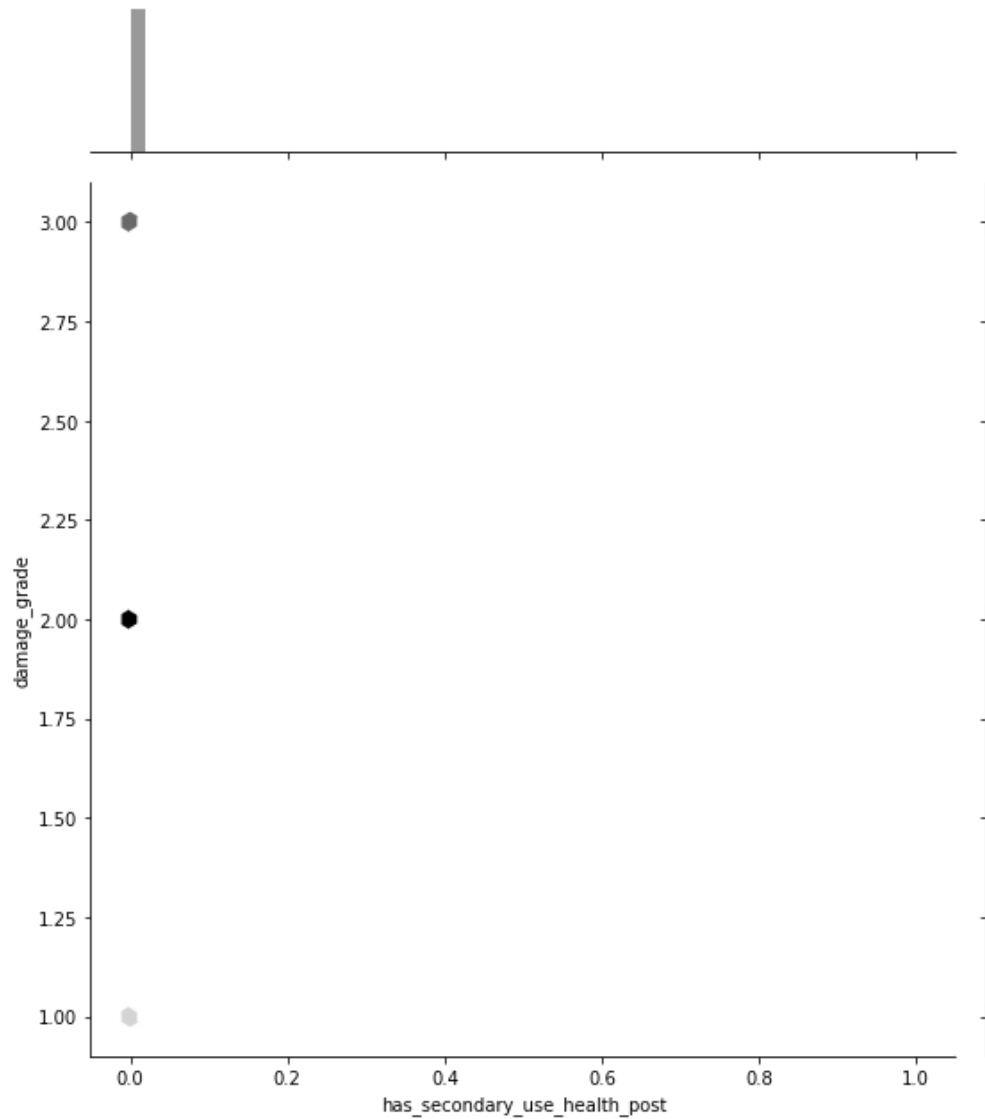


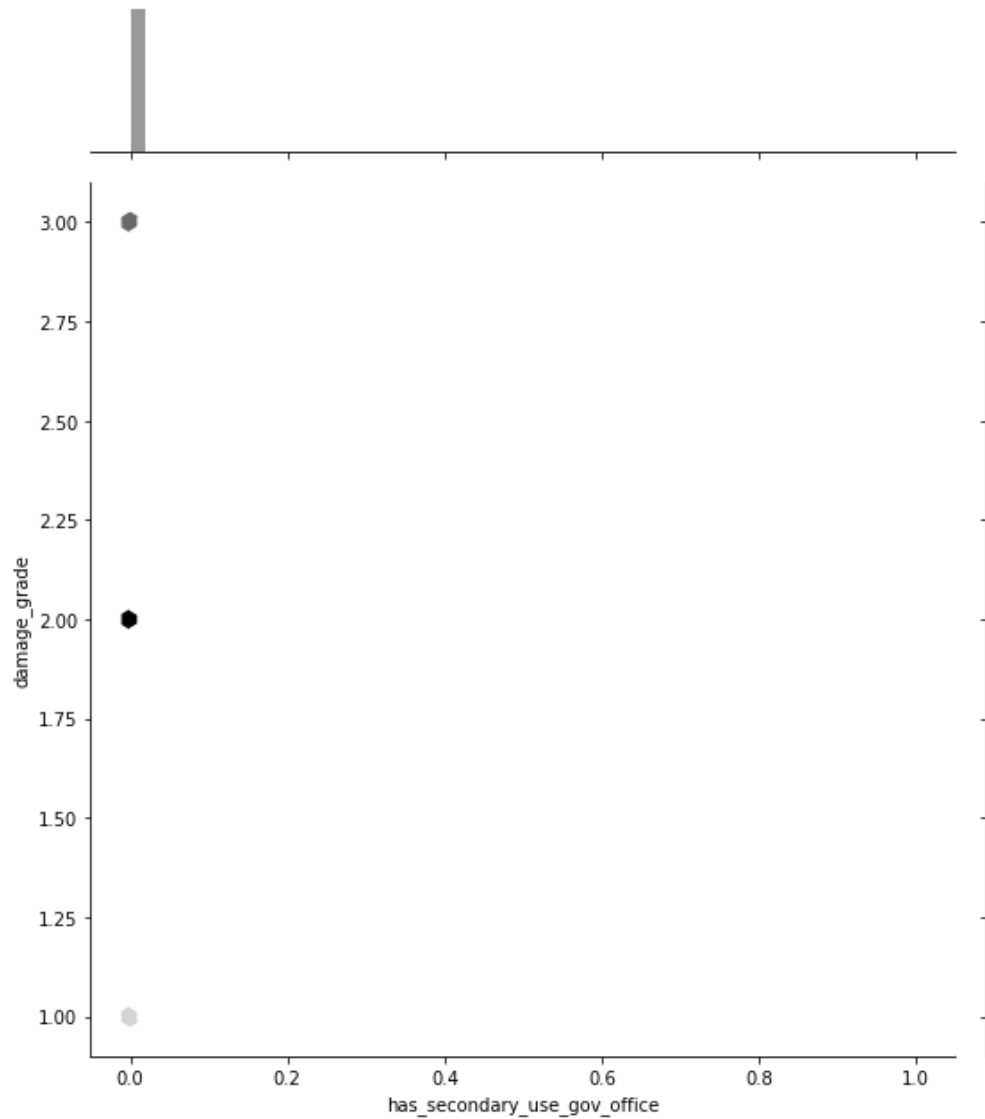


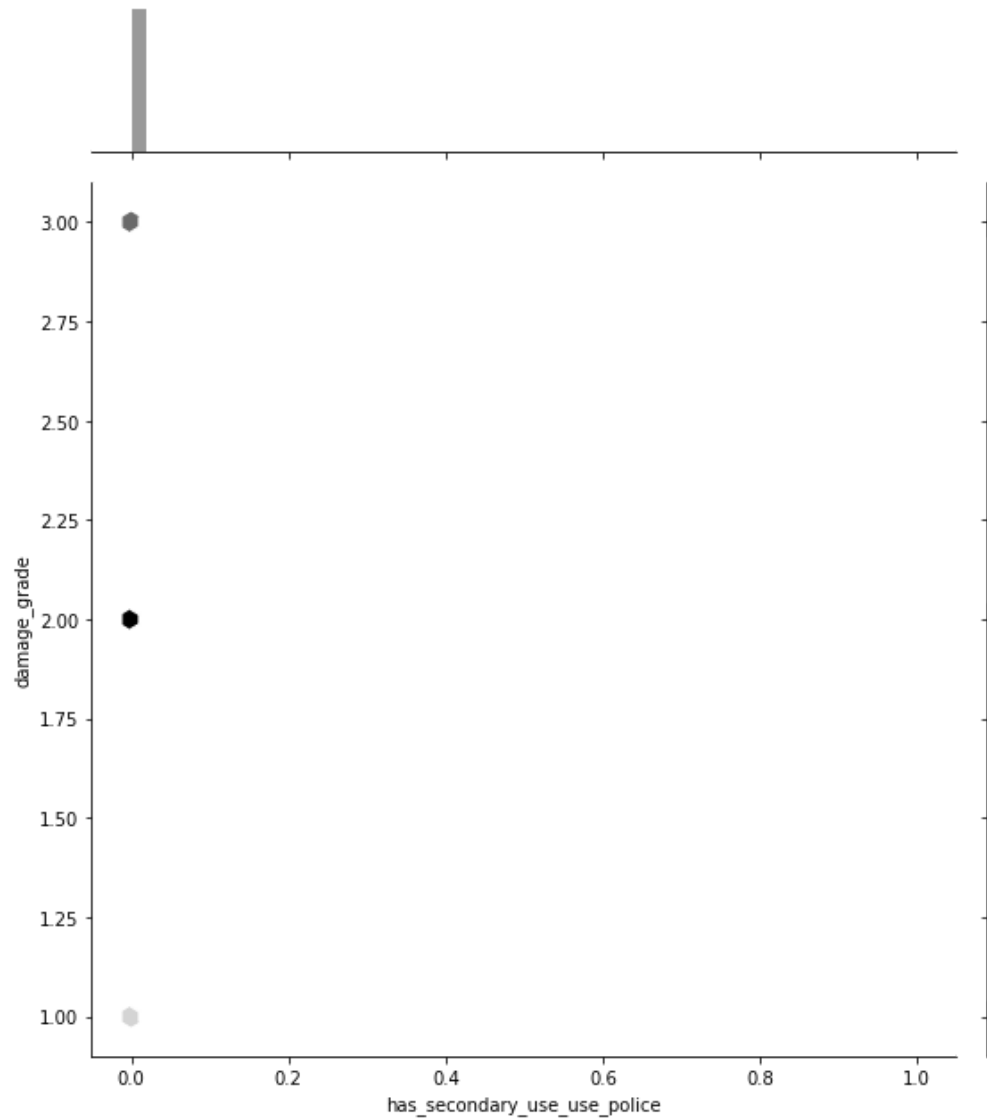


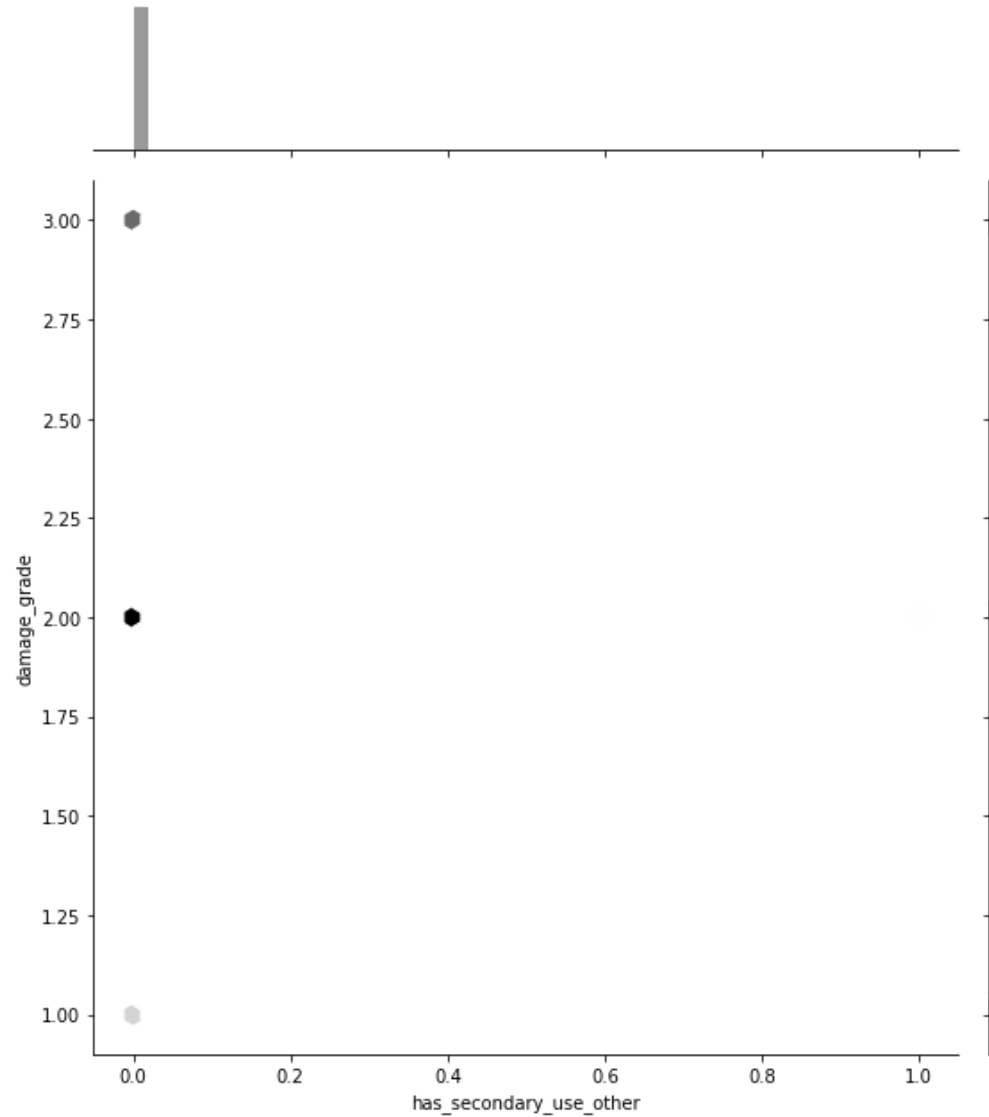


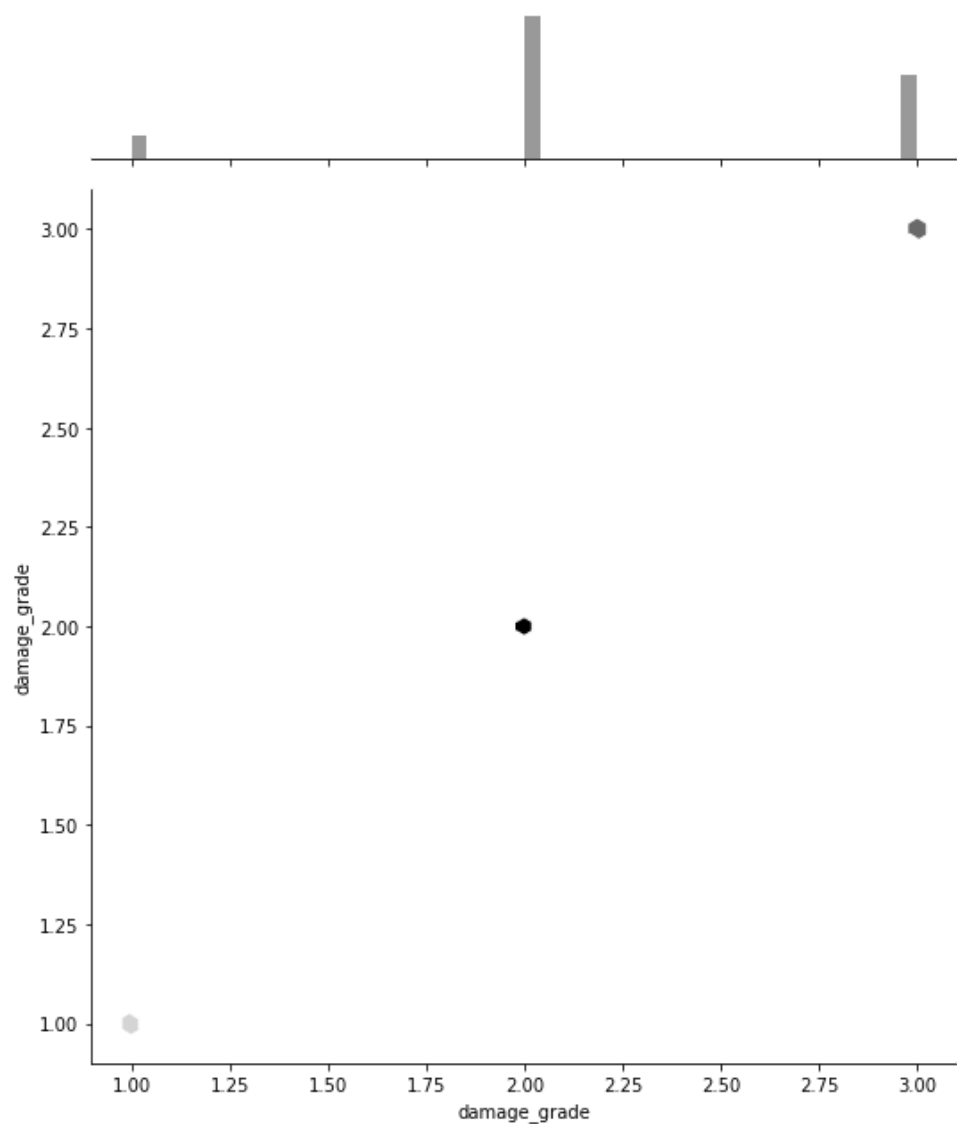












In [9]:

```
sns_data=train_values[object_cols].join(train_labels)
for col in sns_data.columns:
    sns.catplot(x=col, y="damage_grade", data=sns_data,kind="violin");
```

