In [4]:
```
import numpy as np
import vectorbt as vbt
import pandas as pd
import yfinance as yf
```

Import Data

In [ ]:
```

```

In [35]:
```
start_date = '2002-01-01'
end_date = '2022-01-01'

oil = yf.download( "CL=F",
            start = start_date, end =end_date,
            interval = "1d")[["Adj Close"]]
price = yf.download( 'GC=F',
            start = start_date, end =end_date,
            interval = "1d")[["Adj Close"]]
```

```
[**********************100%**********************]  1 of 1 completed
[**********************100%**********************]  1 of 1 completed
```

Create Matrix of outputs for parameters

In [36]:
```
windows = np.arange(1, 1001, 10)

fast_avg, slow_avg = vbt.MA.run_combs(oil, window = windows, r = 2,
                                      short_names = ["fast", "slow"])


entries = fast_avg.ma_crossed_above(slow_avg)
exits = fast_avg.ma_crossed_below(slow_avg)

df = pd.concat([price, entries, exits], axis = 1)
df["Adj Close"] = df["Adj Close"].ffill()
df.dropna(inplace = True)
price = df.iloc[:,0]

pf_kwargs = dict(size=np.inf, fees=0.001, freq='1D')
pf = vbt.Portfolio.from_signals(price, entries, exits, **pf_kwargs)
```

In [42]:
```
fig = pf.sharpe_ratio().vbt.heatmap(x_level='fast_window', y_level='slow_window',  symmetric=True,
                                    trace_kwargs=dict(colorbar=dict(title='Total return', tickformat='%')))
fig.show()
```

Get the optimal Portfolio

In [50]:
```
fast_length = 250
slow_length = 350

fast_avg = vbt.MA.run(oil, fast_length, short_name = "MA1")
slow_avg = vbt.MA.run(oil, slow_length, short_name = "MA")

entries = fast_avg.ma_crossed_above(slow_avg)
exits = fast_avg.ma_crossed_below(slow_avg)
```

In [51]:
```
len(oil), len(entries), len(price)
```

Out[51]:
```
(5026, 5026, 5026)
```

In [52]:
```
pf = vbt.Portfolio.from_signals(price, entries, exits)
```

In [ ]:
```

```

In [53]:
```
for col in pf.wrapper.columns:
    pf.plot(column=col).show()
```

In [54]:
```
for col in pf.wrapper.columns:
    pf.plot_underwater(column=col).show()
```

In [ ]:
```

```