In [40]:
```python
# Put the ARN of your AWS Secrets Manager secret for your redshift cluster here:
secret_name="arn:aws:secretsmanager:us-west-1:603574119146:secret:secretkey-project-EQOP4N"
# This will get the secret from AWS Secrets Manager.
import boto3
import json
session = boto3.session.Session()
client = session.client(
    service_name='secretsmanager'
)
get_secret_value_response = client.get_secret_value(
    SecretId=secret_name
)
if 'SecretString' in get_secret_value_response:
    connection_info = json.loads(get_secret_value_response['SecretString'])
else:
    print("ERROR: no secret data found")
```

In [41]:
```python
expected_keys = set(['dbname','host','dbClusterIdentifier','password', 'port', 'engine', 'username'])
if not expected_keys.issubset(connection_info.keys()):
    print("Expected values for ",expected_keys)
    print("Received values for ",set(connection_info.keys()))
    print("Please adjust query or assignment as required!")
```

In [42]:
```python
import time
import psycopg2
con=psycopg2.connect(
    dbname   = connection_info["dbname"],
    host     = connection_info["host"],
    port     = connection_info["port"],
    user     = connection_info["username"],
    password = connection_info["password"]
)
```

In [43]:
```python
cur = con.cursor()
cur.execute("select * from usa_housing_population limit 5")
res = cur.fetchall()
print(res)
cur.close()
```

```
[('cleveland', 'OH', '11/1/17', 142952, 44102, 191450, 0.161, 1.7389, 133, -0.0764, 0.031, None, 0.163265306, 0.17931
0345, 24, -0.25, -0.1429, 4, None, None, 20, -0.375, -0.1667, 65, 0.1017, 3.3333, 115.2431818, 0.1815, 1.5241, 1628,
0.0317, 0.1922, 284058.5734, 0.0704, 1.0009, 198, -0.0246, 0.375, 0.488721805), ('greenville', 'NC', '11/1/17', 13224
1, 27834, 142400, -0.0659, 0.1005, 126, -0.0455, -0.2921, 88, 0.257142857, -0.293172691, 24, -0.1429, -0.25, 4, None,
None, 20, 0.6667, -0.375, 90, -0.0816, 0.1688, 86.73016101, -0.0537, 0.0581, 1604, -0.0795, 0.0409, 203423.6565, -0.0
572, 0.0966, 216, -0.0609, -0.1529, 0.714285714), ('tampa', 'FL', '11/1/17', 201871, 33616, 259900, 0.0, 0.0554, 73,
-0.0135, -0.0519, 69, -0.086092715, 0.568181818, 28, 0.1667, 0.0, 4, 0.0, None, 20, -0.5455, -0.1667, 45, -0.1346, 0.
25, 172.6422387, -0.0024, 0.0661, None, -0.0245, 0.058, 323086.6429, 0.0136, 0.0115, 118, -0.0635, 0.0442, 0.61643835
6), ('miami', 'FL', '11/1/17', 278083, 33185, 439500, 0.0163, 0.0222, 84, 0.0244, -0.1429, 62, -0.107913669, -0.21019
1083, 28, -0.125, -0.3, 4, 0.0, 0.0, 20, 0.25, 0.6667, 61, -0.0615, 0.0, 189.8023434, 0.0189, 0.0895, None, 0.0905, -
0.0654, 432623.6469, 0.0214, 0.0107, 145, -0.0136, -0.0881, 0.726190476), ('denver', 'CO', '11/1/17', 408911, 80231,
427250, 0.5399, -0.1453, 48, -0.25, -0.0943, 36, 0.2, -0.152941176, 32, -0.3846, 0.1429, 4, None, None, 20, -0.2857,
0.0, 1, -0.5, 0.0, 172.9302533, 0.0053, 0.0435, 2642, 0.5783, -0.0833, 618434.2169, 0.1667, -0.1487, 49, -0.2576, -0.
0926, 0.020833333)]
```

In [44]:
```python
import pandas as pd
import pandas.io.sql as sqlio
import psycopg2
import pandas as pd
import numpy as np
import seaborn as sns
import datetime
import matplotlib.pyplot as plt
%matplotlib inline
```

In [45]:
```python
sql = "select * from usa_housing_population;"
housing_population= sqlio.read_sql_query(sql, con)
```

In [46]:
```python
housing_population
```

Out[46]:

	city	state	date	population	postal_code	median_listing_price	median_listing_price_mm	median_listing_price_yy	active_listing_count	active_l
0	cleveland	OH	11/1/17	142952	44102	191450.0	0.1610	1.7389	133	
1	greenville	NC	11/1/17	132241	27834	142400.0	-0.0659	0.1005	126	
2	tampa	FL	11/1/17	201871	33616	259900.0	0.0000	0.0554	73	
3	miami	FL	11/1/17	278083	33185	439500.0	0.0163	0.0222	84	
4	denver	CO	11/1/17	408911	80231	427250.0	0.5399	-0.1453	48	
...	...	...	...	...	...	...	...	...	...	
214084	fort myers	FL	7/1/21	305212	33913	489900.0	0.1134	NaN	63	
214085	new york	NY	7/1/21	554085	10065	1925000.0	0.0405	0.0280	443	
214086	sioux falls	SD	7/1/21	261064	57106	265000.0	0.0004	NaN	190	
214087	cullman	AL	7/1/21	156254	35058	229000.0	-0.1820	0.1456	27	
214088	muncie	IN	7/1/21	116519	47303	155000.0	0.3784	NaN	41	

214089 rows × 39 columns

In [47]:
```python
housing_population.describe()
```

Out[47]:

	population	postal_code	median_listing_price	median_listing_price_mm	median_listing_price_yy	active_listing_count	active_listing_count_mm	activ
count	2.140890e+05	214089.000000	1.989090e+05	171010.000000	156691.000000	214089.000000	171010.000000	
mean	2.354718e+05	54865.608761	3.262474e+05	0.008634	0.069869	91.396980	0.002834	
std	1.545481e+05	26330.251459	3.530873e+05	0.102221	0.170113	91.231932	0.232148	
min	3.866300e+04	1104.000000	2.500000e+03	-0.960900	-0.929200	13.000000	-0.727300	
25%	1.457480e+05	32966.000000	1.500000e+05	-0.027600	-0.020800	36.000000	-0.085500	
50%	1.967090e+05	52246.000000	2.390000e+05	0.000000	0.055600	65.000000	-0.009600	
75%	2.670410e+05	78260.000000	3.750000e+05	0.035800	0.144300	115.000000	0.068700	
max	1.455985e+06	99950.000000	6.622500e+06	14.018800	7.410900	1536.000000	68.000000	

8 rows × 36 columns

In [48]:
```python
housing_population= housing_population.sort_values(['state', 'date'], ascending = (True, True)).reset_index(drop = T
rue)
```

In [49]:
```python
housing_population['median_square_feet_yy'].mean()
housing_population['average_listing_price'].mean()
```

Out[49]: 422409.51317344076

In [50]:
```python
housing_population[['average_listing_price','median_square_feet_yy','median_listing_price_mm','median_listing_price_
yy']] = housing_population[['average_listing_price','median_square_feet_yy','median_listing_price_mm','median_listin
g_price_yy']].fillna(value=housing_population[['average_listing_price','median_square_feet_yy','median_listing_price
_mm','median_listing_price_yy']].mean())
```

In [51]:
```python
housing_population[['average_listing_price_mm','average_listing_price_yy','total_listing_count_mm','total_listing_co
unt_yy']] = housing_population[['average_listing_price_mm','average_listing_price_yy','total_listing_count_mm','tota
l_listing_count_yy']].fillna(value=housing_population[['average_listing_price_mm','average_listing_price_yy','total_
listing_count_mm','total_listing_count_yy']].mean())
```

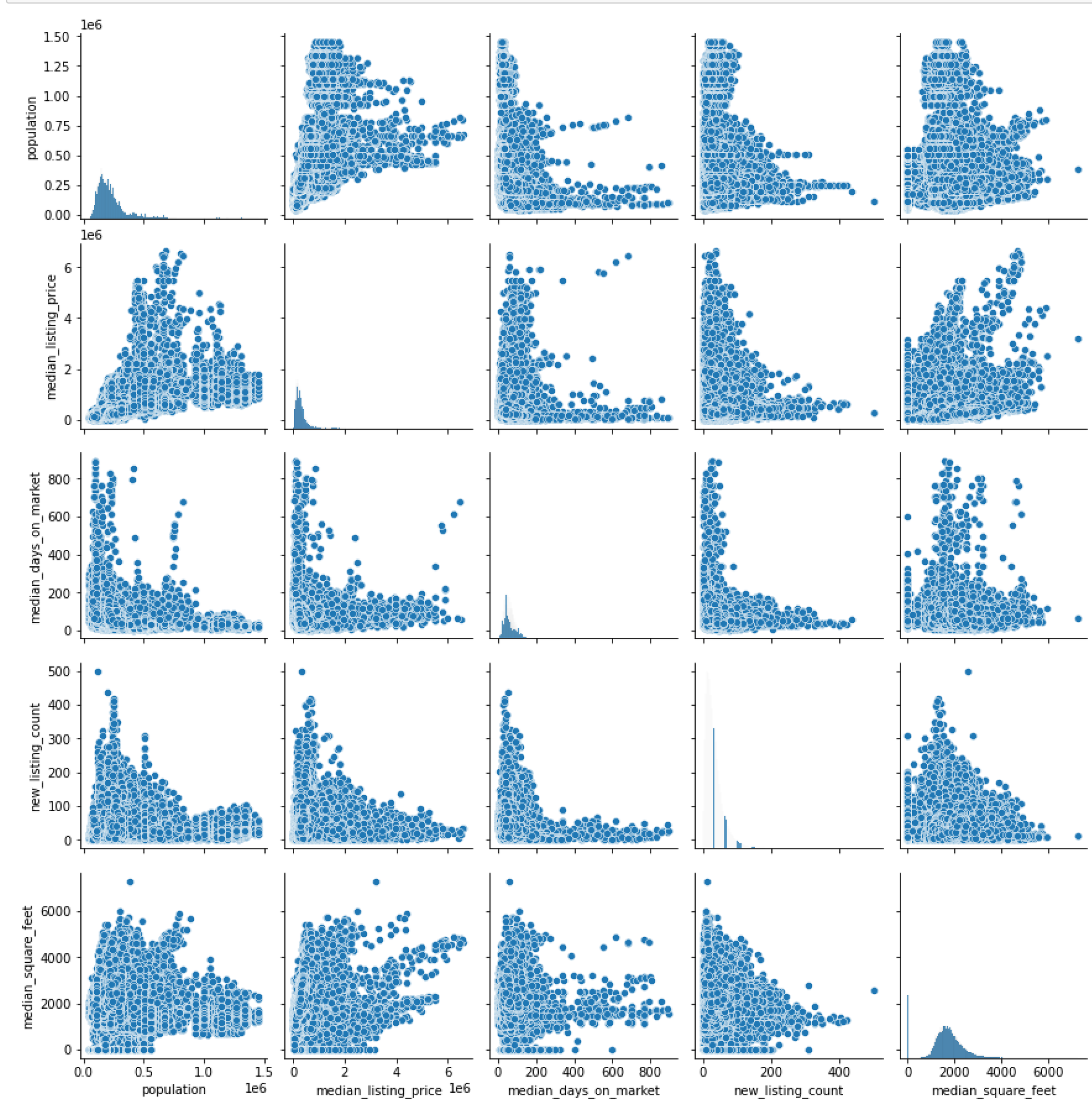
In [52]:
```python
filters = [
   (housing_population['average_listing_price_yy'] < 0) & (housing_population['median_listing_price_yy'] < 0),
   (housing_population['average_listing_price_mm'] < 0) & (housing_population['median_listing_price_mm'] < 0),
]
values = ["Down from last year", "Down from last month"]

housing_population["category"] = np.select(filters, values, default="Increase in prices" )

print('Price ratios for all the records:')
housing_population['category'].value_counts()/len(housing_population['category'])*100

#compared to previous years, there has been an increase in prices by 65.38%
```

Price ratios for all the records:

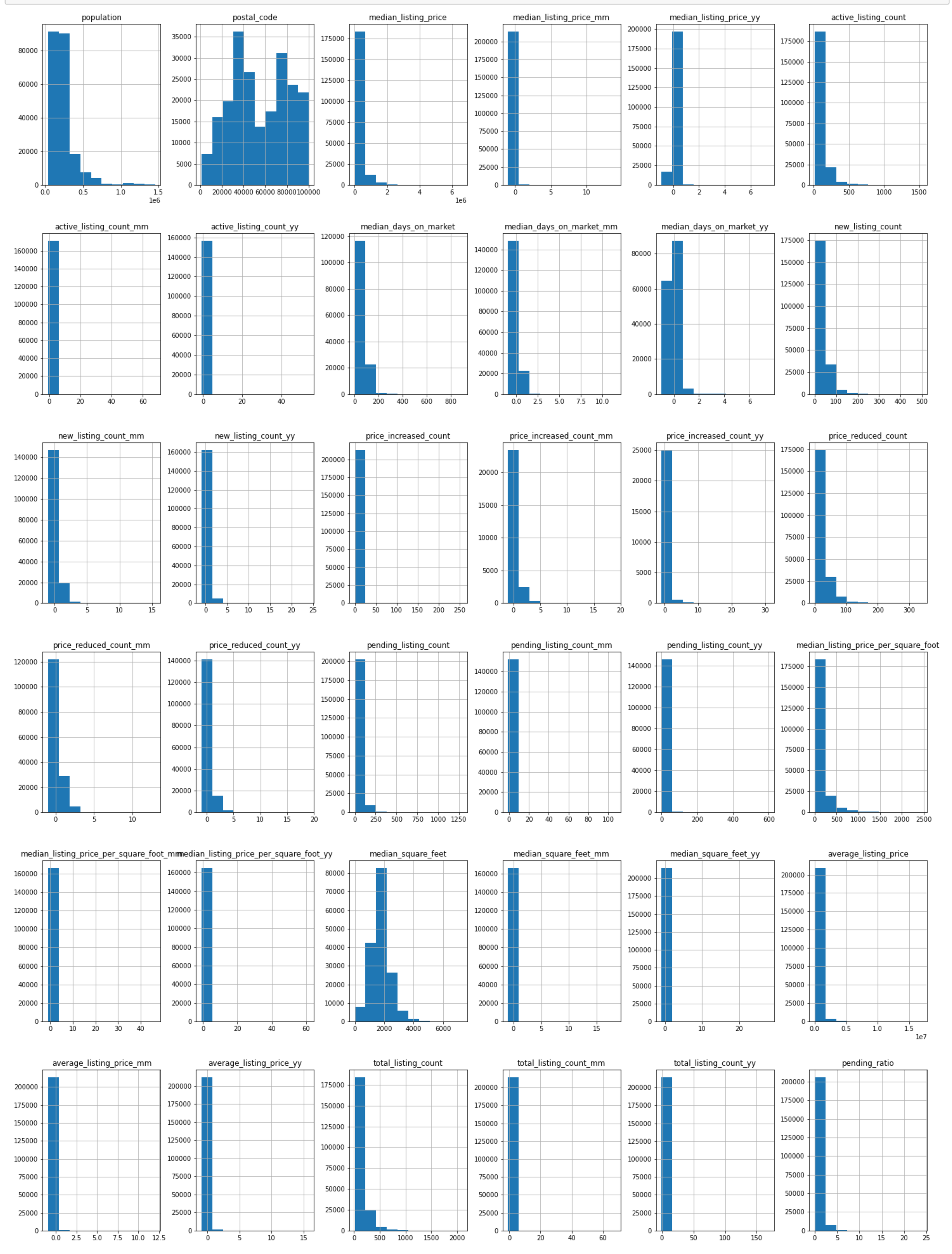
Out[52]:
```
Increase in prices      65.388693
Down from last month    19.763276
Down from last year     14.848030
Name: category, dtype: float64
```

In [54]:
```python
housing_population[["month", "day", "year"]] = housing_population["date"].str.split("/", expand = True)
```

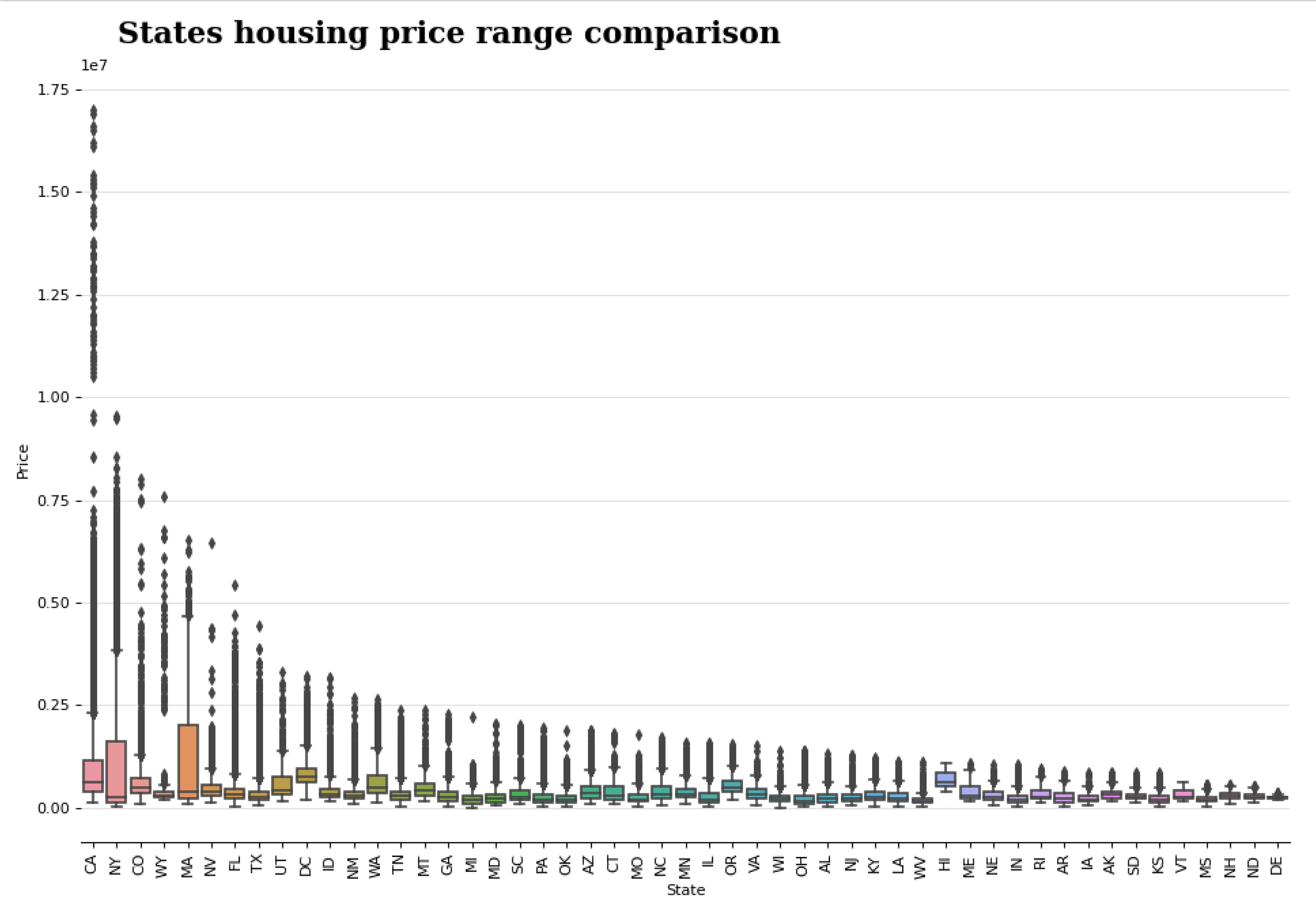
In [55]:
```python
data1=housing_population[['population', 'median_listing_price','median_days_on_market','new_listing_count','median_s
quare_feet']]
sns.pairplot(data1.iloc[:,:5])
plt.show()
```



In [56]:
```python
housing_population.iloc[:,:].hist(figsize=(25,35))
plt.show()
```

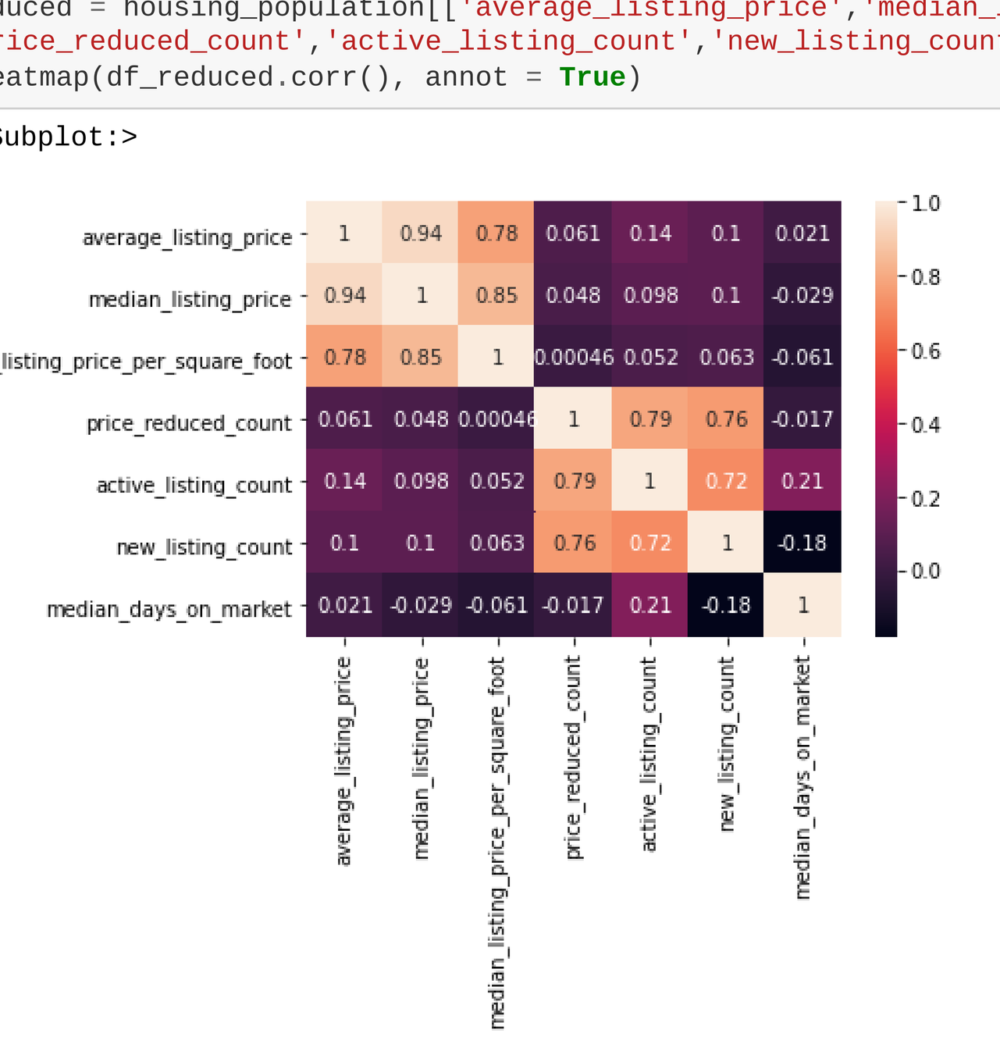


In [57]:
```python
fig, ax = plt.subplots(1,1, figsize=(12, 8), dpi=80)
fig.text(0.09, 1, 'States housing price range comparison', fontsize=20, fontweight='bold', fontfamily='serif')


#plt.figure(figsize=(10, 8), dpi=80)
box_plot = sns.boxplot(x = 'state',y = 'average_listing_price',data = housing_population.sort_values('average_listin
g_price',ascending=False))
plt.ylabel('Price')
plt.xlabel('State')

ax = box_plot.axes
lines = ax.get_lines()
categories = ax.get_xticks()
ax.tick_params(axis='x', rotation=90)

for s in ['top', 'left', 'right']:
    ax.spines[s].set_visible(False)
ax.grid(axis='y', linestyle='-', alpha=0.4)

box_plot.figure.tight_layout()

#fig = box_plot.get_figure()
```



In [59]:
```python
df_reduced = housing_population[['average_listing_price','median_listing_price','median_listing_price_per_square_foo
t','price_reduced_count','active_listing_count','new_listing_count','median_days_on_market','year']]
sns.heatmap(df_reduced.corr(), annot = True)
```

Out[59]: <AxesSubplot:>



In [ ]: