

Name: Xander Sam E. Galapia

Section: CPE22S3

Introduction to Seaborn

Setup

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import pandas as pd
fb = pd.read_csv(
    'data/fb_stock_prices_2018.csv', index_col='date', parse_dates=True
)
quakes = pd.read_csv('data/earthquakes.csv')
```

Categorical data

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

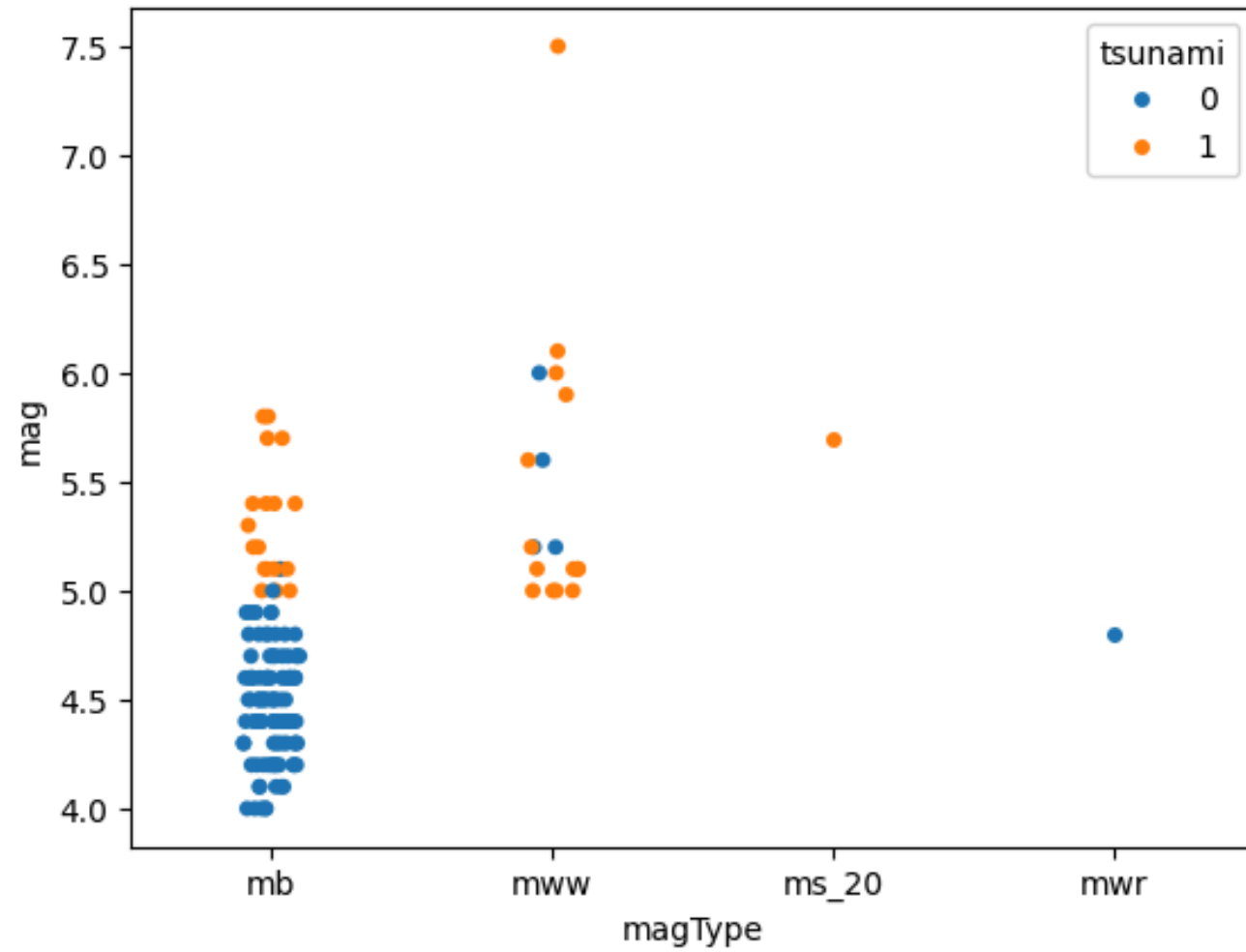
	mag	magType	place	tsunami	parsed_place
time					
2018-09-28 10:02:43.480	7.5	mww	78km N of Palu, Indonesia	1	Indonesia



stripplot()

```
sns.stripplot(
    x='magType',
    y='mag',
    hue='tsunami',
    data=quakes.query('parsed_place == "Indonesia"')
)
```

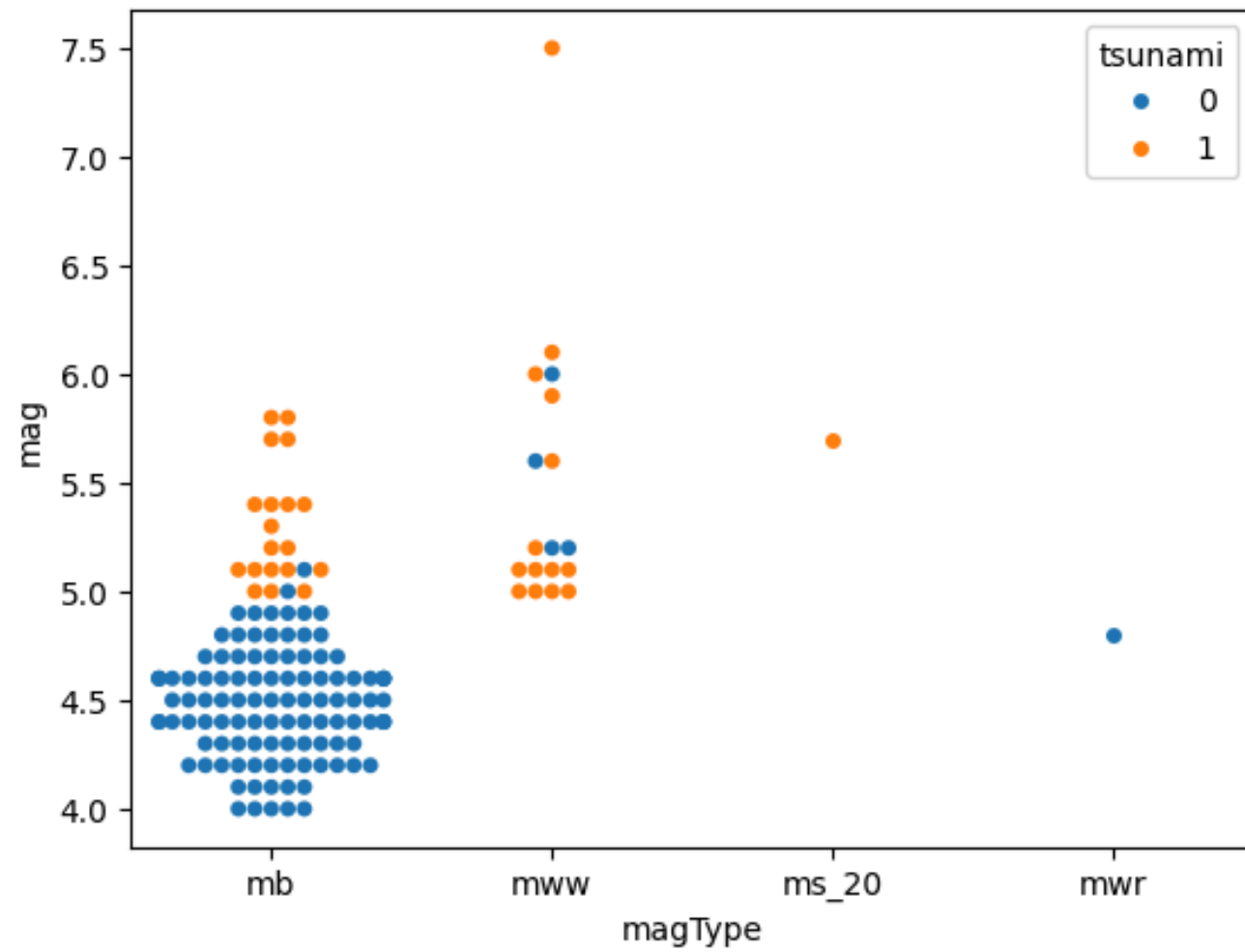
```
<Axes: xlabel='magType', ylabel='mag'>
```



- swarmplot()

```
sns.swarmplot(
    x='magType',
    y='mag',
    hue='tsunami',
    data=quakes.query('parsed_place == "Indonesia"')
)
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398: UserWarning: 10.2% of the points cannot be placed; you may want to decrease the size of the marker
  warnings.warn(msg, UserWarning)
```

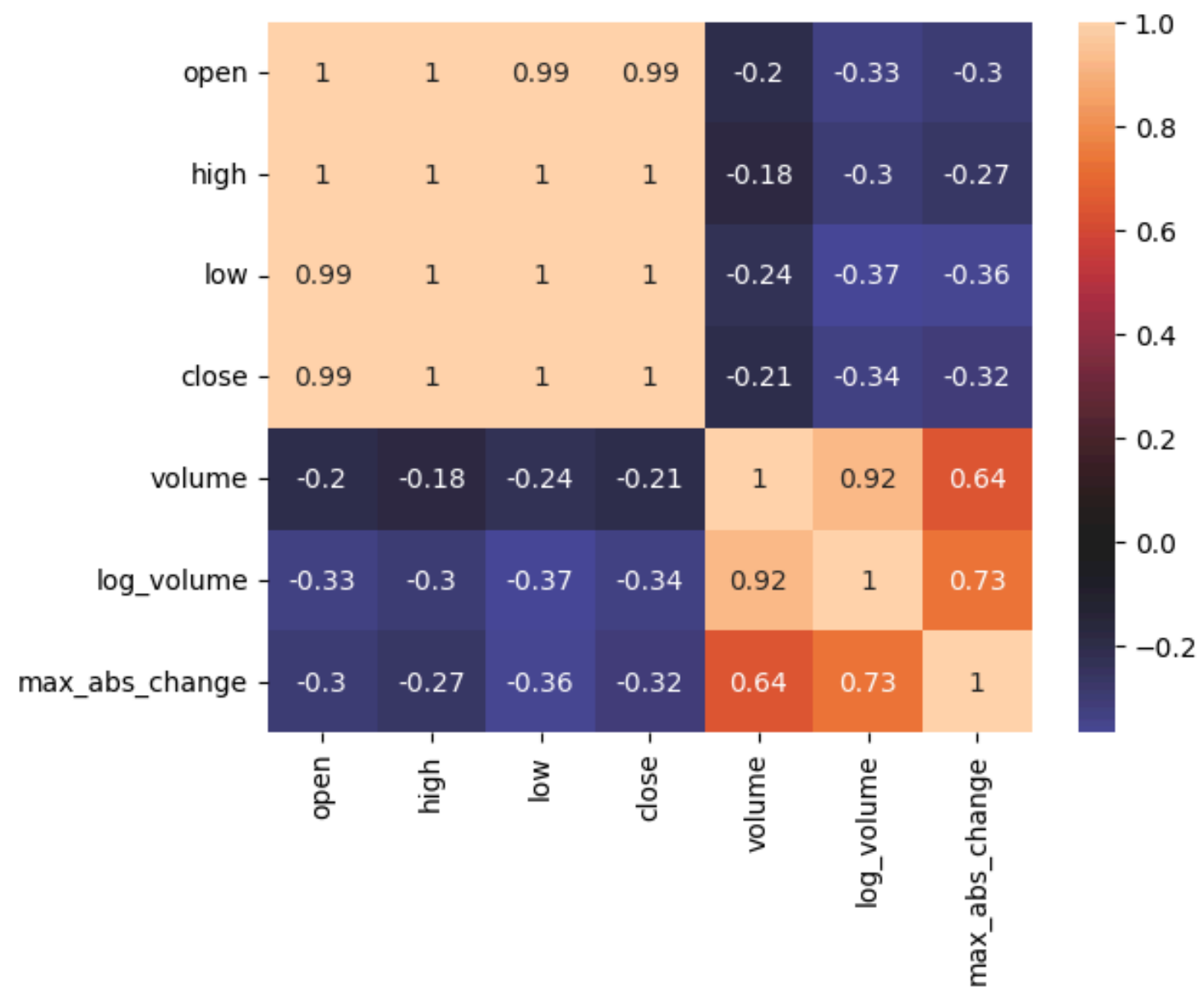


✦ Correlations and Heatmaps

- ▼ heatmap()

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

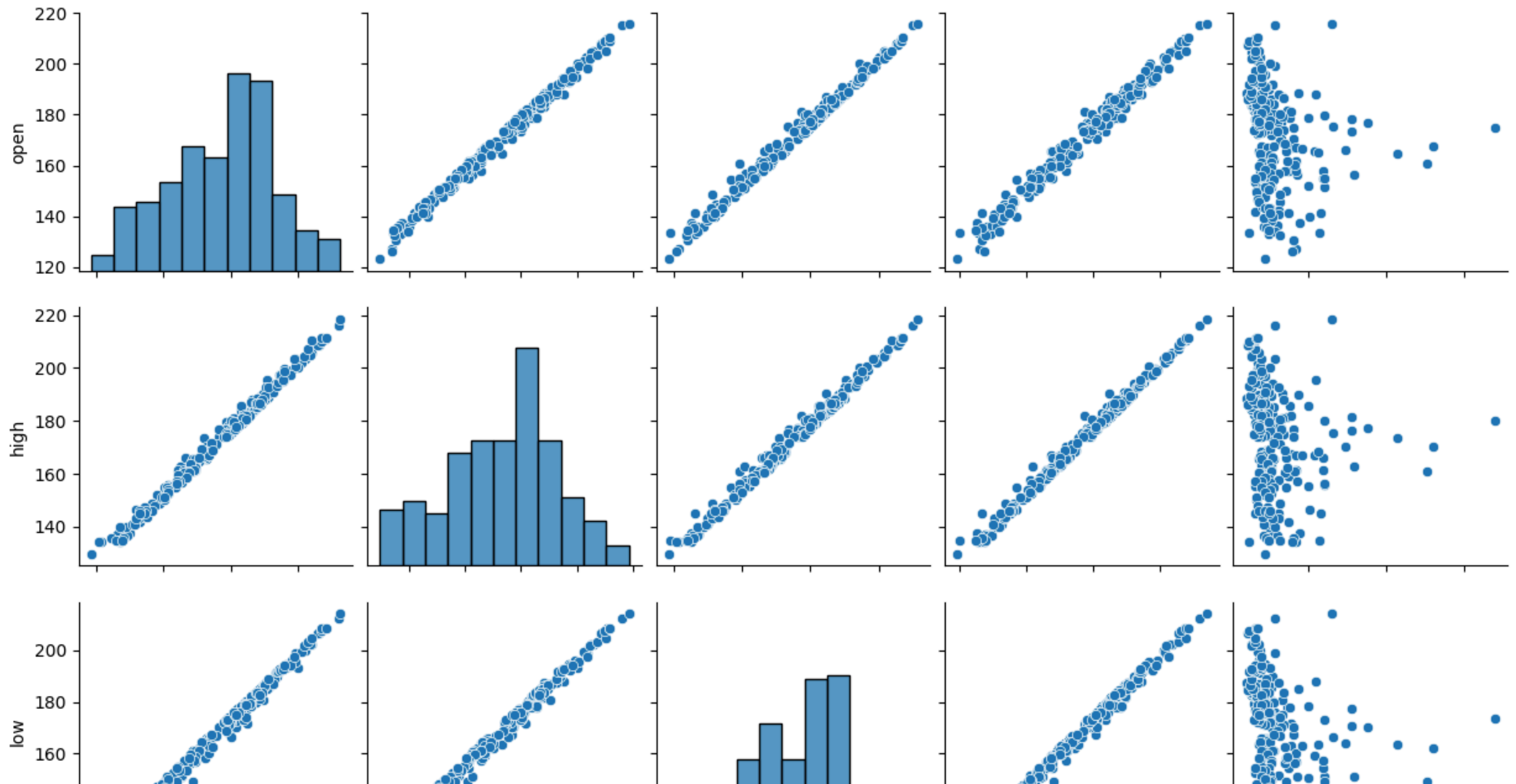
<Axes: >



✕ pairplot()

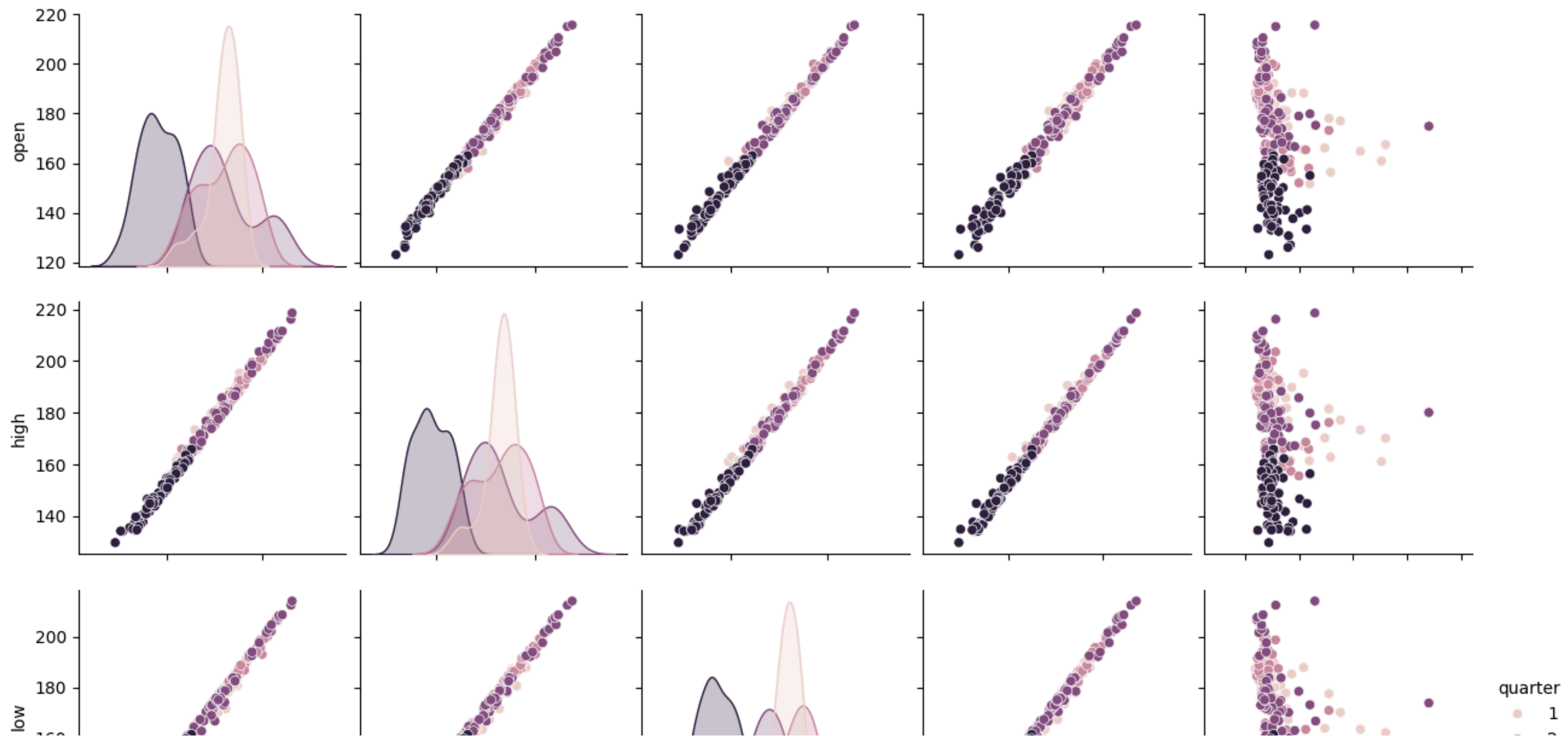
```
sns.pairplot(fb)
```

<seaborn.axisgrid.PairGrid at 0x78b509717760>



```
sns.pairplot(  
    fb.assign(quarter=lambda x: x.index.quarter),  
    diag_kind='kde',  
    hue='quarter'  
)
```

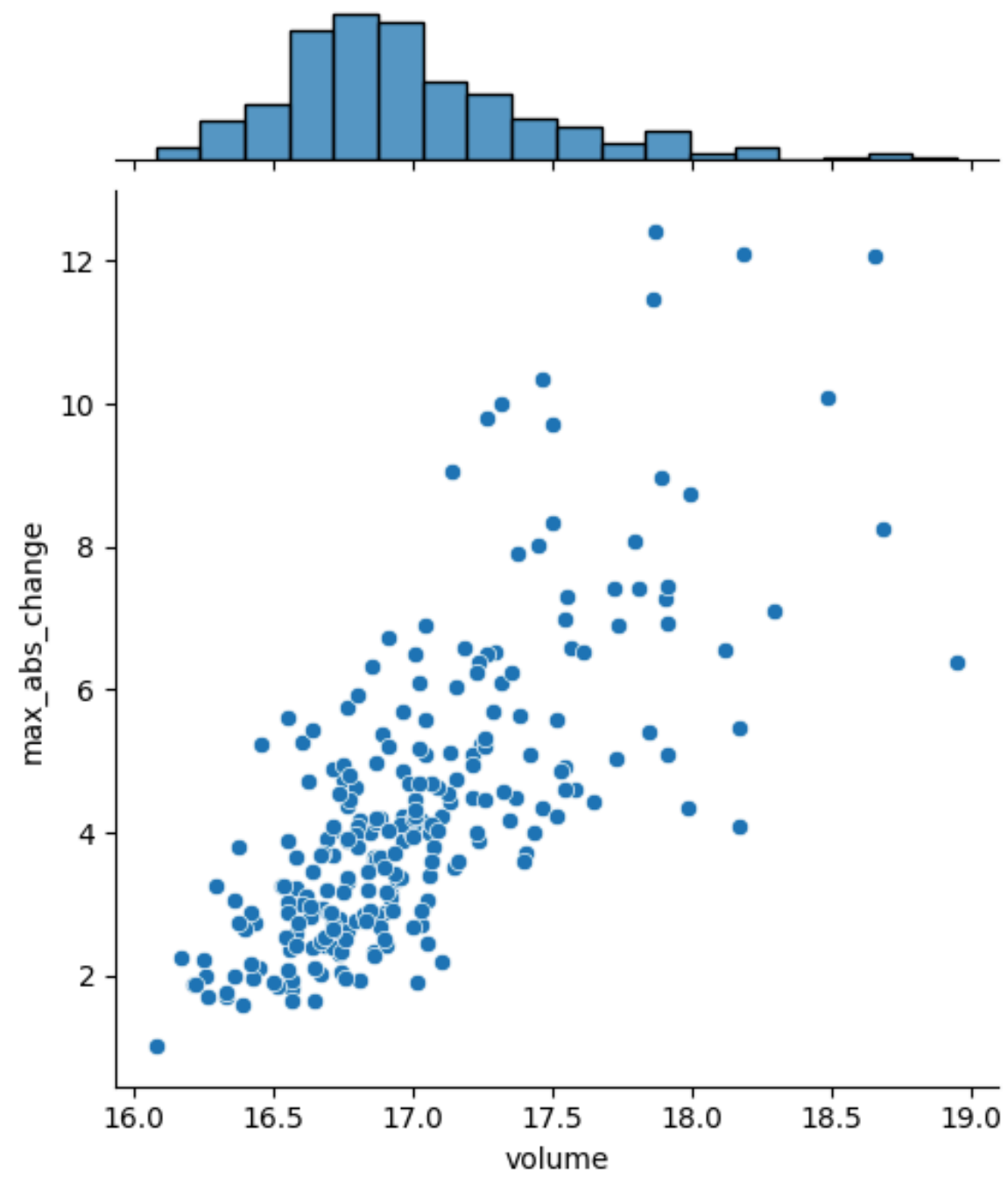
<seaborn.axisgrid.PairGrid at 0x78b5099c4c70>



✓ jointplot()

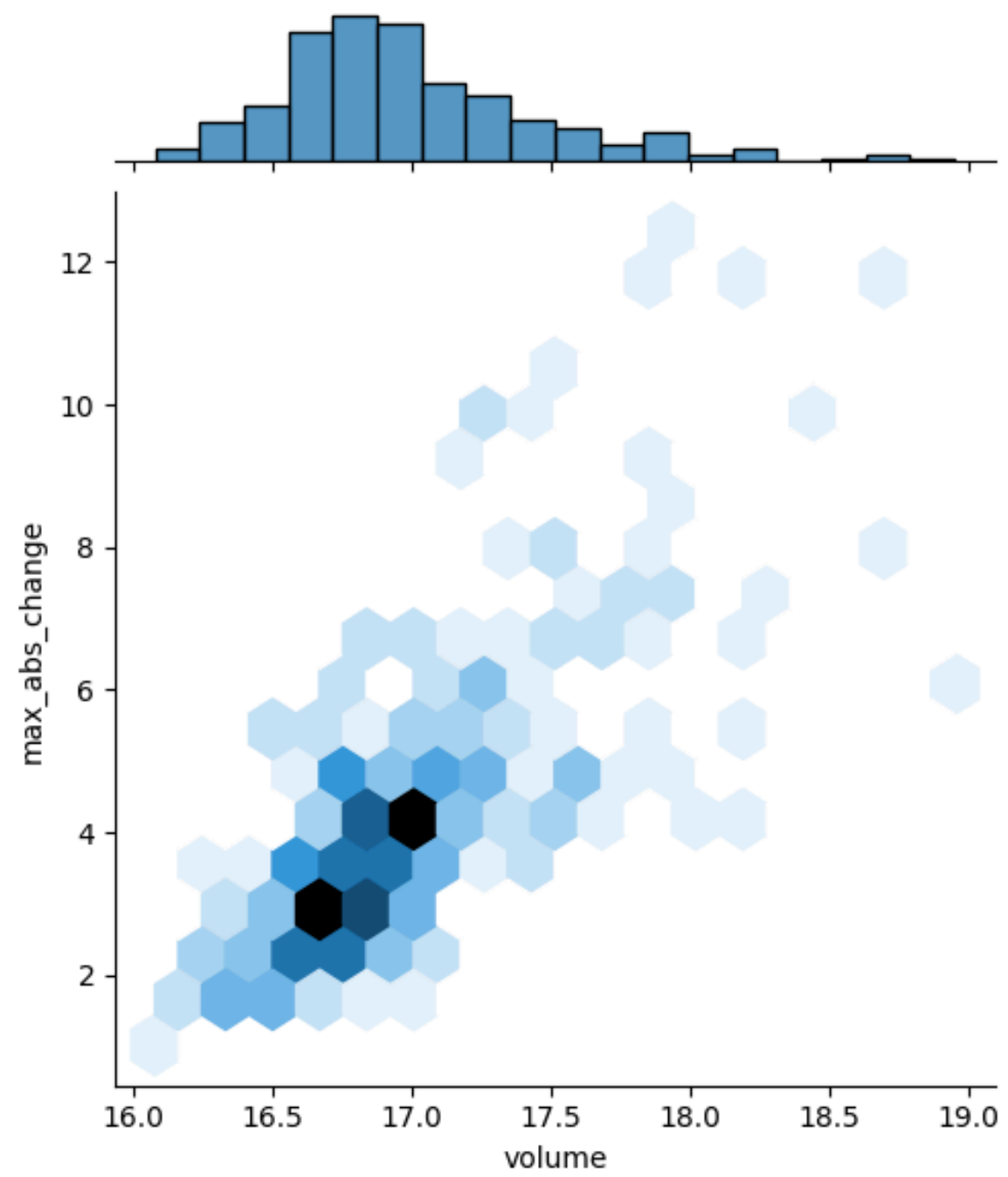
```
sns.jointplot(  
    x='volume',  
    y='max_abs_change',  
    data=fb.assign(  
        volume=np.log(fb.volume),  
        max_abs_change=fb.high - fb.low  
    )  
)
```

<seaborn.axisgrid.JointGrid at 0x78b500786770>



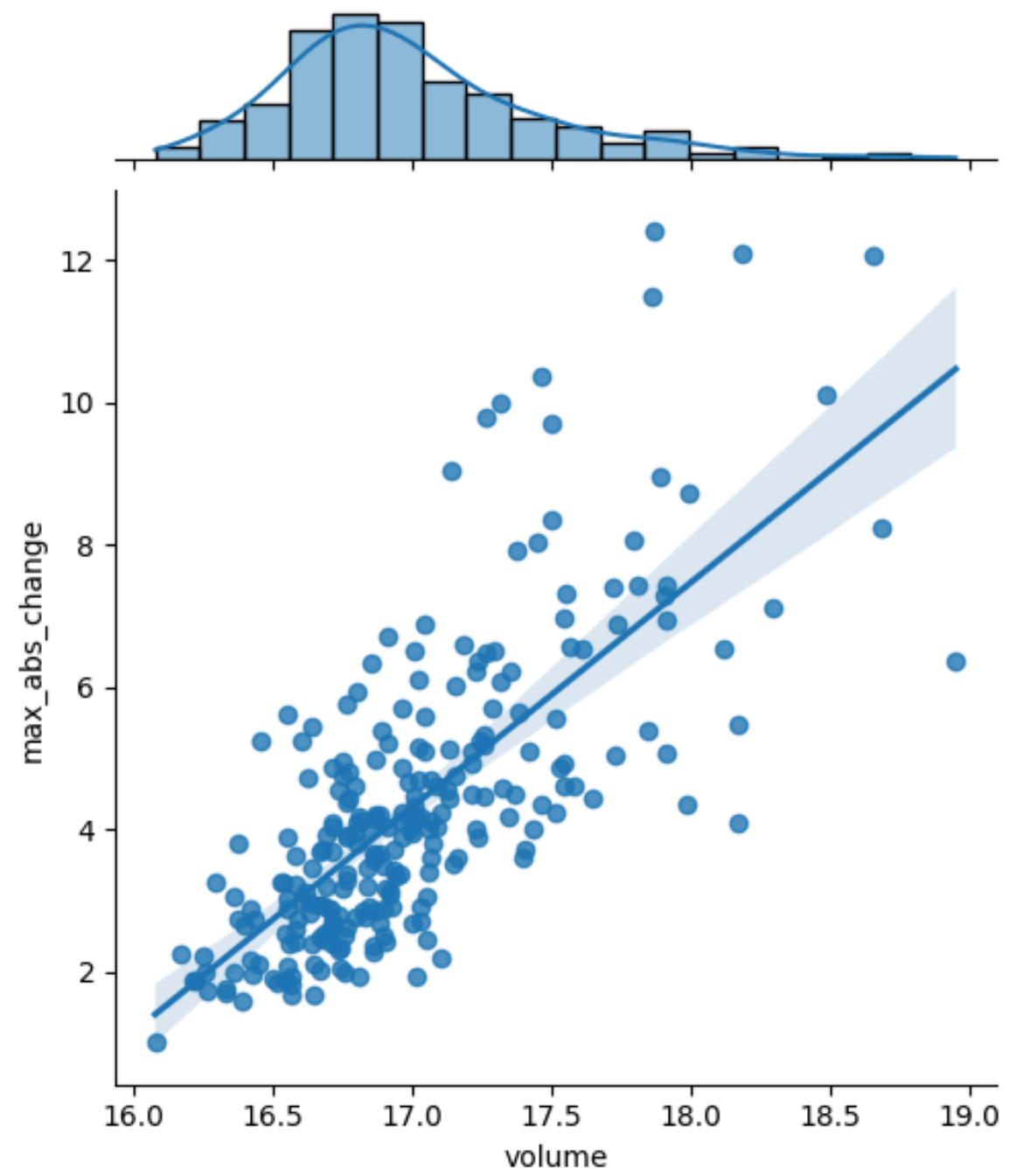
```
sns.jointplot(  
    x='volume',  
    y='max_abs_change',  
    kind='hex',  
    data=fb.assign(  
        volume=np.log(fb.volume),  
        max_abs_change=fb.high - fb.low  
    )  
)
```

<seaborn.axisgrid.JointGrid at 0x78b509a0c070>



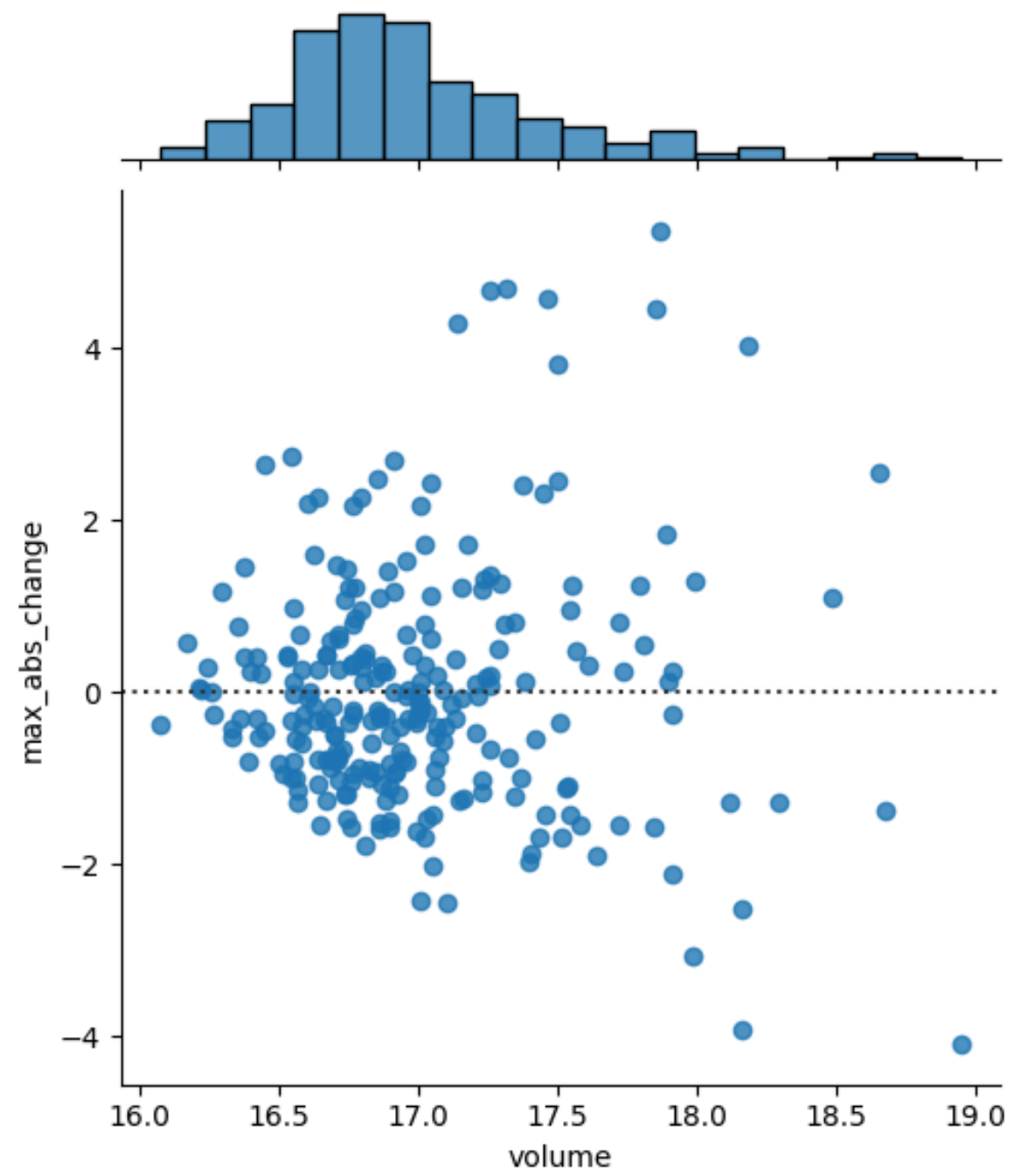
```
sns.jointplot(  
    x='volume',  
    y='max_abs_change',  
    kind='reg',  
    data=fb.assign(  
        volume=np.log(fb.volume),  
        max_abs_change=fb.high - fb.low  
    )  
)
```


<seaborn.axisgrid.JointGrid at 0x78b5004acfa0>



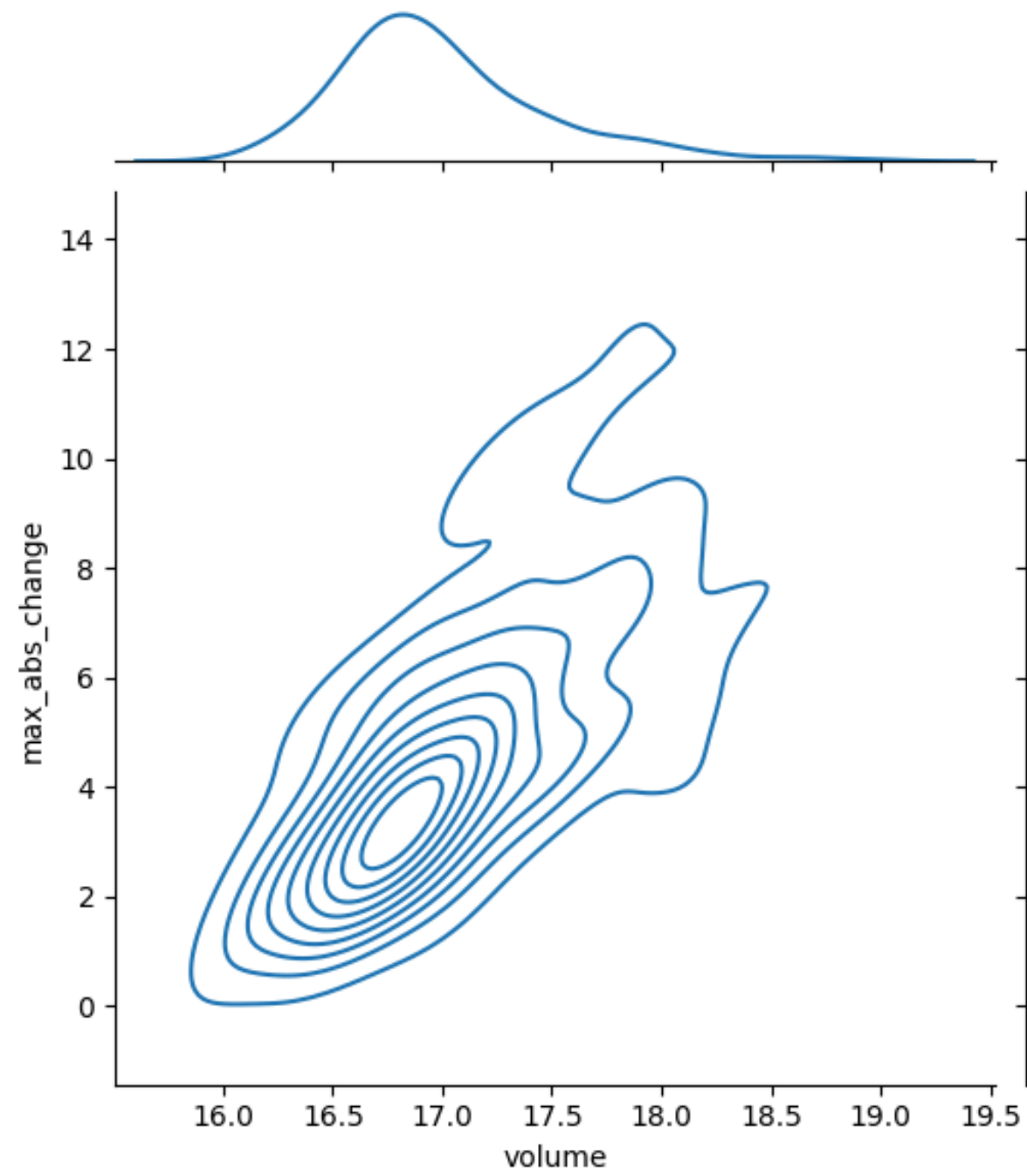
```
sns.jointplot(  
    x='volume',  
    y='max_abs_change',  
    kind='resid',  
    data=fb.assign(  
        volume=np.log(fb.volume),  
        max_abs_change=fb.high - fb.low  
    )  
)
```

<seaborn.axisgrid.JointGrid at 0x78b5004329b0>



```
sns.jointplot(  
    x='volume',  
    y='max_abs_change',  
    kind='kde',  
    data=fb.assign(  
        volume=np.log(fb.volume),  
        max_abs_change=fb.high - fb.low  
    )  
)
```

<seaborn.axisgrid.JointGrid at 0x78b50033c2b0>



▽ Regression plots

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

```
import itertools
```

```

iterator = itertools.repeat("I'm an iterator", 1)

for i in iterator:
    print(f'-->{i}')
print('This printed once because the iterator has been exhausted')
for i in iterator:
    print(f'-->{i}')

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

```

```

iterable = list(itertools.repeat("I'm an iterable", 1))

```

```

for i in iterable:
    print(f'-->{i}')
print('This prints again because it\'s an iterable:')
for i in iterable:
    print(f'-->{i}')

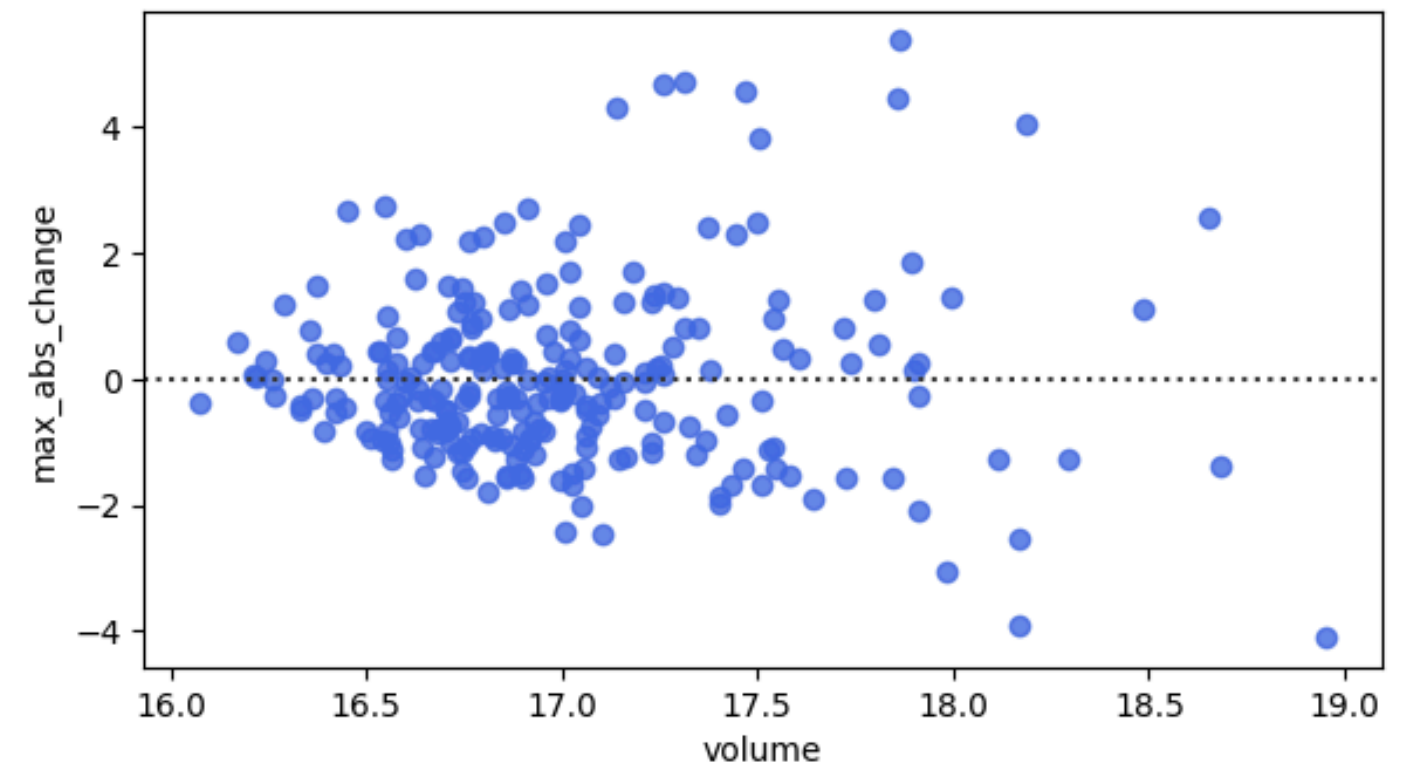
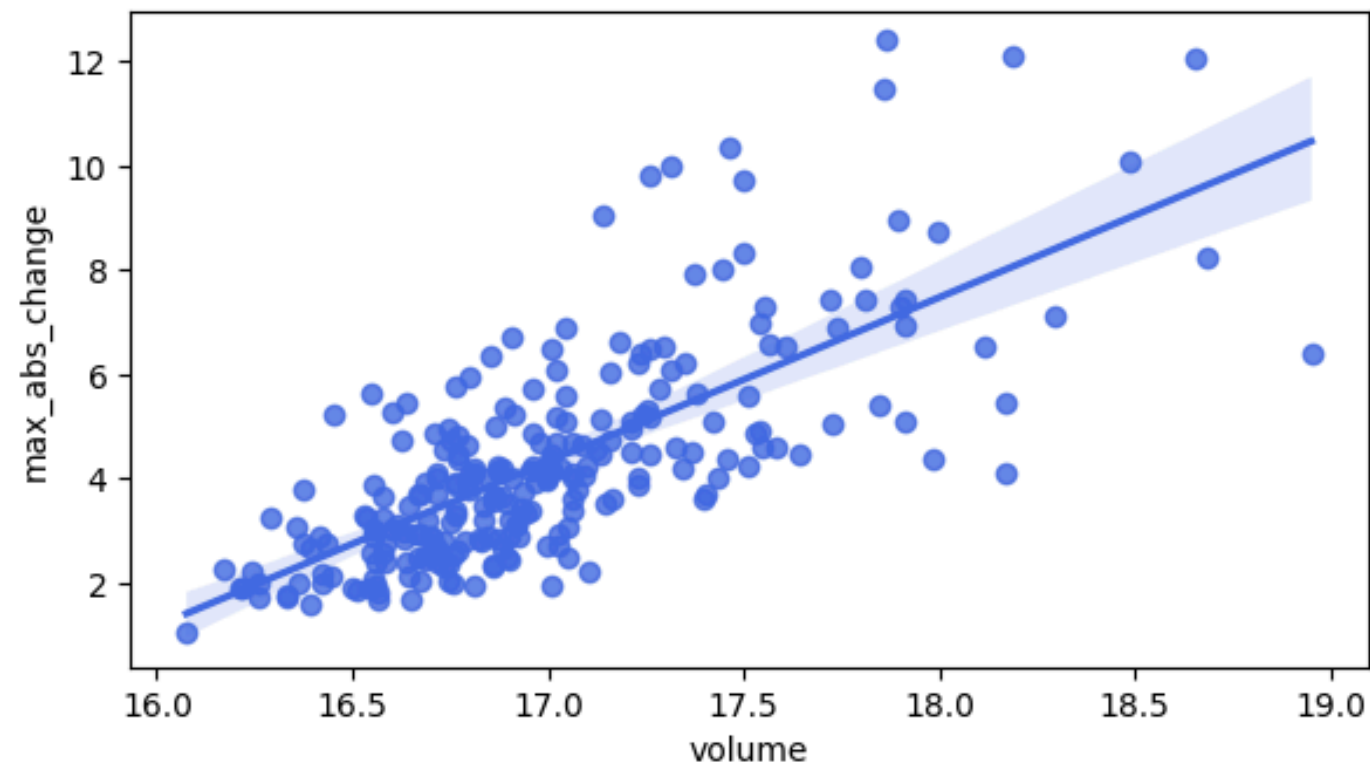
-->I'm an iterable
This prints again because it's an iterable:
-->I'm an iterable

```

```

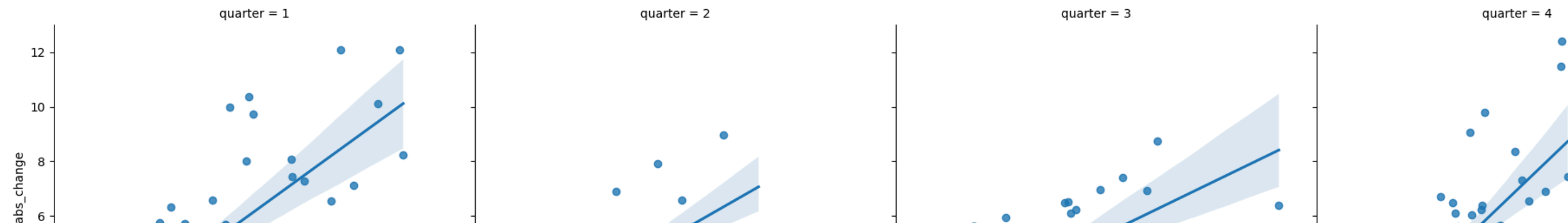
from reg_resid_plot import reg_resid_plots
reg_resid_plots(fb_reg_data)

```



```
sns.lmplot(
    x='volume',
    y='max_abs_change',
    data=fb.assign(
        volume=np.log(fb.volume),
        max_abs_change=fb.high - fb.low,
        quarter=lambda x: x.index.quarter
    ),
    col='quarter'
)
```

<seaborn.axisgrid.FacetGrid at 0x78b4f99cc040>



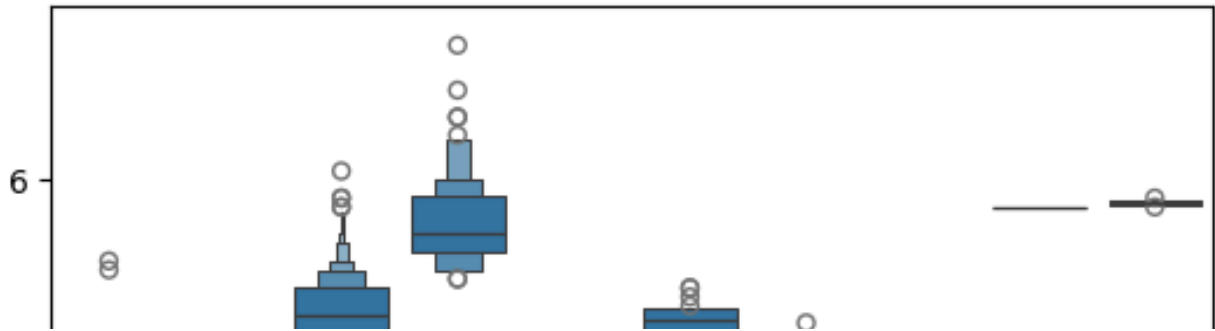
Distributions

✓ boxenplot()

```
sns.boxenplot(
    x='magType', y='mag', data=quakes[['magType', 'mag']]
)
plt.suptitle('Comparing earthquake magnitude by magType')
```

Text(0.5, 0.98, 'Comparing earthquake magnitude by magType')

Comparing earthquake magnitude by magType



▼ violinplot()



```
fig, axes = plt.subplots(figsize=(10, 5))
sns.violinplot(
    x='magType', y='mag', data=quakes[['magType', 'mag']],
    ax=axes, scale='width' # all violins have same width
)
plt.suptitle('Comparing earthquake magnitude by magType')
```

<ipython-input-31-82aafe312639>:2: FutureWarning:

The `scale` parameter has been renamed and will be removed in v0.15.0. Pass `density_norm='width'` for the same effect.

```
sns.violinplot(
Text(0.5, 0.98, 'Comparing earthquake magnitude by magType')
```

Comparing earthquake magnitude by magType