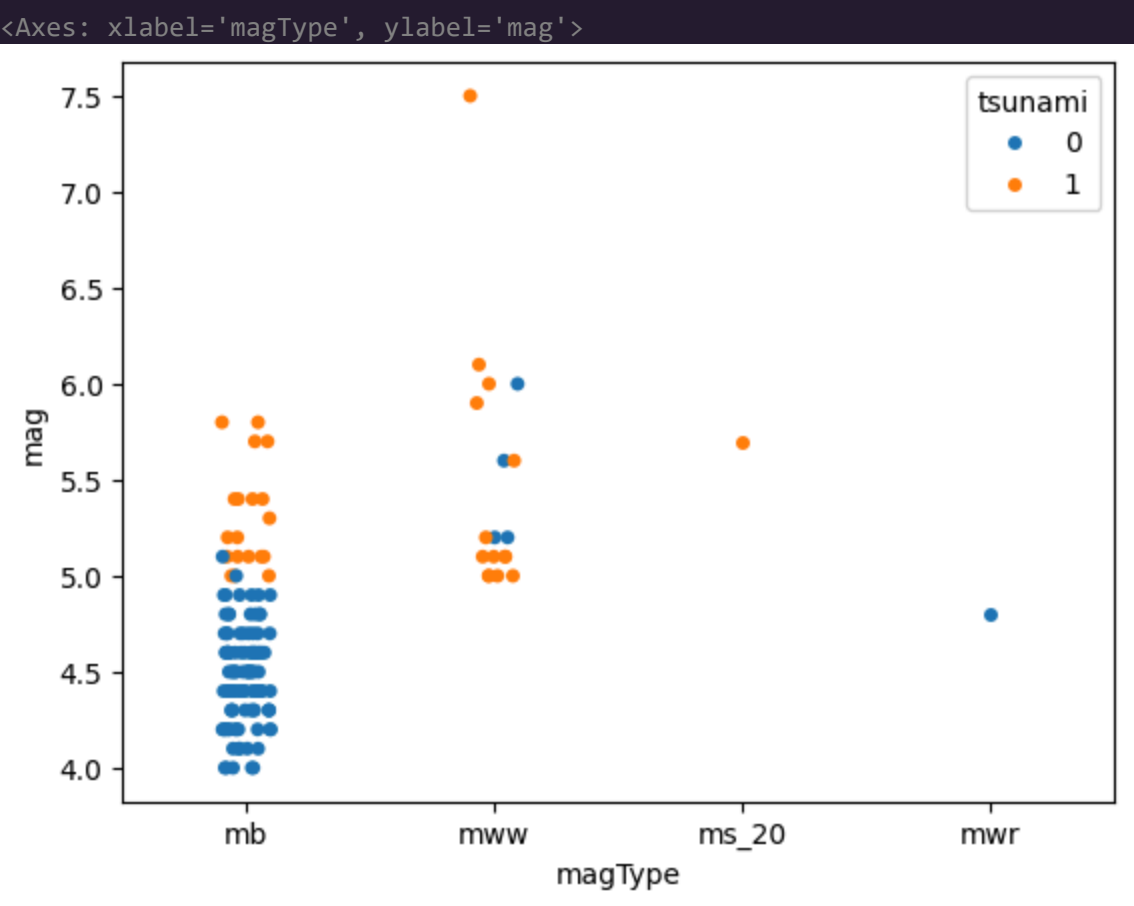


```
1 %matplotlib inline
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import seaborn as sns
5 import pandas as pd
6 fb = pd.read_csv(
7     '/content/fb_stock_prices_2018.csv', index_col='date', parse_dates=True
8 )
9 quakes = pd.read_csv('/content/earthquakes-1.csv')
```

```
1 quakes.assign(
2     time=lambda x: pd.to_datetime(x.time, unit='ms')
3 ).set_index('time').loc['2018-09-28'].query(
4     "parsed_place == 'Indonesia' and tsunami == 1 and mag == 7.5"
5 )
```

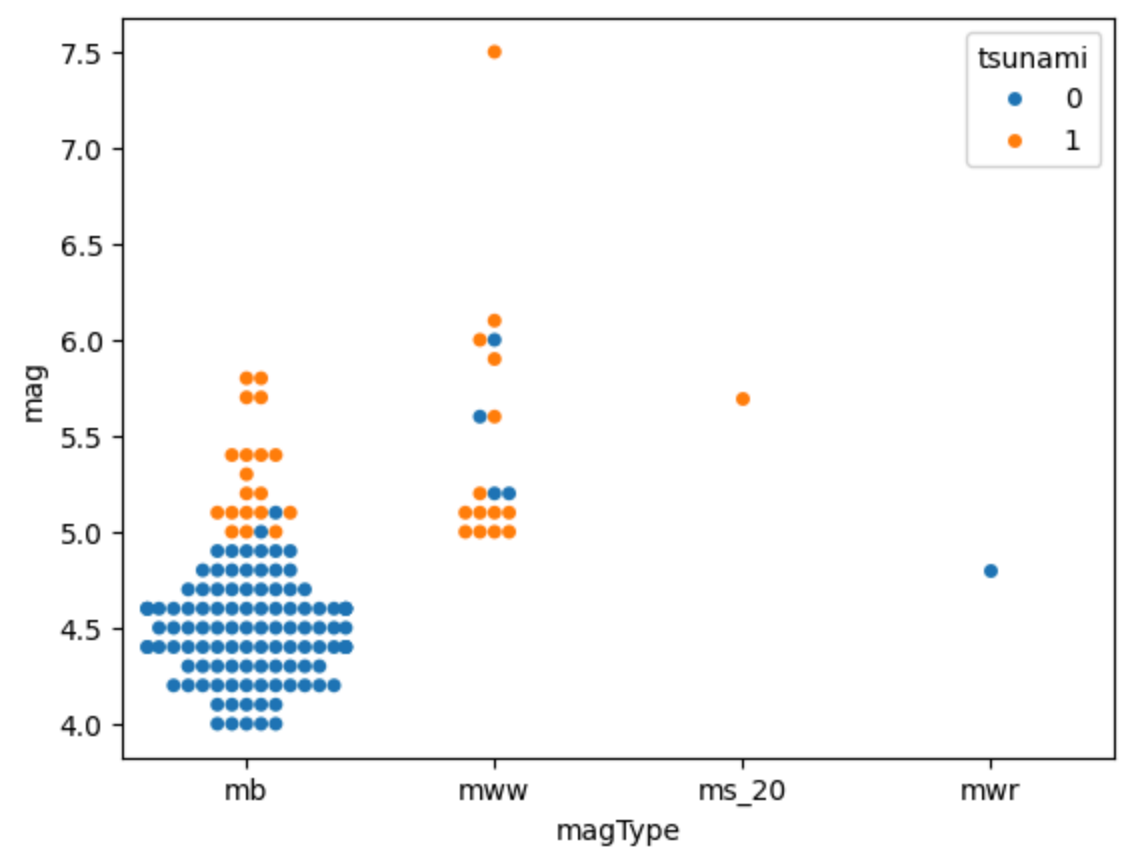
|  | mag | magType | place                     | tsunami | parsed_place |
|--|-----|---------|---------------------------|---------|--------------|
|  | 7.5 | mww     | 78km N of Palu, Indonesia | 1       | Indonesia    |

```
1 sns.stripplot(
2     x='magType',
3     y='mag',
4     hue='tsunami',
5     data=quakes.query('parsed_place == "Indonesia"')
6 )
7
```

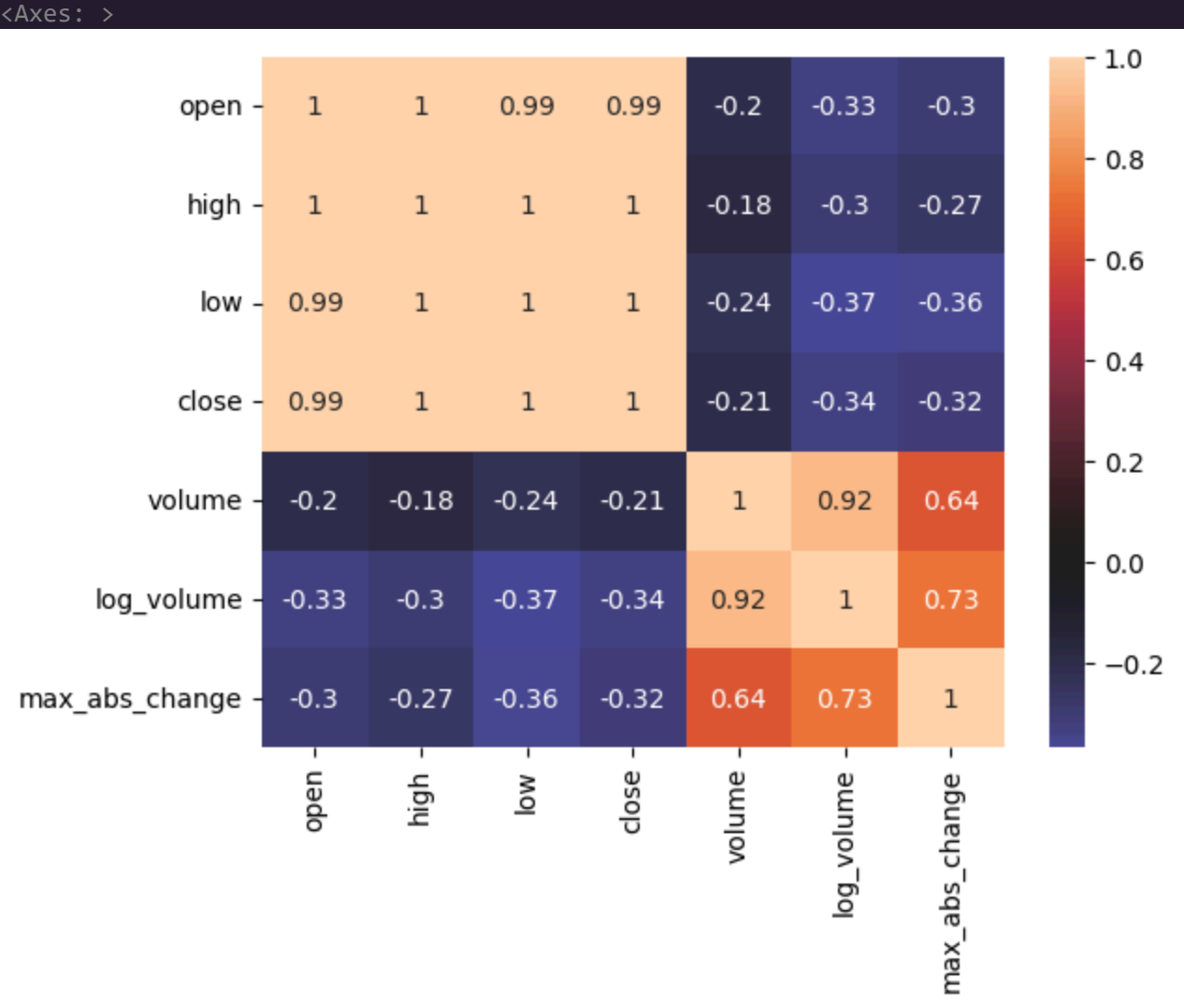


```
1 sns.swarmplot(
2     x='magType',
3     y='mag',
4     hue='tsunami',
5     data=quakes.query('parsed_place == "Indonesia"')
6 )
```

```
<Axes: xlabel='magType', ylabel='mag'>
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398: UserWarning: 10.2% of the po
warnings.warn(msg, UserWarning)
```

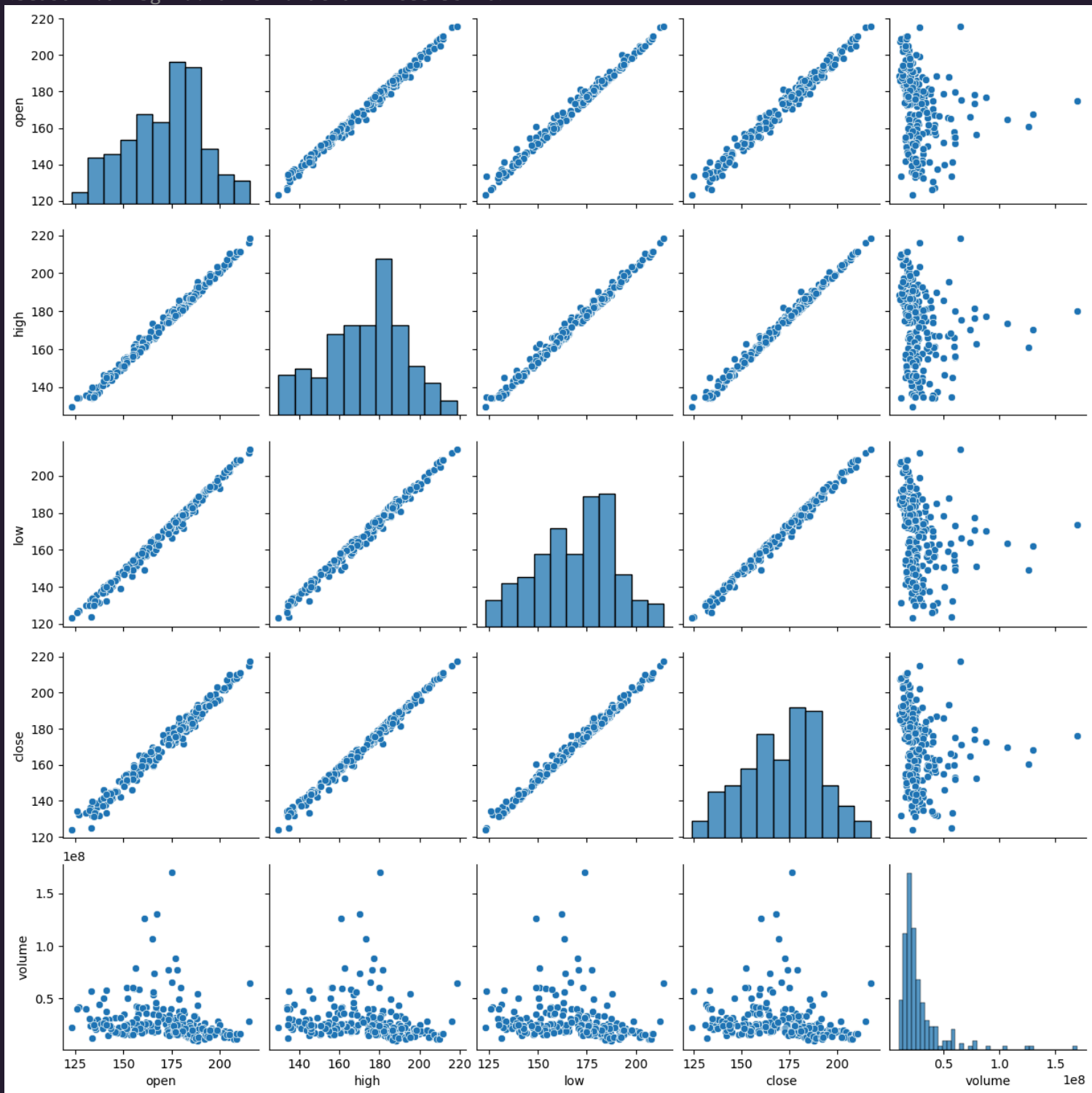


```
1 sns.heatmap(
2     fb.sort_index().assign(
3         log_volume=np.log(fb.volume),
4         max_abs_change=fb.high - fb.low
5     ).corr(), annot=True, center=0
6 )
```



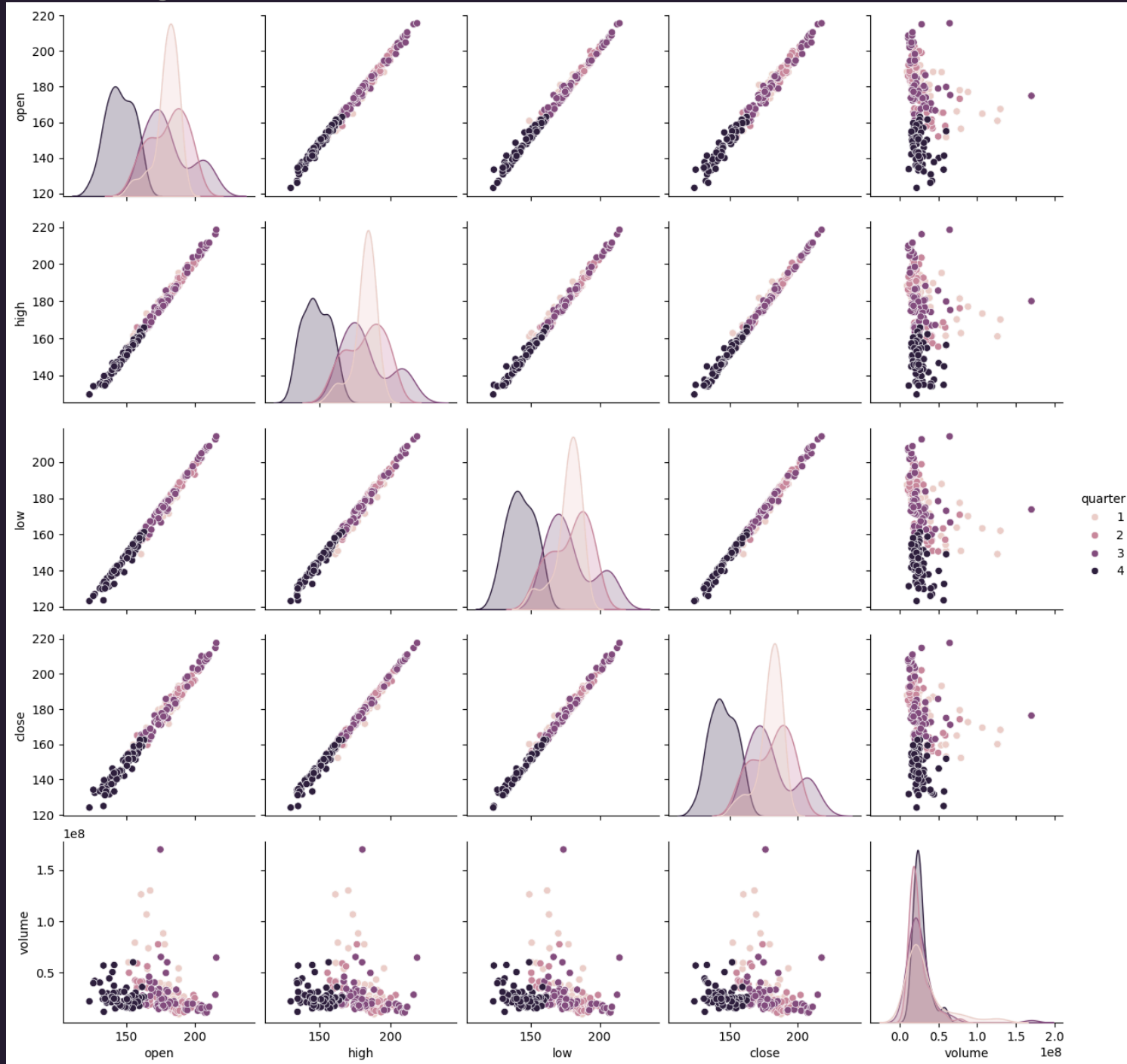
```
1 sns.pairplot(fb)
```

<seaborn.axisgrid.PairGrid at 0x7f7ceef5bf40>



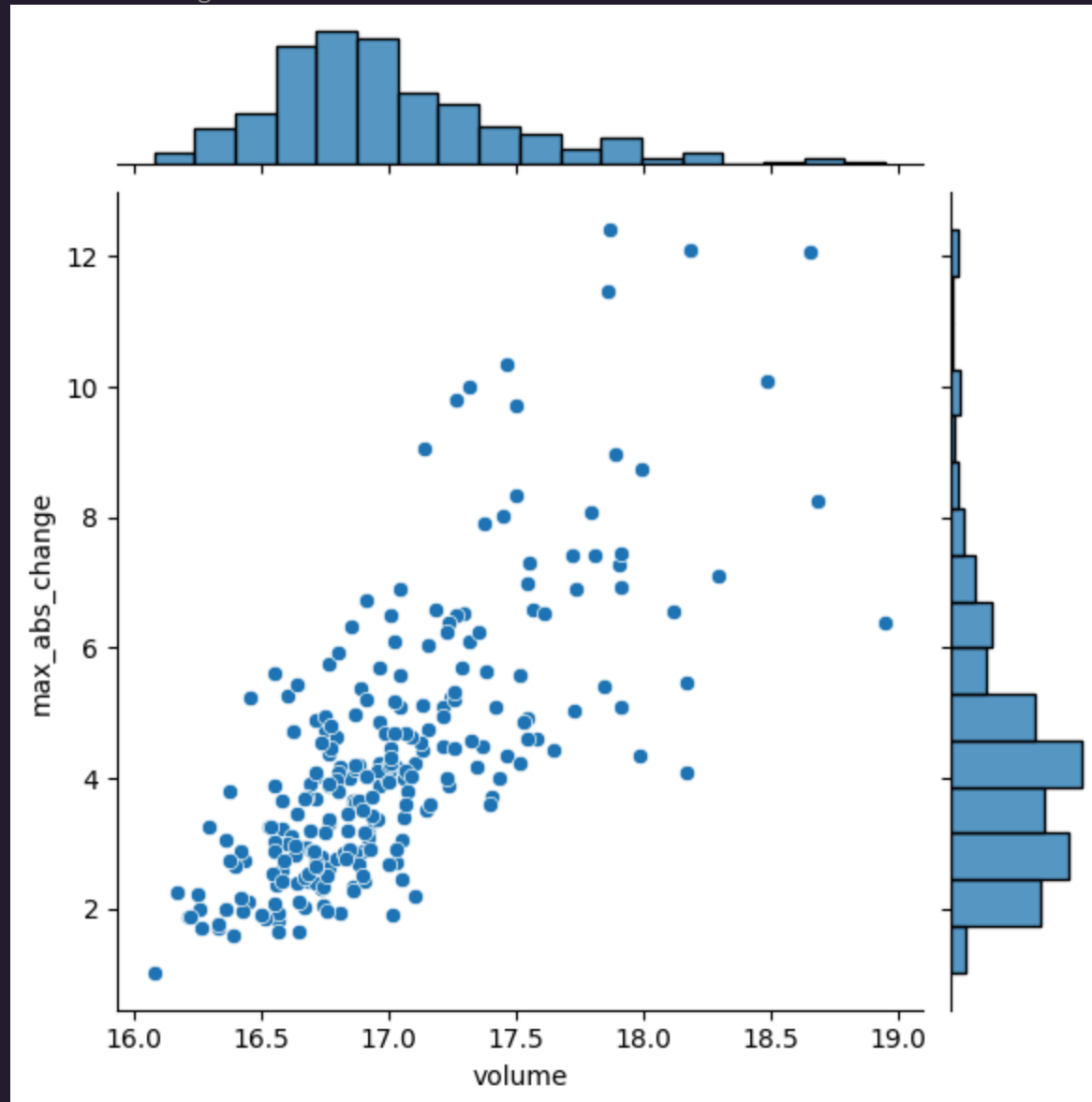
```
1 sns.pairplot(  
2     fb.assign(quarter=lambda x: x.index.quarter),  
3     diag_kind='kde',  
4     hue='quarter'  
5 )
```

<seaborn.axisgrid.PairGrid at 0x7f7ceef7d450>



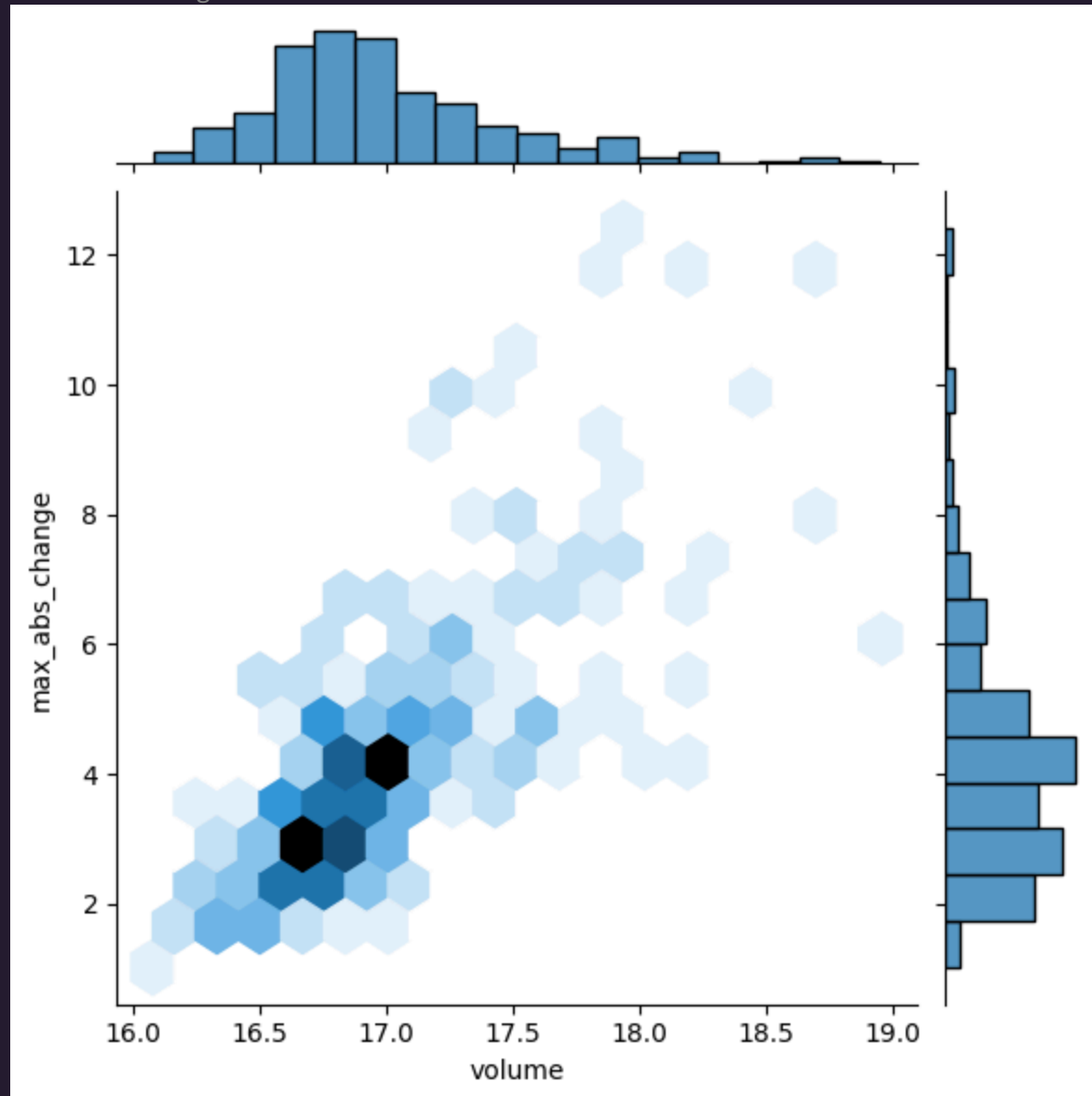
```
1 sns.jointplot(  
2     x='volume',  
3     y='max_abs_change',  
4     data=fb.assign(  
5         volume=np.log(fb.volume),  
6         max_abs_change=fb.high - fb.low  
7     )  
8 )
```

<seaborn.axisgrid.JointGrid at 0x7f7ced7c45e0>



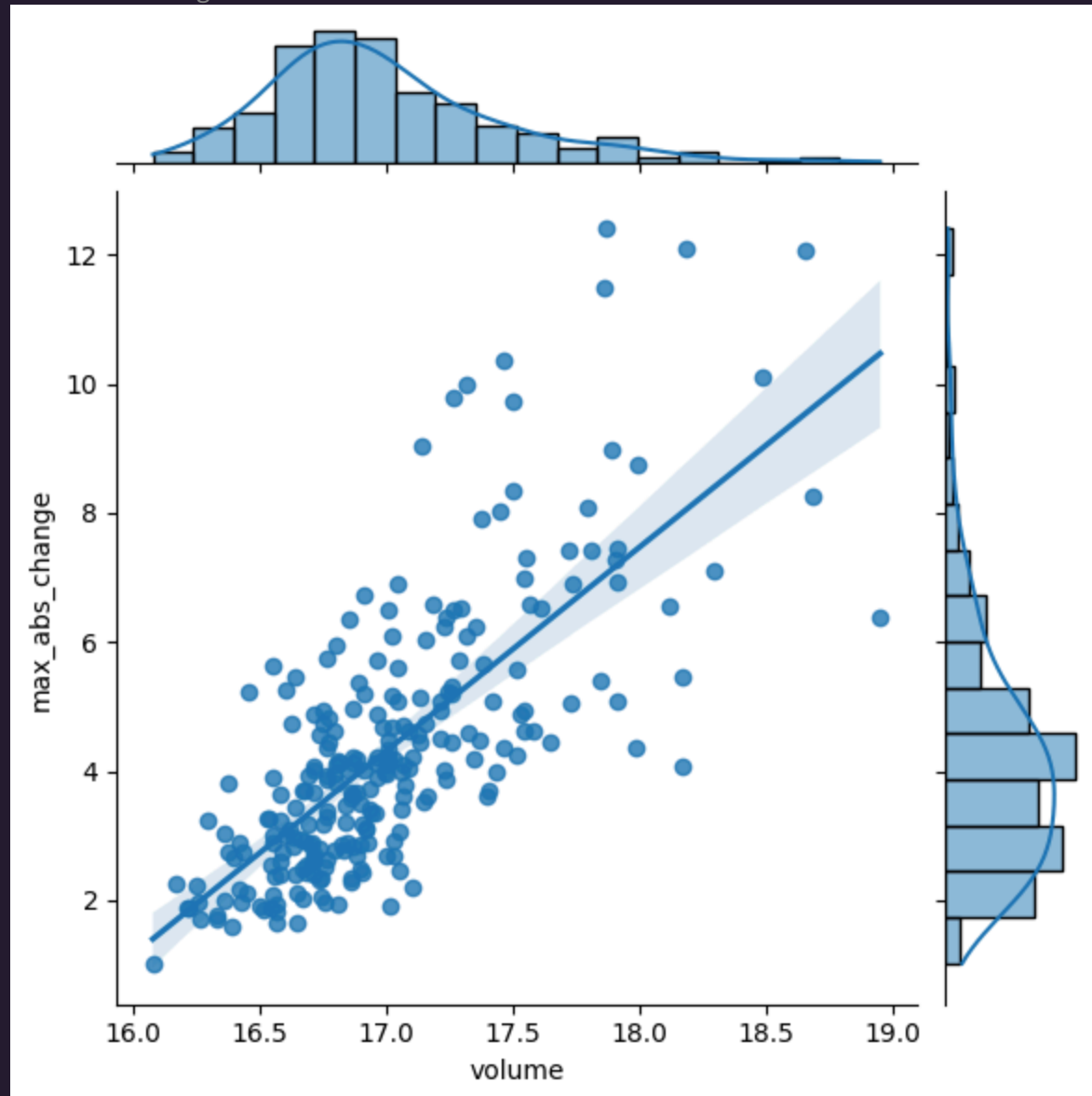
```
1 sns.jointplot(  
2     x='volume',  
3     y='max_abs_change',  
4     kind='hex',  
5     data=fb.assign(  
6         volume=np.log(fb.volume),  
7         max_abs_change=fb.high - fb.low  
8     )  
9 )
```

<seaborn.axisgrid.JointGrid at 0x7f7ced447250>



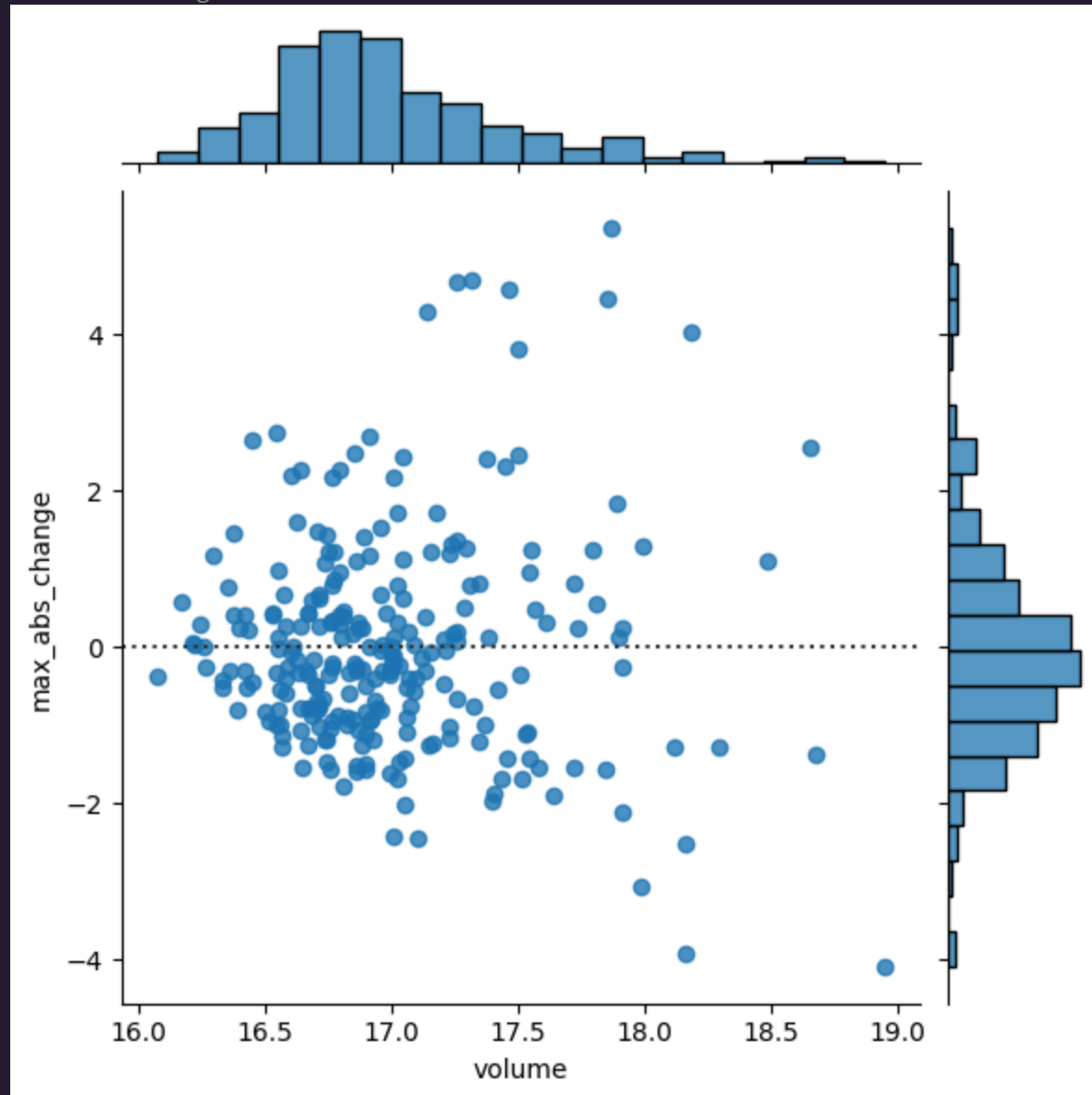
```
1 sns.jointplot(  
2     x='volume',  
3     y='max_abs_change',  
4     kind='reg',  
5     data=fb.assign(  
6         volume=np.log(fb.volume),  
7         max_abs_change=fb.high - fb.low  
8     )  
9 )
```

<seaborn.axisgrid.JointGrid at 0x7f7ced677c70>



```
1 sns.jointplot(  
2     x='volume',  
3     y='max_abs_change',  
4     kind='resid',  
5     data=fb.assign(  
6         volume=np.log(fb.volume),  
7         max_abs_change=fb.high - fb.low  
8     )  
9 )
```

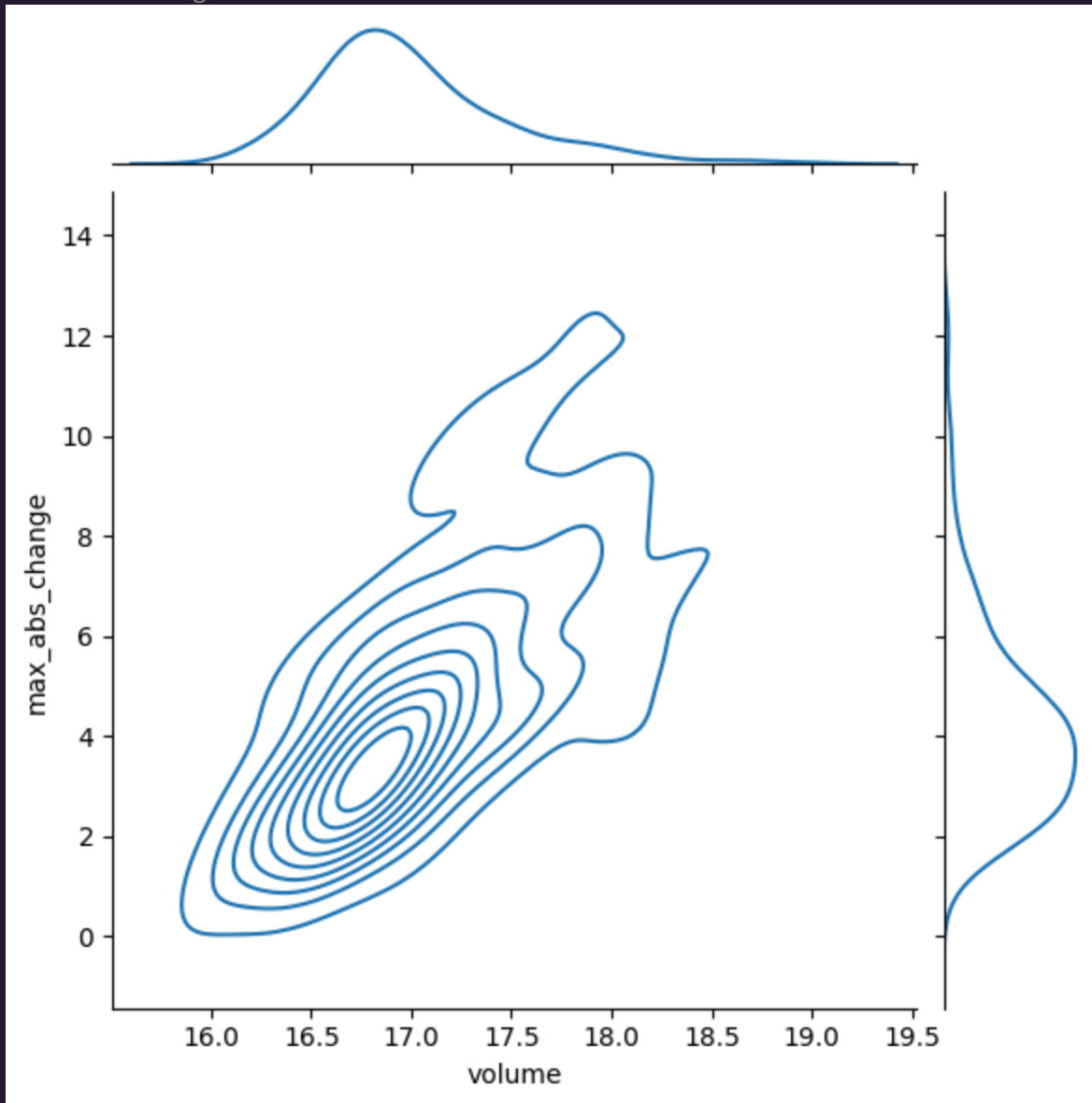
<seaborn.axisgrid.JointGrid at 0x7f7ceef7cdf0>



```
1 sns.jointplot(  
2     x='volume',  
3     y='max_abs_change',  
4     kind='kde',  
5     data=fb.assign(  
6         volume=np.log(fb.volume),  
7         max_abs_change=fb.high - fb.low  
8     )  
9 )
```



```
<seaborn.axisgrid.JointGrid at 0x7f7ced2b4dc0>
```



```
1 fb_reg_data = fb.assign(  
2   volume=np.log(fb.volume),  
3   max_abs_change=fb.high - fb.low  
4 ).iloc[:, -2:]
```

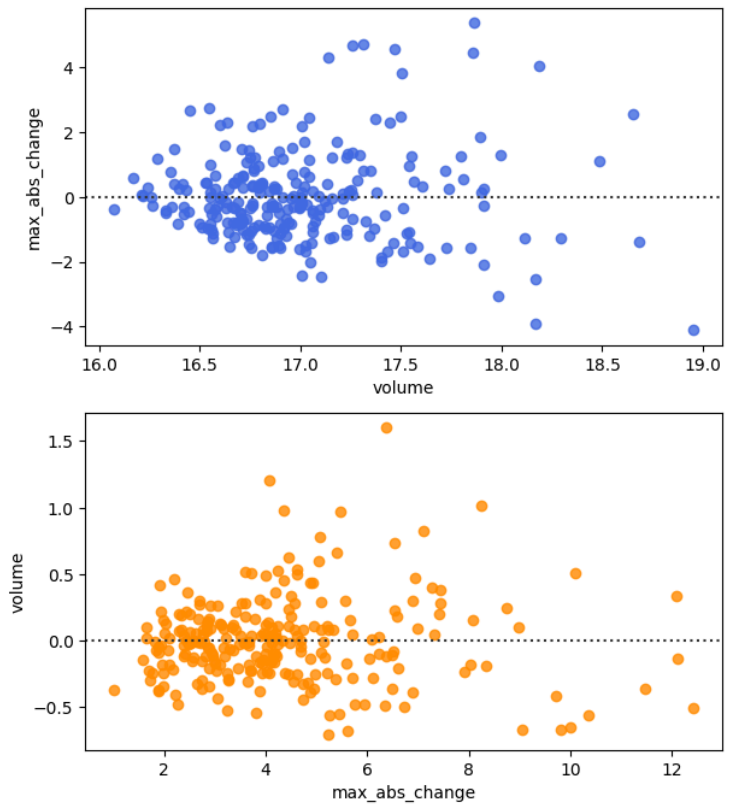
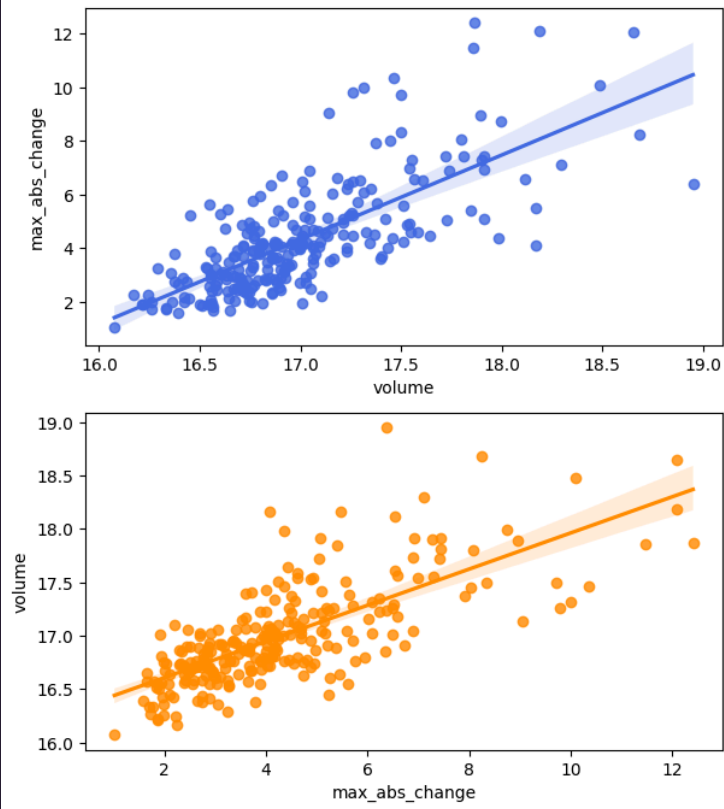
```
1 import itertools  
2 iterator = itertools.repeat("I'm an iterator", 1)  
3 for i in iterator:  
4     print(f'-->{i}')  
5 print('This printed once because the iterator has been exhausted')  
6 for i in iterator:  
7     print(f'-->{i}')
```

```
-->I'm an iterator  
This printed once because the iterator has been exhausted
```

```
1 iterable = list(itertools.repeat("I'm an iterable", 1))  
2 for i in iterable:  
3     print(f'-->{i}')  
4 print('This prints again because it\'s an iterable:')  
5 for i in iterable:  
6     print(f'-->{i}')
```

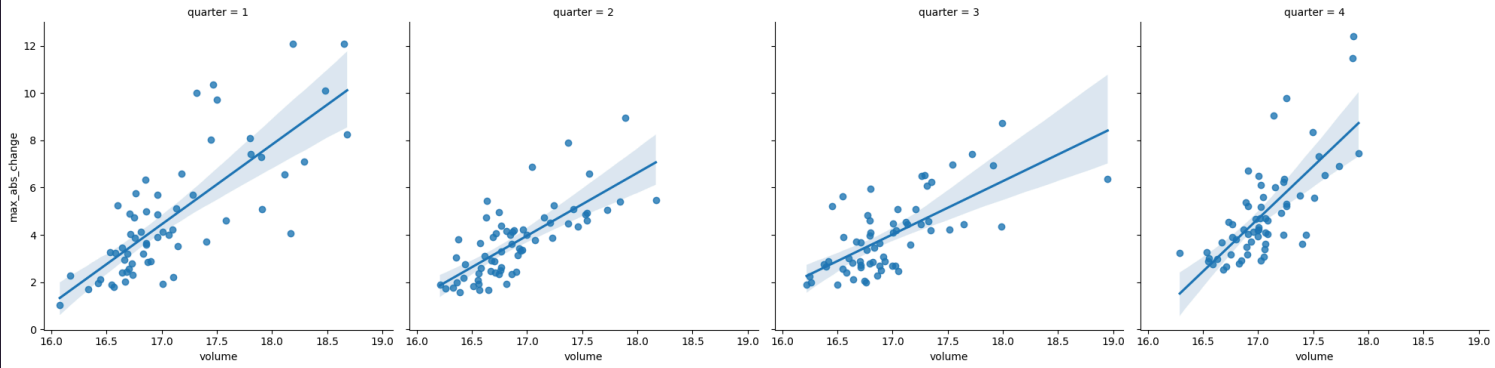
```
-->I'm an iterable  
This prints again because it's an iterable:  
-->I'm an iterable
```

```
1 from reg_resid_plot import reg_resid_plots  
2 reg_resid_plots(fb_reg_data)
```



```
1 sns.lmplot(  
2     x='volume',  
3     y='max_abs_change',  
4     data=fb.assign(  
5         volume=np.log(fb.volume),  
6         max_abs_change=fb.high - fb.low,  
7         quarter=lambda x: x.index.quarter  
8     ),  
9     col='quarter'  
10 )
```

<seaborn.axisgrid.FacetGrid at 0x7f7cecd4ce20>



```
1 sns.boxenplot(  
2     x='quarter',  
3     y='max_abs_change',  
4     data=fb,  
5     showfliers=False,  
6     notch=True,  
7     boxprops=dict(facecolor='lightblue', edgecolor='darkblue'),  
8     whiskerprops=dict(facecolor='lightblue', edgecolor='darkblue'),  
9     flierprops=dict(facecolor='lightblue', edgecolor='darkblue'),  
10    )
```