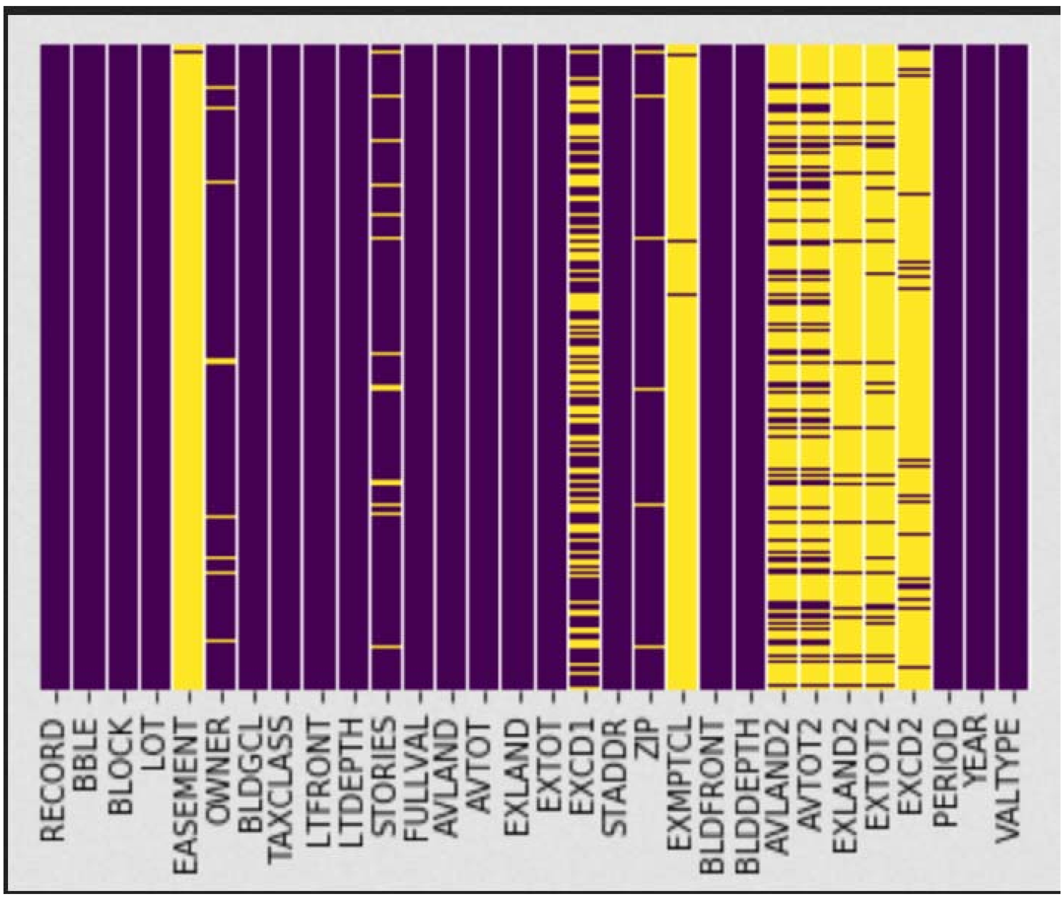
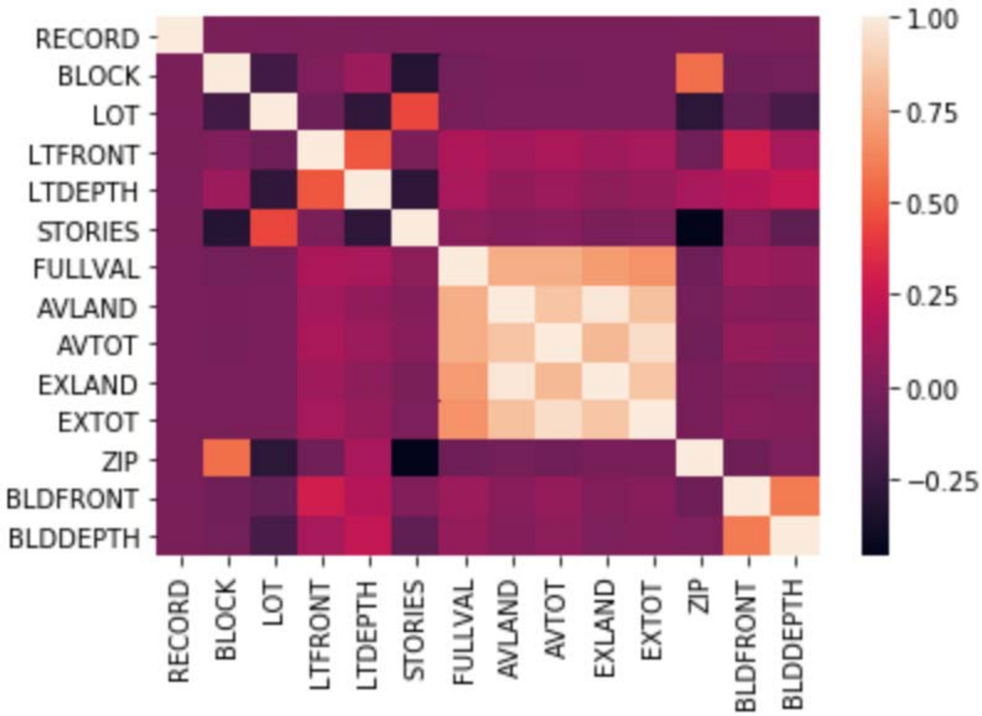
To:	Professor Stephen Coggeshall
From:	Alok Abhishek
Date:	01/24/2018
Subject:	DSO 562: Fraud Analytics

Before starting data analysis and visualization I created a heatmap for missing values. I then removed the columns from deeper analysis because so much data is missing that interpreting few entries does not help me in looking at the bigger picture. (I've added python code for data analysis and visualization at the end the assignment)

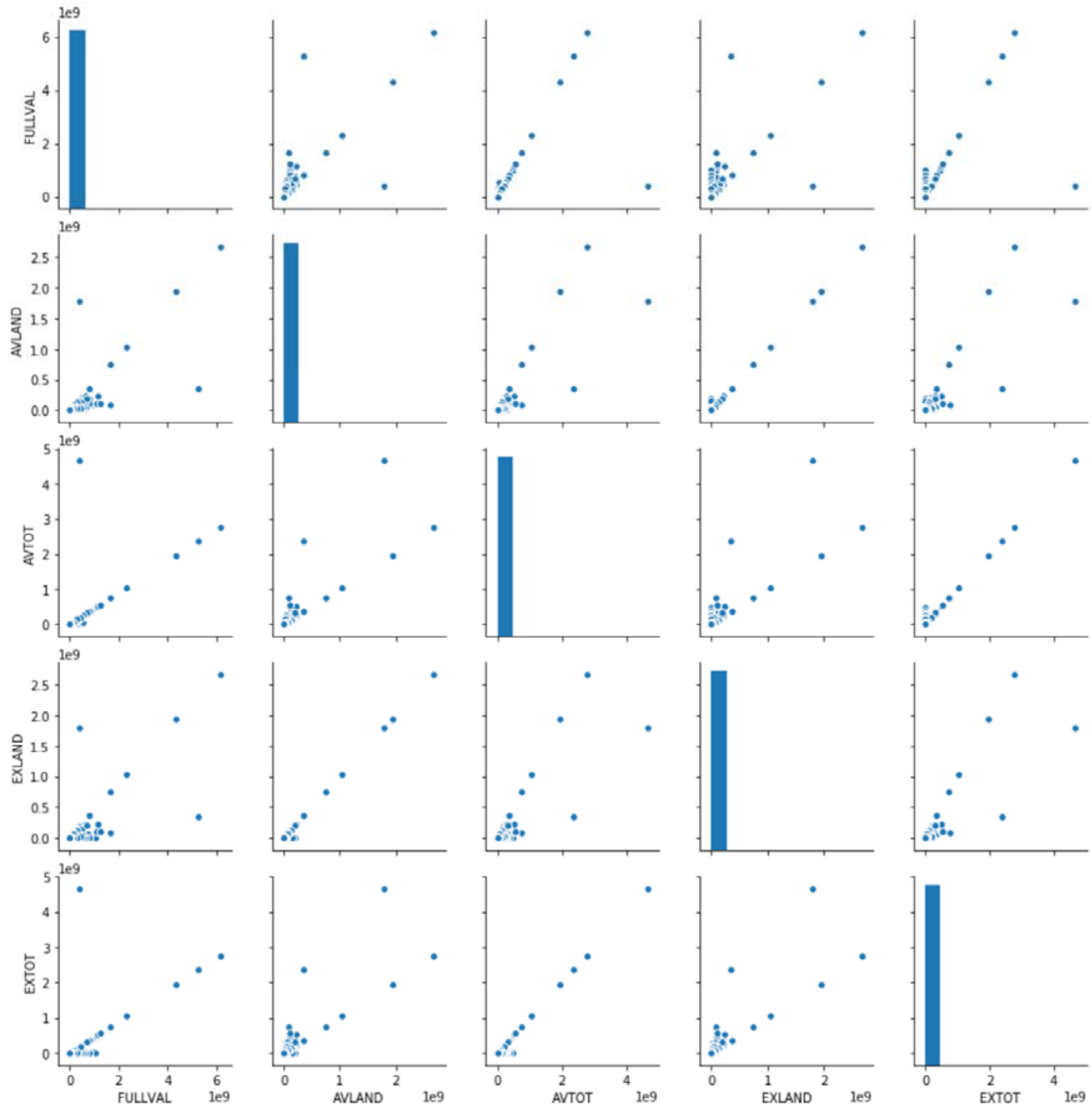


I then created heat map of correlation in between different variables to identify trends in data. I could use this relationship within data to identify outliers and potential fraud candidates.

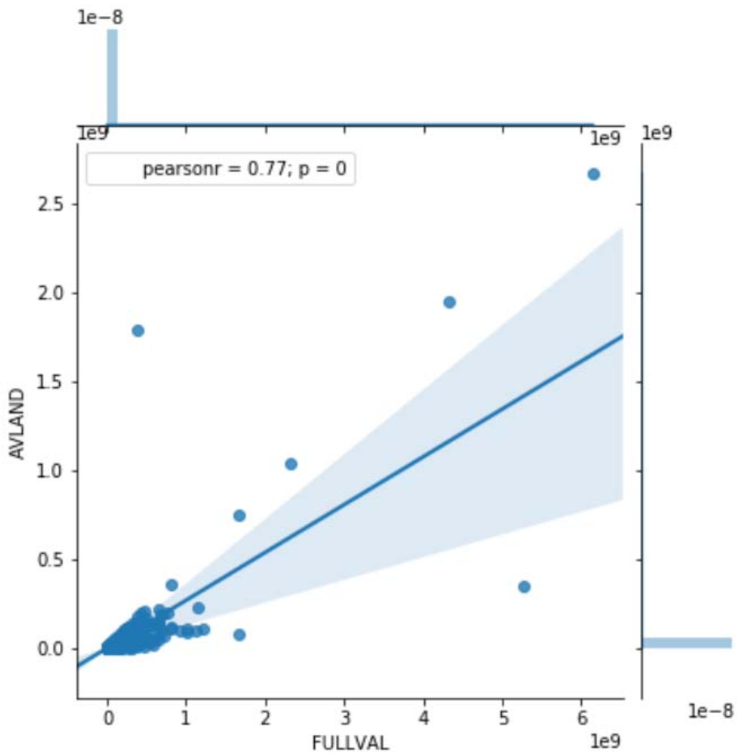


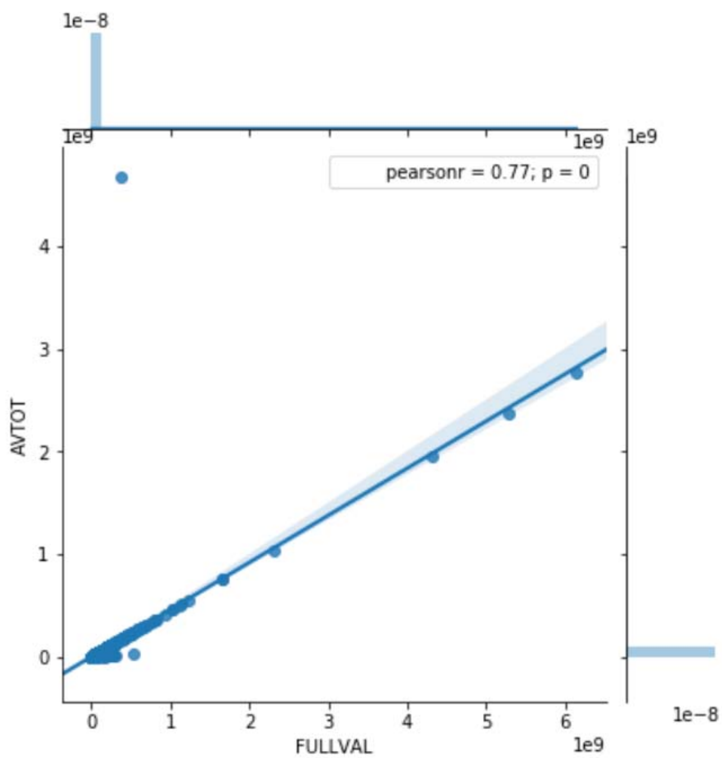
From the correlation plot we can see that there is high correlation in between Full Value, Average Land, Average Total, Ex Land, and Ex Total. I did a pair plot for these variables to identify the outliers.

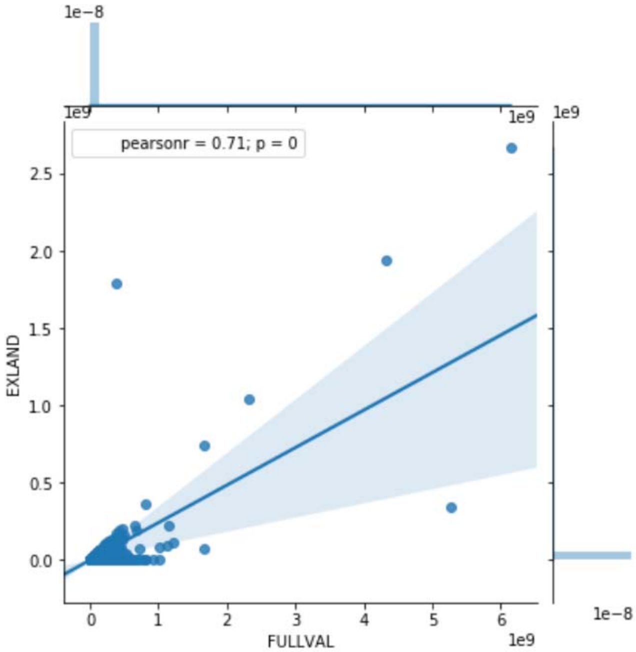
Most of the variable looks well aligned across the 45 degree slope with few outliers.

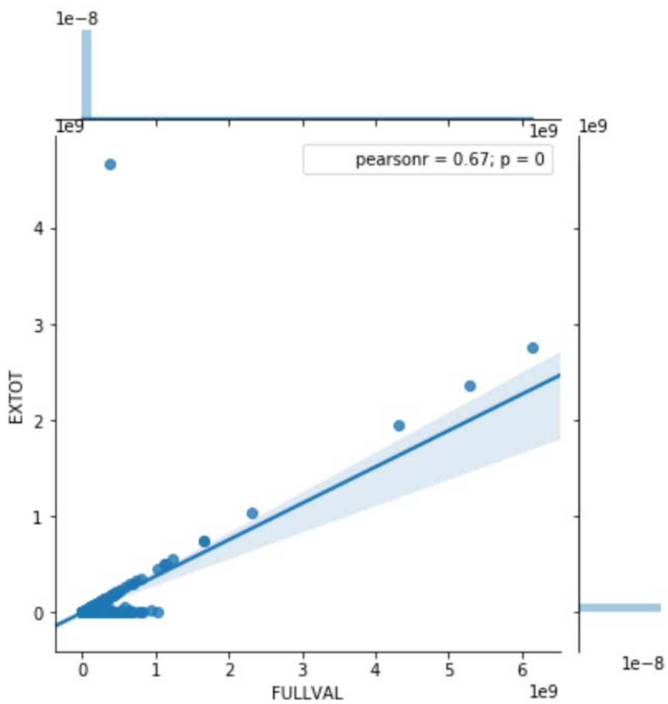


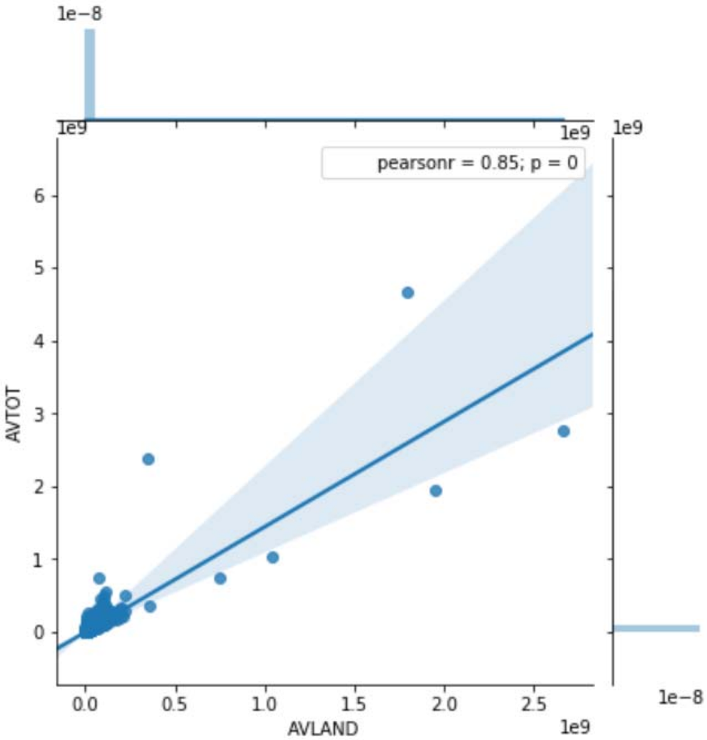
Following is more in depth look at some plots which has out of pattern data points. These data points which resides outside the shaded regression region are good candidate to look at for fraud. Some of these will have valid reason and some of them could be data entry error and some may be fraudulent data.


1e−8
1e9
1e9
1e9
pearsonr = 0.77; p = 0
2.5
2.0
1.5
1.0
0.5
0.0
AVLAND
0
1
2
3
4
5
6
FULLVAL
1e9
1e−8

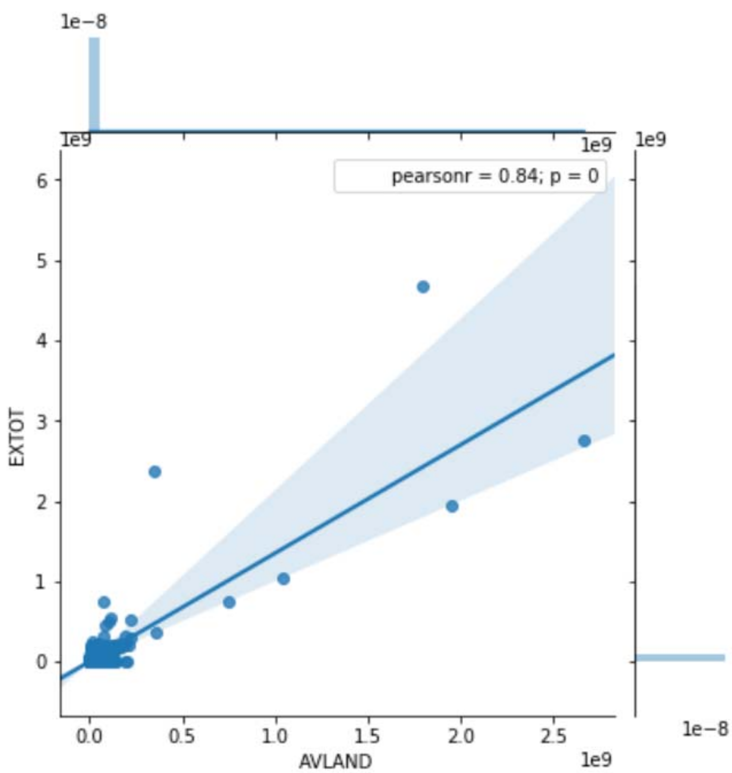

1e−8
1e9
1e9
1e9
pearsonr = 0.77; p = 0
4
3
2
1
0
AVTOT
0
1
2
3
4
5
6
FULLVAL
1e9
1e−8


1e−8
1e9
1e9
1e9
pearsonr = 0.71; p = 0
2.5
2.0
1.5
1.0
0.5
0.0
EXLAND
0
1
2
3
4
5
6
FULLVAL
1e9
1e−8


1e−8
1e9
1e9
1e9
pearsonr = 0.67; p = 0
4
3
2
1
0
EXTOT
0
1
2
3
4
5
6
FULLVAL
1e9
1e−8



1e−8
1e9
1e9
1e9
pearsonr = 0.85; p = 0
AVTOT
AVLAND
1e9
1e−8



1e−8
1e9
1e9
1e9
pearsonr = 0.84; p = 0
EXTOT
AVLAND
1e9
1e−8

While looking at the stories of buildings I noticed that there are several buildings with # of stories in decimals. This looks odd.

```
NYC_Property_data.groupby('STORIES').count()['RECORD']
```

```
STORIES
1.0      93606
1.1          3
1.2         33
1.3          3
1.4          2
1.5      24354
1.6       8816
1.7       5051
1.8         21
1.9         10
2.0     403318
2.1          1
2.2         40
2.3         19
2.4          2
2.5      81304
2.6        226
2.7      13543
2.8          3
2.9          1
3.0     128493
3.2         14
3.3          5
3.5       1188
3.6         11
3.7        251
4.0      38337
4.2          1
4.5        290
4.7         10
```

I also looked at lot front, lot depth, building front, and building depth. I noticed that there are a lot of properties with value zero for these fields which looks odd because how can building have 0 lot front or 0 lot depth.

```
NYC_Property_data_2.groupby('LTFRONT').count()['RECORD']
```

```
LTFRONT
0        168867
1           819
2           750
```

```
NYC_Property_data_2.groupby('LTDEPTH').count()['RECORD']
```

```
LTDEPTH
0        169888
1           126
2            79
```

```
NYC_Property_data_2.groupby('BLDFRONT').count()['RECORD']
```

```
BLDFRONT
0        224661
1            70
2            20
3            14
```

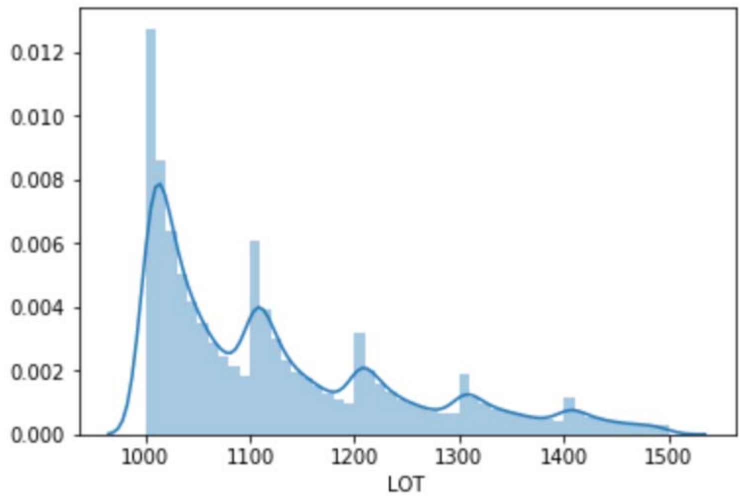
```
NYC_Property_data_2.groupby('BLDDEPTH').count()['RECORD']
```

```
BLDDEPTH
0        224699
1            52
2             9
3            90
4            60
```

Looking at the lot size in depth it seems like lot depth increases in increment of 100s.

```
tmp = NYC_Property_data[NYC_Property_data['LOT']<=1500]
tmp = tmp[tmp['LOT']>=1000]
sbrn.distplot(tmp['LOT'],bins=50)
```

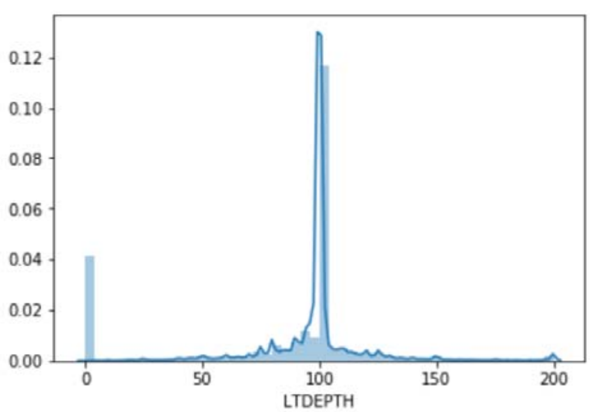
```
<matplotlib.axes._subplots.AxesSubplot at 0x1766618e470>
```



Looking at the lot depth it seems interesting that most of the lot depths are approximately 100.

```
sbrn.distplot(NYC_Property_data_2[NYC_Property_data_2['LTDEPTH']<=200]['LTDEPTH'],bins=50)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2108cd91940>
```



In [2]:
```
import numpy as np
import pandas as pd
import sklearn as sk
import seaborn as sbrn
import matplotlib.pyplot as plt
%matplotlib inline
```

In [3]:
```
NYC_Property_data = pd.read_csv('NY property 1 million.csv')
```

In [4]:
```
NYC_Property_data.describe()
```

Out[4]:

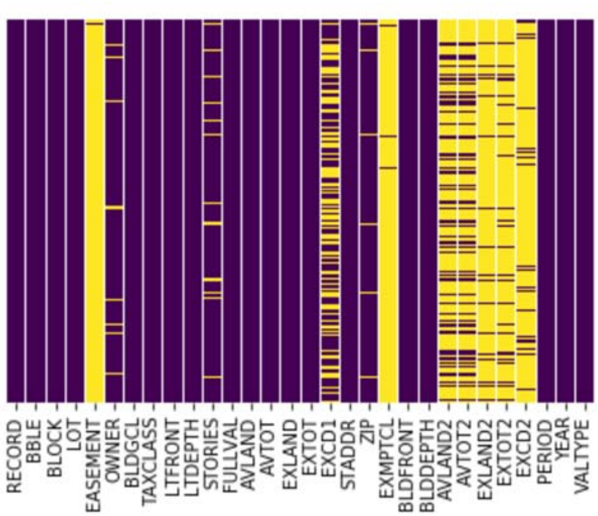
	RECORD	BLOCK	LOT	LTFRONT	LTDEPTH	STORIES	F
count	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	996433.000000	1.048
mean	5.242880e+05	4.708867e+03	3.700924e+02	3.617425e+01	8.827643e+01	5.063363	8.804
std	3.026977e+05	3.699547e+03	8.605382e+02	7.373356e+01	7.547885e+01	8.431372	1.170
min	1.000000e+00	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	1.000000	0.000
25%	2.621445e+05	1.534000e+03	2.300000e+01	1.900000e+01	8.000000e+01	2.000000	3.030
50%	5.242880e+05	3.944000e+03	4.900000e+01	2.500000e+01	1.000000e+02	2.000000	4.460
75%	7.864315e+05	6.797000e+03	1.460000e+02	4.000000e+01	1.000000e+02	3.000000	6.190
max	1.048575e+06	1.635000e+04	9.978000e+03	9.999000e+03	9.999000e+03	119.000000	6.150

In [5]:
```
NYC_Property_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 30 columns):
RECORD      1048575 non-null int64
BBLE        1048575 non-null object
BLOCK       1048575 non-null int64
LOT         1048575 non-null int64
EASEMENT    4043 non-null object
OWNER       1017492 non-null object
BLDGCL      1048575 non-null object
TAXCLASS    1048575 non-null object
LTFRONT     1048575 non-null int64
LTDEPTH     1048575 non-null int64
STORIES     996433 non-null float64
FULLVAL     1048575 non-null int64
AVLAND      1048575 non-null int64
AVTOT       1048575 non-null int64
EXLAND      1048575 non-null int64
EXTOT       1048575 non-null int64
EXCD1       622642 non-null float64
STADDR      1047934 non-null object
ZIP         1022219 non-null float64
EXMPTCL     14992 non-null object
BLDFRONT    1048575 non-null int64
BLDDEPTH    1048575 non-null int64
AVLAND2     280966 non-null float64
AVTOT2      280972 non-null float64
EXLAND2     86675 non-null float64
EXTOT2      129933 non-null float64
EXCD2       90941 non-null float64
PERIOD      1048575 non-null object
YEAR        1048575 non-null object
VALTYPE     1048575 non-null object
dtypes: float64(8), int64(12), object(10)
memory usage: 240.0+ MB
```

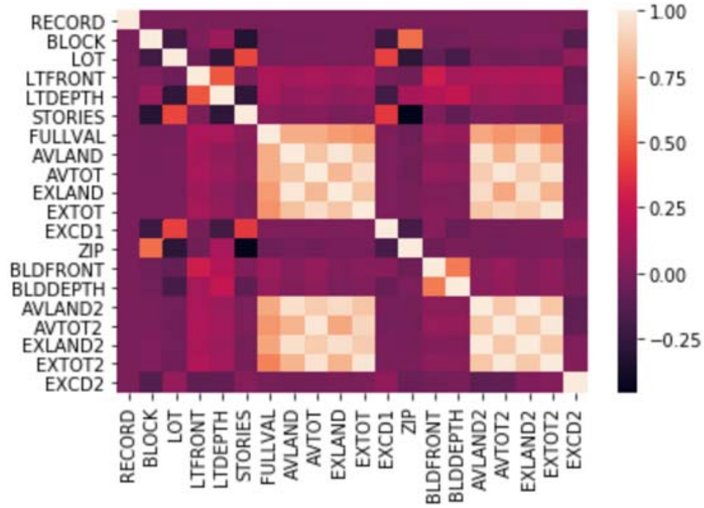
In [6]:
```
sbrn.heatmap(NYC_Property_data.isnull(),yticklabels=False,cbar=False,cmap='viridis'
)
```

Out[6]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21090ac4cf8>
```



In [10]:
```
sbrn.heatmap(NYC_Property_data.corr(),annot=False)
```

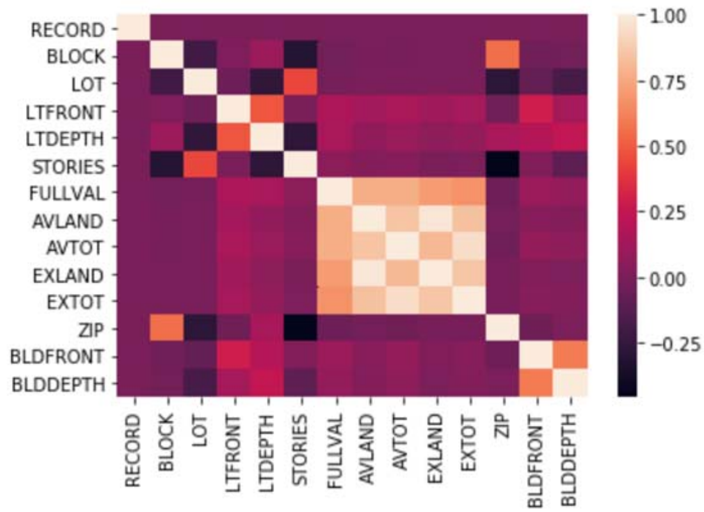
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x17613a56f98>



In [7]:
```
NYC_Property_data_2 = NYC_Property_data.drop(['EASEMENT', 'EXCD1','EXMPTCL','EXCD1'
,'AVLAND2', 'AVTOT2', 'EXLAND2', 'EXTOT2', 'EXCD2'], axis=1)
```

In [8]:
```
sbrn.heatmap(NYC_Property_data_2.corr(),annot=False)
```

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x2108d939390>

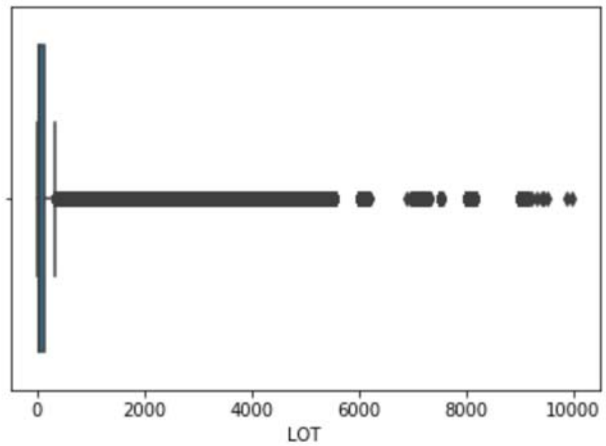


In [41]:
```
NYC_Property_data['LOT'].describe()
```

Out[41]:
```
count    1.048575e+06
mean     3.700924e+02
std      8.605382e+02
min      1.000000e+00
25%      2.300000e+01
50%      4.900000e+01
75%      1.460000e+02
max      9.978000e+03
Name: LOT, dtype: float64
```

In [42]:
```
sbrn.boxplot(NYC_Property_data['LOT'])
```

Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x176665884e0>

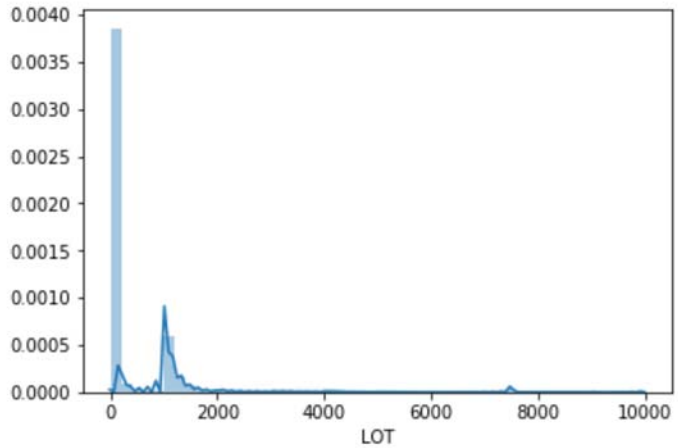


In [36]:
```
NYC_Property_data_2.groupby('LOT').count()['RECORD']
```

```
Out[36]: LOT
         1       23570
         2        6552
         3        9503
         4        8993
         5       10433
         6       11418
         7       11070
         8       10673
         9       10872
         10      10876
         11      10773
         12      11894
         13      11086
         14      11864
         15      11904
         16      11810
         17      11728
         18      11763
         19      11408
         20      12045
         21      11593
         22      11462
         23      11469
         24      11392
         25      11692
         26      11390
         27      11107
         28      11170
         29      11149
         30      11354
                 ...
         9102        2
         9103        1
         9104        1
         9105        1
         9106        1
         9107        1
         9108        1
         9109        1
         9110        2
         9111        1
         9112        1
         9113        1
         9114        1
         9115        1
         9116        1
         9117        1
         9121        3
         9130        1
         9132        1
         9134        1
         9150        1
         9172        1
         9220        1
         9300        1
         9401        1
         9421        1
         9450        1
         9502        1
         9878        1
         9978        1
         Name: RECORD, Length: 6366, dtype: int64
```

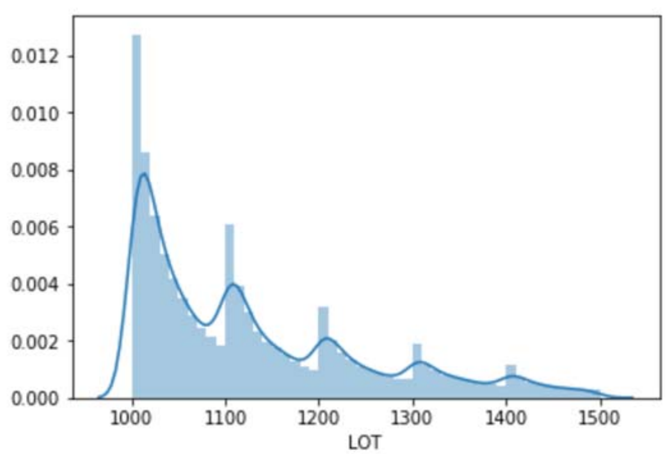
In [38]:
```
sbrn.distplot(NYC_Property_data['LOT'],bins=50)
```

Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x17665f71a20>



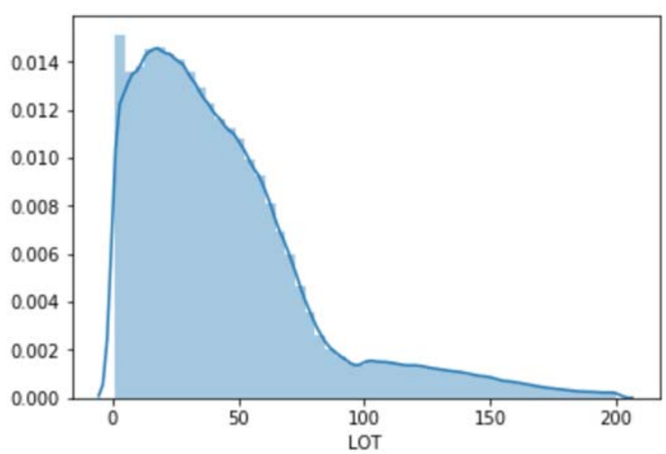
In [40]:
```
tmp = NYC_Property_data[NYC_Property_data['LOT']<=1500]
tmp = tmp[tmp['LOT']>=1000]
sbrn.distplot(tmp['LOT'],bins=50)
```

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x1766618e470>



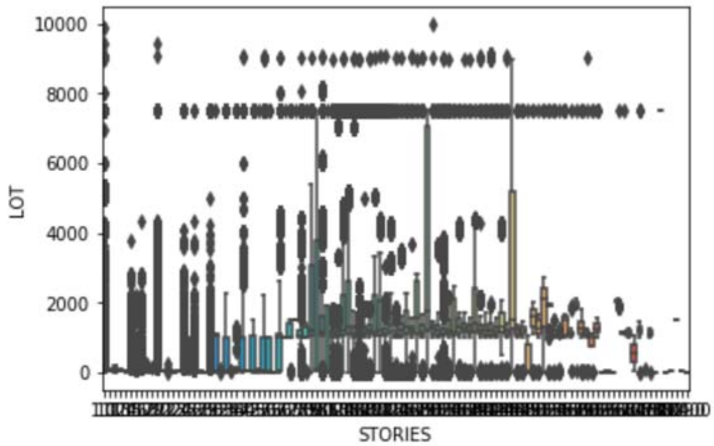
In [37]:
```
sbrn.distplot(NYC_Property_data[NYC_Property_data['LOT']<=200]['LOT'],bins=50,kde=True)
```

Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x17665fff438>



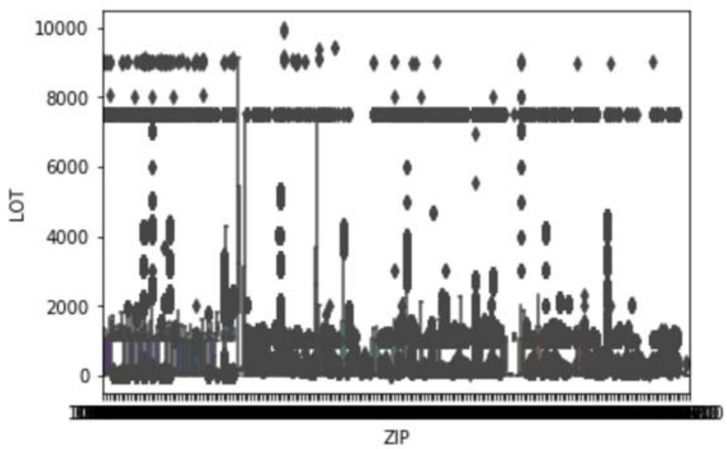
In [43]:
```
sbrn.boxplot(x='STORIES',y='LOT',data=NYC_Property_data,palette='rainbow')
```

Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x17666632400>



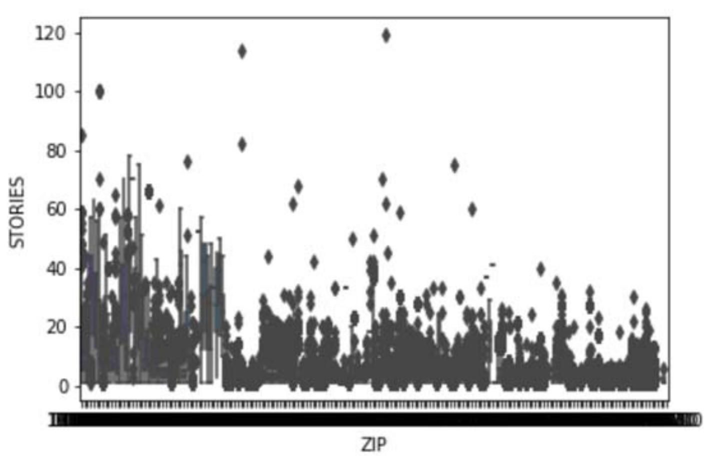
In [44]:
```
sbrn.boxplot(x='ZIP',y='LOT',data=NYC_Property_data,palette='rainbow')
```

Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x17615541320>



In [45]:
```
sbrn.boxplot(x='ZIP',y='STORIES',data=NYC_Property_data,palette='rainbow')
```

Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x176157eaba8>

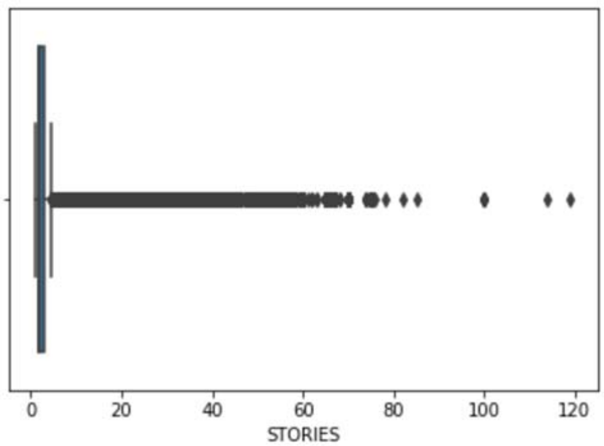


In [50]:
```
NYC_Property_data['STORIES'].describe()
```

Out[50]:
```
count    996433.000000
mean          5.063363
std           8.431372
min           1.000000
25%           2.000000
50%           2.000000
75%           3.000000
max         119.000000
Name: STORIES, dtype: float64
```

In [51]:
```
sbrn.boxplot(NYC_Property_data['STORIES'])
```

Out[51]: <matplotlib.axes._subplots.AxesSubplot at 0x1761bdeb4e0>



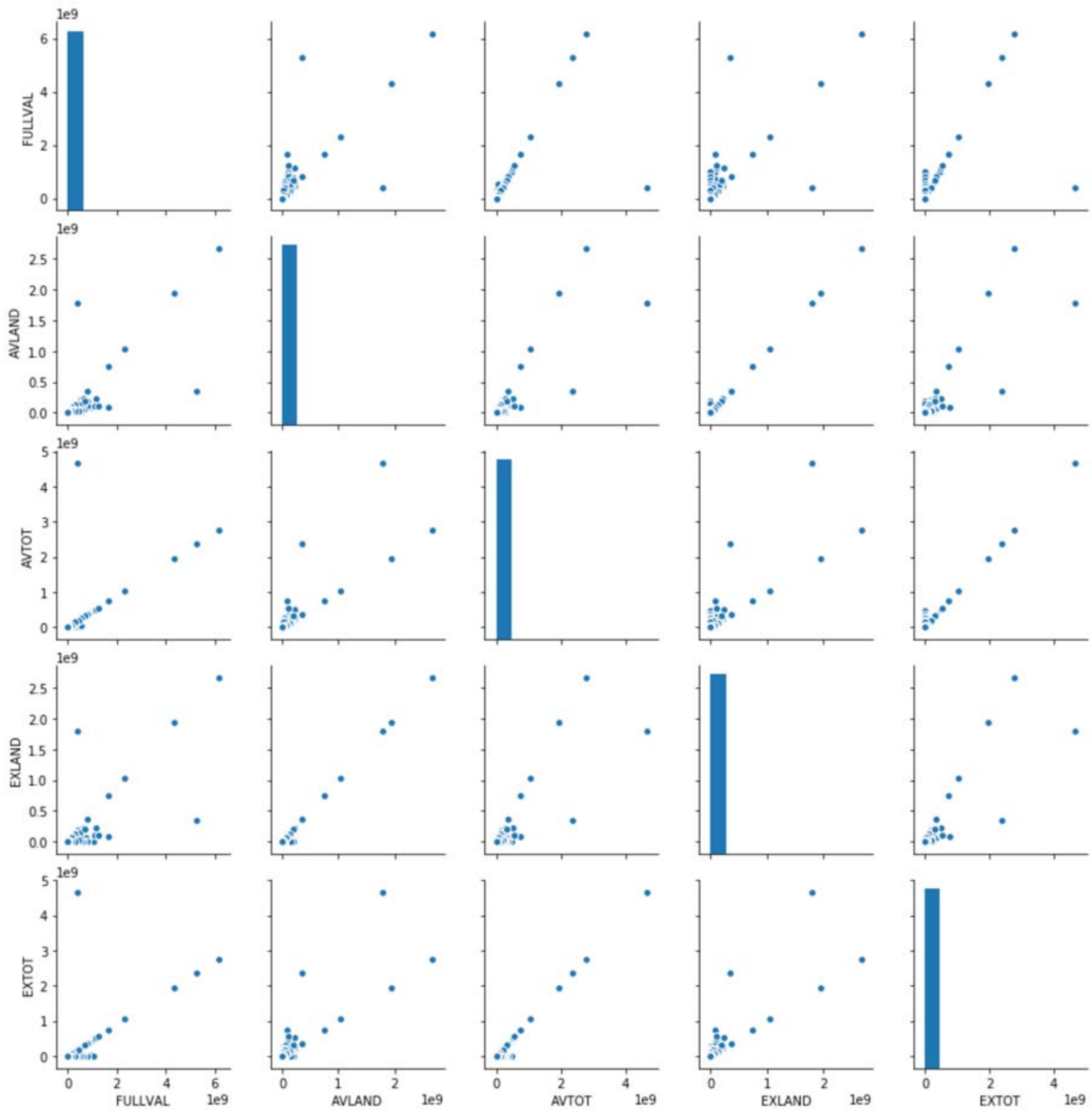
In [58]:
```
NYC_Property_data.groupby('STORIES').count()['RECORD']
```

```
Out[58]: STORIES
         1.0       93606
         1.1           3
         1.2          33
         1.3           3
         1.4           2
         1.5       24354
         1.6        8816
         1.7        5051
         1.8          21
         1.9          10
         2.0      403318
         2.1           1
         2.2          40
         2.3          19
         2.4           2
         2.5       81304
         2.6         226
         2.7       13543
         2.8           3
         2.9           1
         3.0      128493
         3.2          14
         3.3           5
         3.5        1188
         3.6          11
         3.7         251
         4.0       38337
         4.2           1
         4.5         290
         4.7          10
                   ...
         48.0        861
         49.0        472
         50.0       1214
         51.0        103
         52.0        344
         53.0         64
         54.0        366
         55.0        380
         56.0        226
         57.0       1445
         58.0        253
         59.0         12
         60.0        561
         61.0          1
         62.0          3
         63.0          2
         65.0         72
         66.0         66
         67.0        242
         68.0          2
         70.0        849
         74.0          6
         75.0         31
         76.0          1
         78.0          1
         82.0          1
         85.0          1
         100.0         5
         114.0         1
         119.0         1
         Name: RECORD, Length: 111, dtype: int64
```

In [9]:
```
NYC_Property_data_3 = NYC_Property_data_2[['FULLVAL','AVLAND','AVTOT','EXLAND','EXT
OT']]

sbrn.pairplot(NYC_Property_data_3)
```

Out[9]: <seaborn.axisgrid.PairGrid at 0x2108c0d6438>

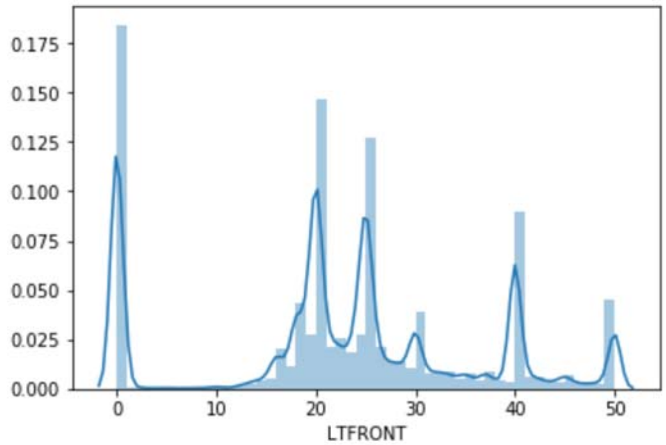


In [14]:
```
NYC_Property_data_2.groupby('LTFRONT').count()['RECORD']
```

```
Out[14]: LTFRONT
         0       168867
         1          819
         2          750
         3          304
         4          269
         5          505
         6          231
         7          270
         8          363
         9          403
         10        1139
         11         294
         12        1172
         13        2327
         14        4027
         15        4864
         16       18359
         17       10372
         18       40188
         19       25185
         20      134447
         21       19319
         22       23304
         23       16801
         24       25180
         25      116301
         26       19415
         27       12485
         28       12963
         29        9249
                  ...
         4129         1
         4152         1
         4171         1
         4300         1
         4318         1
         4507         1
         4644         1
         4646         1
         4775         1
         4824         1
         4910         1
         4989         1
         5262         1
         5370         1
         5380         1
         5400         1
         5425         1
         5878         1
         6078         1
         6317         1
         6500         1
         7536         1
         7653         1
         8000         2
         8715         2
         8744         1
         8821         2
         9170         1
         9742         1
         9999         3
         Name: RECORD, Length: 1277, dtype: int64
```

In [19]:
```
sbrn.distplot(NYC_Property_data_2[NYC_Property_data_2['LTFRONT']<=50]['LTFRONT'],bi
ns=50)
```

Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x2108cc56ba8>

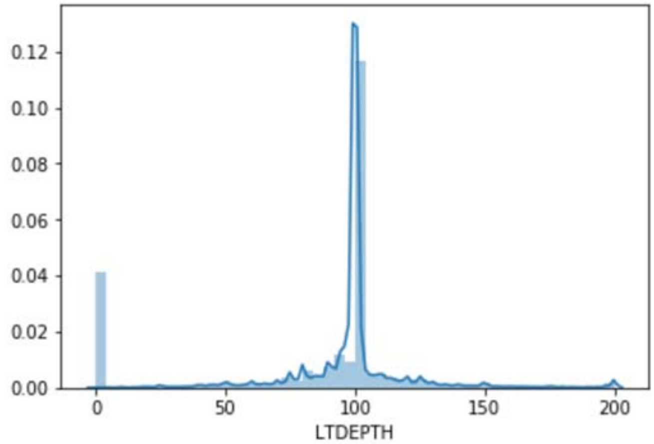


In [16]:
```
NYC_Property_data_2.groupby('LTDEPTH').count()['RECORD']
```

```
Out[16]: LTDEPTH
         0       169888
         1          126
         2           79
         3           81
         4           85
         5          180
         6           64
         7           89
         8           75
         9           79
         10         547
         11          73
         12         100
         13          82
         14          78
         15         268
         16         145
         17         160
         18         451
         19         154
         20         656
         21         201
         22         259
         23         187
         24         305
         25        1881
         26         291
         27         306
         28         233
         29         220
                  ...
         3700         2
         3756         1
         3900         1
         4000         2
         4050         2
         4056         1
         4356         1
         4463         1
         4471         1
         4500         1
         4563         1
         4720         1
         4770         1
         4900         1
         4934         1
         5000         2
         5100         1
         5143         1
         5360         2
         5463         1
         5853         1
         5948         1
         6074         1
         6400         1
         7055         1
         7960         1
         8000         1
         8847         1
         9619         1
         9999         1
         Name: RECORD, Length: 1336, dtype: int64
```

In [20]:
```
sbrn.distplot(NYC_Property_data_2[NYC_Property_data_2['LTDEPTH']<=200]['LTDEPTH'],b
ins=50)
```

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x2108cd91940>

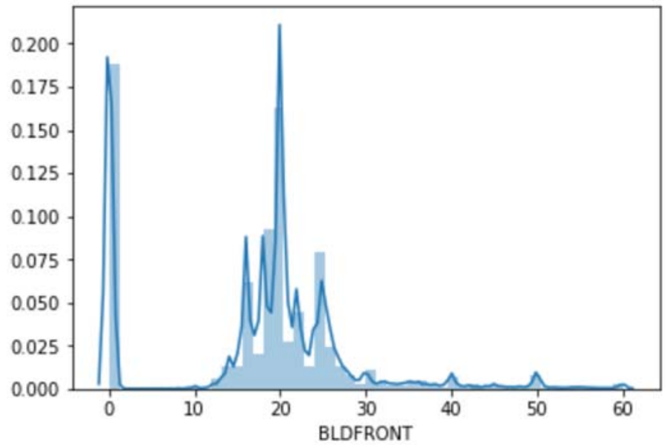


In [17]:
```
NYC_Property_data_2.groupby('BLDFRONT').count()['RECORD']
```

```
Out[17]: BLDFRONT
         0       224661
         1           70
         2           20
         3           14
         4           14
         5           45
         6           33
         7           23
         8          209
         9          197
         10        1385
         11         143
         12        2076
         13        4854
         14       15792
         15       16013
         16       73671
         17       23821
         18       76808
         19       33073
         20      193812
         21       32593
         22       53227
         23       16036
         24       32472
         25       61770
         26       28443
         27       15112
         28        9035
         29        3841
                  ...
         900          1
         911          1
         961          1
         982          1
         1000         2
         1102         1
         1160         1
         1169         1
         1225         1
         1227         1
         1280         1
         1362         1
         1394         1
         1812         1
         1844         2
         1925         1
         1943         1
         2025         1
         2030         1
         2500         1
         3100         1
         3285         1
         4017         1
         4149         1
         5518         1
         5614         1
         6020         1
         6414         1
         7538         1
         7575         1
         Name: RECORD, Length: 610, dtype: int64
```

In [21]:
```
sbrn.distplot(NYC_Property_data_2[NYC_Property_data_2['BLDFRONT']<=60]['BLDFRONT'],
bins=50)
```

Out[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2108ce11ba8>
```

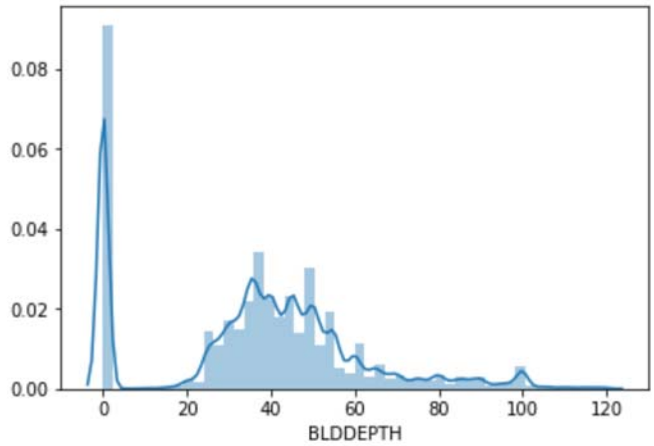


In [18]:
```
NYC_Property_data_2.groupby('BLDDEPTH').count()['RECORD']
```

```
Out[18]: BLDDEPTH
         0       224699
         1           52
         2            9
         3           90
         4           60
         5           51
         6           48
         7           13
         8           53
         9           14
         10         535
         11          22
         12         174
         13          54
         14         133
         15         673
         16         416
         17         213
         18        1345
         19         339
         20        4305
         21         896
         22        3020
         23        1459
         24        9892
         25       11491
         26       14512
         27        7243
         28       19709
         29        4605
                  ...
         980          3
         992          2
         999          2
         1000         2
         1007         1
         1075         1
         1131         1
         1150         1
         1175         1
         1222         1
         1300         2
         1375         1
         1399         1
         1971         1
         1980         1
         2023         1
         2436         1
         3104         1
         3390         1
         4500         1
         4600         1
         5000         1
         5020         1
         5600         1
         5641         1
         6308         1
         7360         1
         8500         1
         9388         1
         9393         1
         Name: RECORD, Length: 620, dtype: int64
```

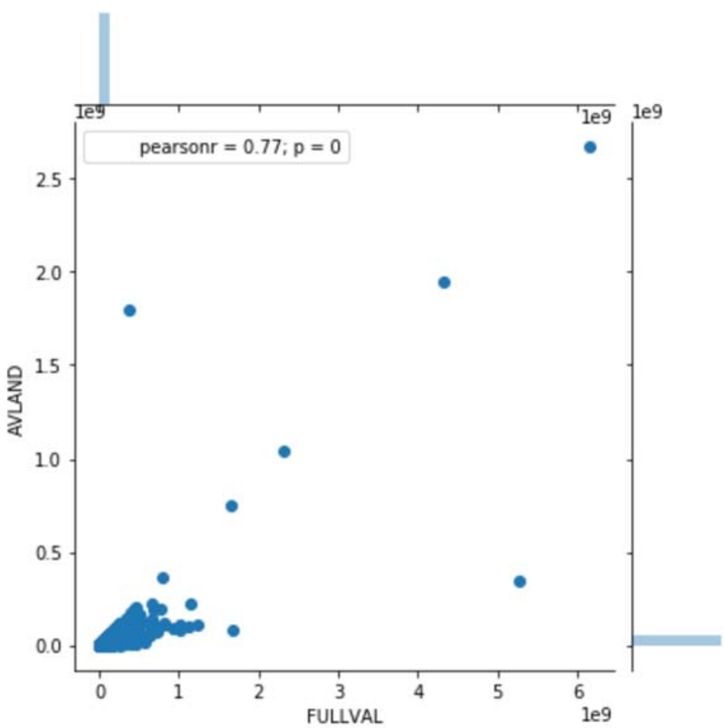
In [22]:
```
sbrn.distplot(NYC_Property_data_2[NYC_Property_data_2['BLDDEPTH']<=120]['BLDDEPTH']
,bins=50)
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x2108e21ee48>



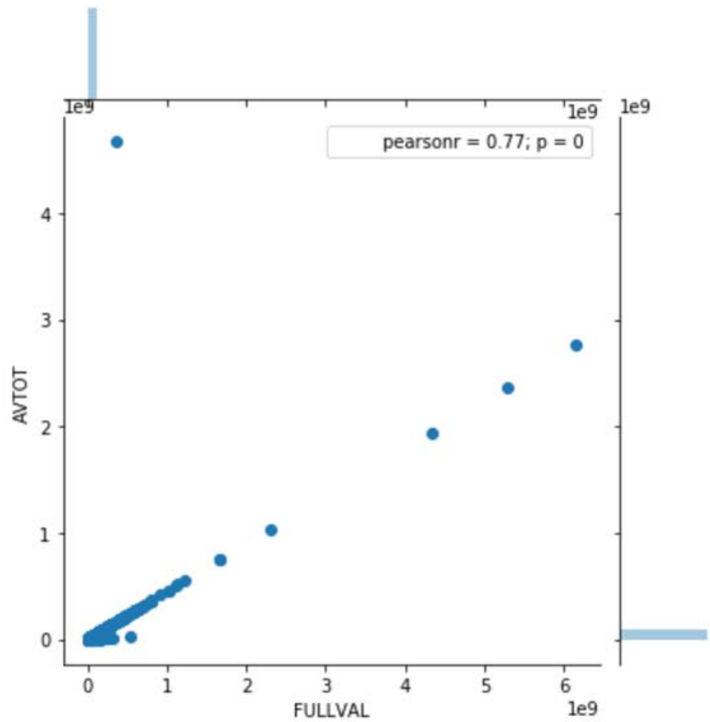
In [25]:
```
sbrn.jointplot(x='FULLVAL',y='AVLAND',data=NYC_Property_data_3)
```

Out[25]: <seaborn.axisgrid.JointGrid at 0x2108e483b00>



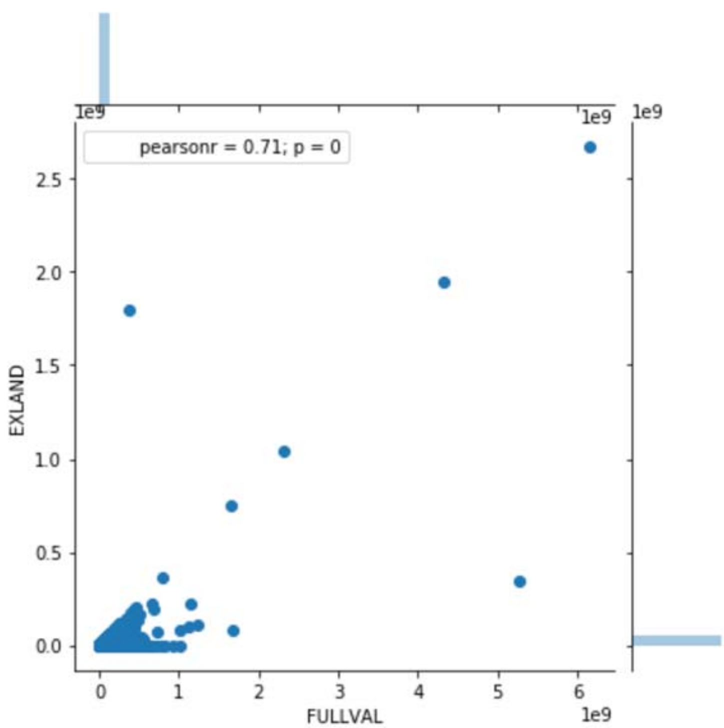
In [26]:
```
sbrn.jointplot(x='FULLVAL',y='AVTOT',data=NYC_Property_data_3)
```

Out[26]: <seaborn.axisgrid.JointGrid at 0x2108e4961d0>



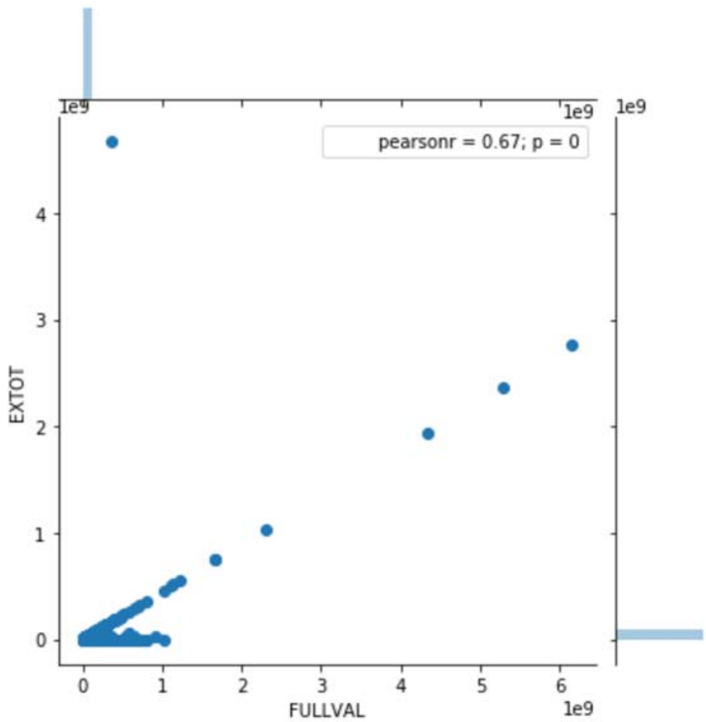
In [27]:
```
sbrn.jointplot(x='FULLVAL',y='EXLAND',data=NYC_Property_data_3)
```

Out[27]: <seaborn.axisgrid.JointGrid at 0x2108e645898>



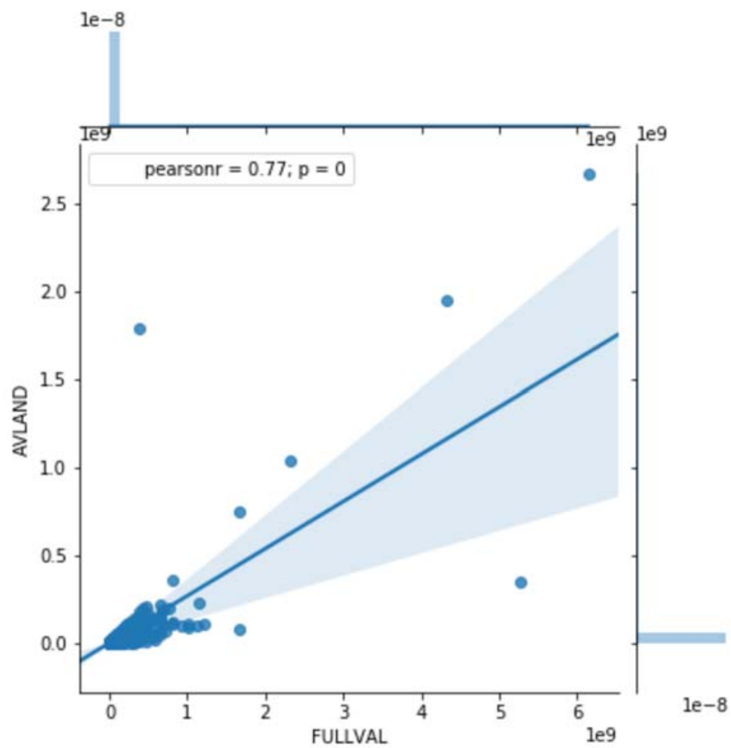
In [28]:
```
sbrn.jointplot(x='FULLVAL',y='EXTOT',data=NYC_Property_data_3)
```

Out[28]: <seaborn.axisgrid.JointGrid at 0x2108e84f208>



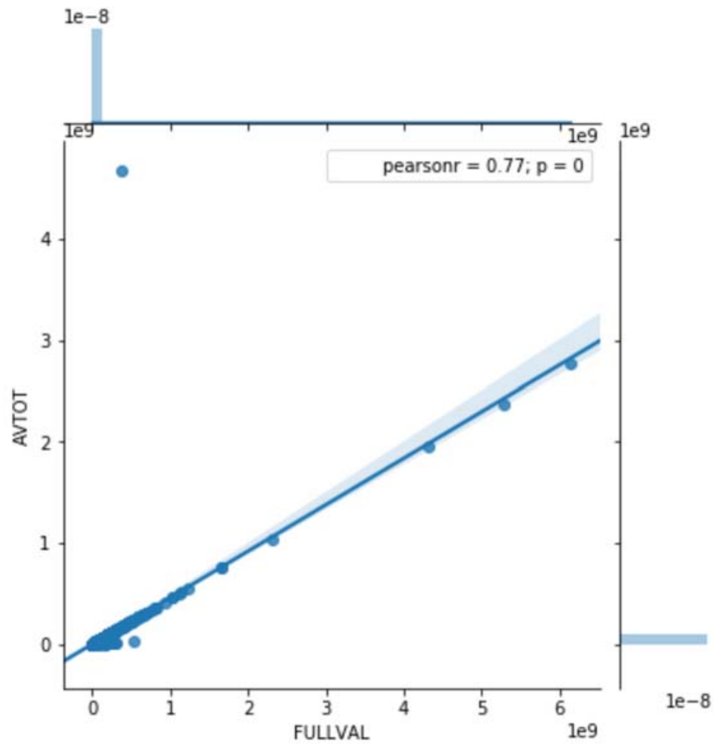
In [29]:
```
sbrn.jointplot("FULLVAL", "AVLAND", data=NYC_Property_data_3, kind="reg")
```

Out[29]: <seaborn.axisgrid.JointGrid at 0x2108eb0fe48>



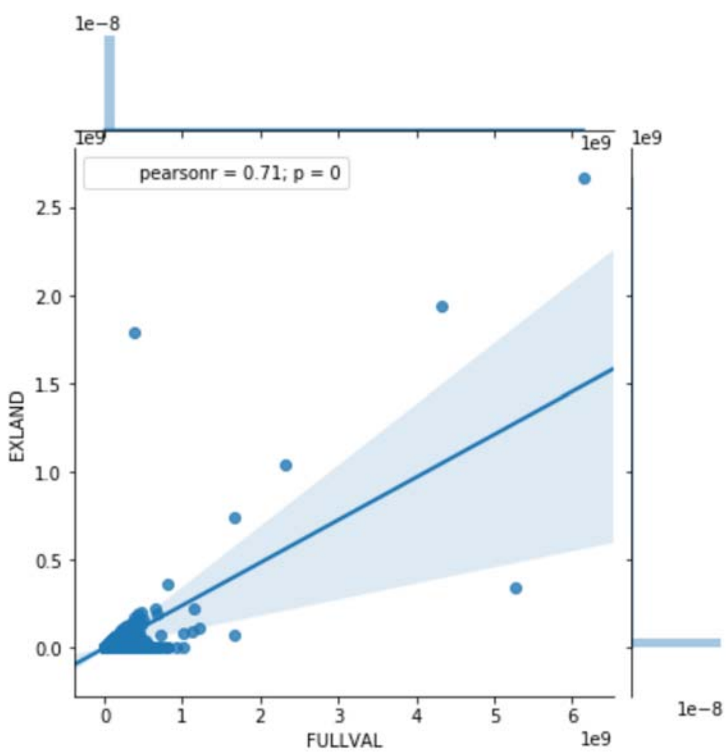
In [30]:
```
sbrn.jointplot("FULLVAL", "AVTOT", data=NYC_Property_data_3, kind="reg")
```

Out[30]: <seaborn.axisgrid.JointGrid at 0x2108eb0f3c8>



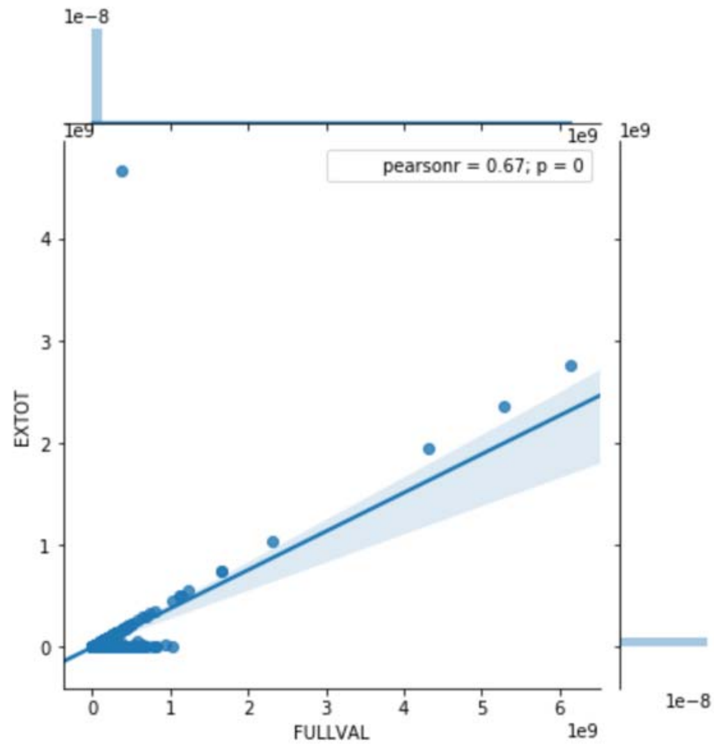
In [31]:
```
sbrn.jointplot("FULLVAL", "EXLAND", data=NYC_Property_data_3, kind="reg")
```

Out[31]: <seaborn.axisgrid.JointGrid at 0x2108ff3d710>



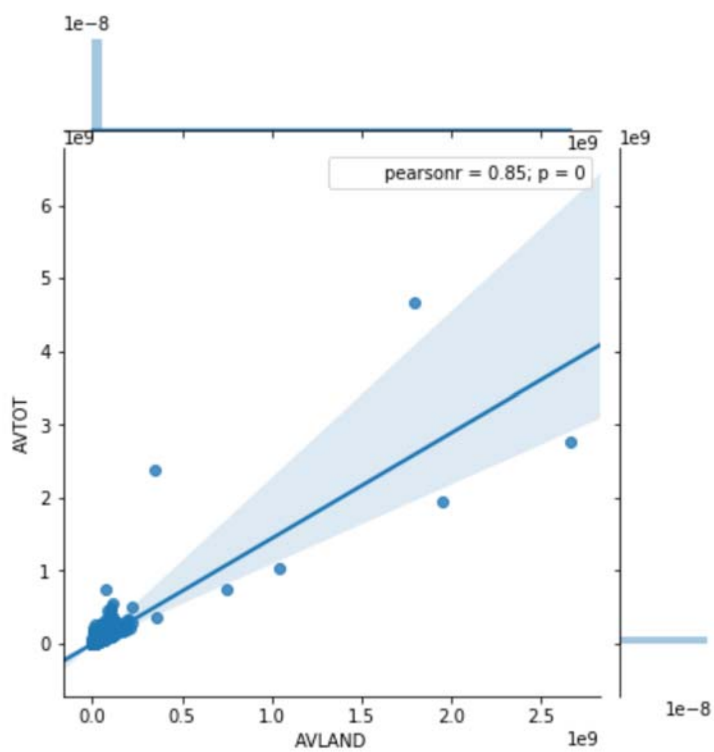
In [32]:
```
sbrn.jointplot("FULLVAL", "EXTOT", data=NYC_Property_data_3, kind="reg")
```

Out[32]: <seaborn.axisgrid.JointGrid at 0x21090a532b0>



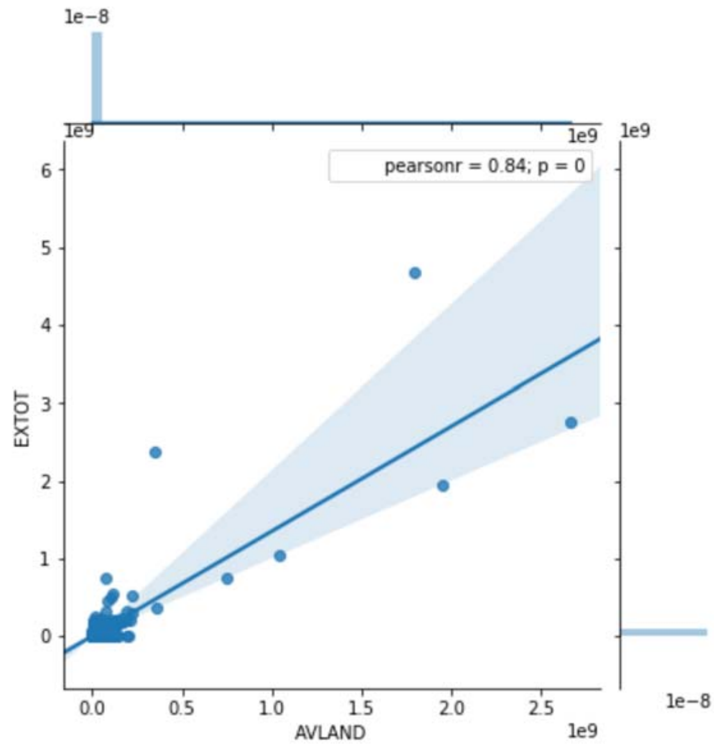
In [37]:
```
sbrn.jointplot("AVLAND", "AVTOT", data=NYC_Property_data_3, kind="reg")
```

Out[37]: <seaborn.axisgrid.JointGrid at 0x210a02e2978>



In [38]:
```
sbrn.jointplot("AVLAND", "EXTOT", data=NYC_Property_data_3, kind="reg")
```

Out[38]: <seaborn.axisgrid.JointGrid at 0x210a2a08da0>



In [39]:
```
sbrn.jointplot("AVLAND", "EXLAND", data=NYC_Property_data_3, kind="reg")
```

Out[39]: <seaborn.axisgrid.JointGrid at 0x210a2a33898>