

```
In [26]: import yfinance as yf
import datetime
import numpy as np
import matplotlib.pyplot as plt
import hvplot.pandas
import pandas as pd
import quantstats as qs
import talib as ta
```

```
In [27]: df = yf.download("BTC-USD")
```

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

```
In [28]: df.tail(15)
```

```
Out[28]:
```

| | Open | High | Low | Close | Adj Close | Volume |
|-------------------|--------------|--------------|--------------|--------------|--------------|-------------|
| Date | | | | | | |
| 2022-05-26 | 29564.777344 | 29834.160156 | 28261.906250 | 29267.224609 | 29267.224609 | 36774325352 |
| 2022-05-27 | 29251.140625 | 29346.943359 | 28326.613281 | 28627.574219 | 28627.574219 | 36582005748 |
| 2022-05-28 | 28622.625000 | 28814.900391 | 28554.566406 | 28814.900391 | 28814.900391 | 35519577634 |
| 2022-05-29 | 29019.867188 | 29498.009766 | 28841.107422 | 29445.957031 | 29445.957031 | 18093886409 |
| 2022-05-30 | 29443.365234 | 31949.630859 | 29303.572266 | 31726.390625 | 31726.390625 | 39277993274 |
| 2022-05-31 | 31723.865234 | 32249.863281 | 31286.154297 | 31792.310547 | 31792.310547 | 33538210634 |
| 2022-06-01 | 31792.554688 | 31957.285156 | 29501.587891 | 29799.080078 | 29799.080078 | 41135817341 |
| 2022-06-02 | 29794.890625 | 30604.734375 | 29652.705078 | 30467.488281 | 30467.488281 | 29083562061 |
| 2022-06-03 | 30467.806641 | 30633.035156 | 29375.689453 | 29704.390625 | 29704.390625 | 26175547452 |
| 2022-06-04 | 29706.138672 | 29930.564453 | 29500.005859 | 29832.914062 | 29832.914062 | 16588370958 |
| 2022-06-05 | 29835.117188 | 30117.744141 | 29574.449219 | 29906.662109 | 29906.662109 | 17264085441 |
| 2022-06-06 | 29910.283203 | 31693.291016 | 29894.187500 | 31370.671875 | 31370.671875 | 31947336829 |

| | Open | High | Low | Close | Adj Close | Volume |
|------------|--------------|--------------|--------------|--------------|--------------|-------------|
| Date | | | | | | |
| 2022-06-07 | 31371.742188 | 31489.683594 | 29311.683594 | 31155.478516 | 31155.478516 | 40770974039 |
| 2022-06-08 | 31151.480469 | 31253.691406 | 29944.404297 | 30214.355469 | 30214.355469 | 30242059107 |
| 2022-06-09 | 30177.673828 | 30432.619141 | 30088.888672 | 30432.619141 | 30432.619141 | 23963238400 |

In [29]:

```
# Count nulls
df.isna().sum()
```

Out[29]:

```
Open      0
High      0
Low       0
Close     0
Adj Close 0
Volume    0
dtype: int64
```

In [30]:

```
df.head()
```

Out[30]:

| | Open | High | Low | Close | Adj Close | Volume |
|------------|------------|------------|------------|------------|------------|----------|
| Date | | | | | | |
| 2014-09-17 | 465.864014 | 468.174011 | 452.421997 | 457.334015 | 457.334015 | 21056800 |
| 2014-09-18 | 456.859985 | 456.859985 | 413.104004 | 424.440002 | 424.440002 | 34483200 |
| 2014-09-19 | 424.102997 | 427.834991 | 384.532013 | 394.795990 | 394.795990 | 37919700 |
| 2014-09-20 | 394.673004 | 423.295990 | 389.882996 | 408.903992 | 408.903992 | 36863600 |
| 2014-09-21 | 408.084991 | 412.425995 | 393.181000 | 398.821014 | 398.821014 | 26580100 |

In [31]:

```
# Drop coulmns
df.drop(columns=["Adj Close", "Volume"])
```

Out[31]:

| | Open | High | Low | Close |
|--|------|------|-----|-------|
|--|------|------|-----|-------|

| Date | Open | High | Low | Close |
|------------|--------------|--------------|--------------|--------------|
| Date | | | | |
| 2014-09-17 | 465.864014 | 468.174011 | 452.421997 | 457.334015 |
| 2014-09-18 | 456.859985 | 456.859985 | 413.104004 | 424.440002 |
| 2014-09-19 | 424.102997 | 427.834991 | 384.532013 | 394.795990 |
| 2014-09-20 | 394.673004 | 423.295990 | 389.882996 | 408.903992 |
| 2014-09-21 | 408.084991 | 412.425995 | 393.181000 | 398.821014 |
| ... | ... | ... | ... | ... |
| 2022-06-05 | 29835.117188 | 30117.744141 | 29574.449219 | 29906.662109 |
| 2022-06-06 | 29910.283203 | 31693.291016 | 29894.187500 | 31370.671875 |
| 2022-06-07 | 31371.742188 | 31489.683594 | 29311.683594 | 31155.478516 |
| 2022-06-08 | 31151.480469 | 31253.691406 | 29944.404297 | 30214.355469 |
| 2022-06-09 | 30177.673828 | 30432.619141 | 30088.888672 | 30432.619141 |

2823 rows × 4 columns

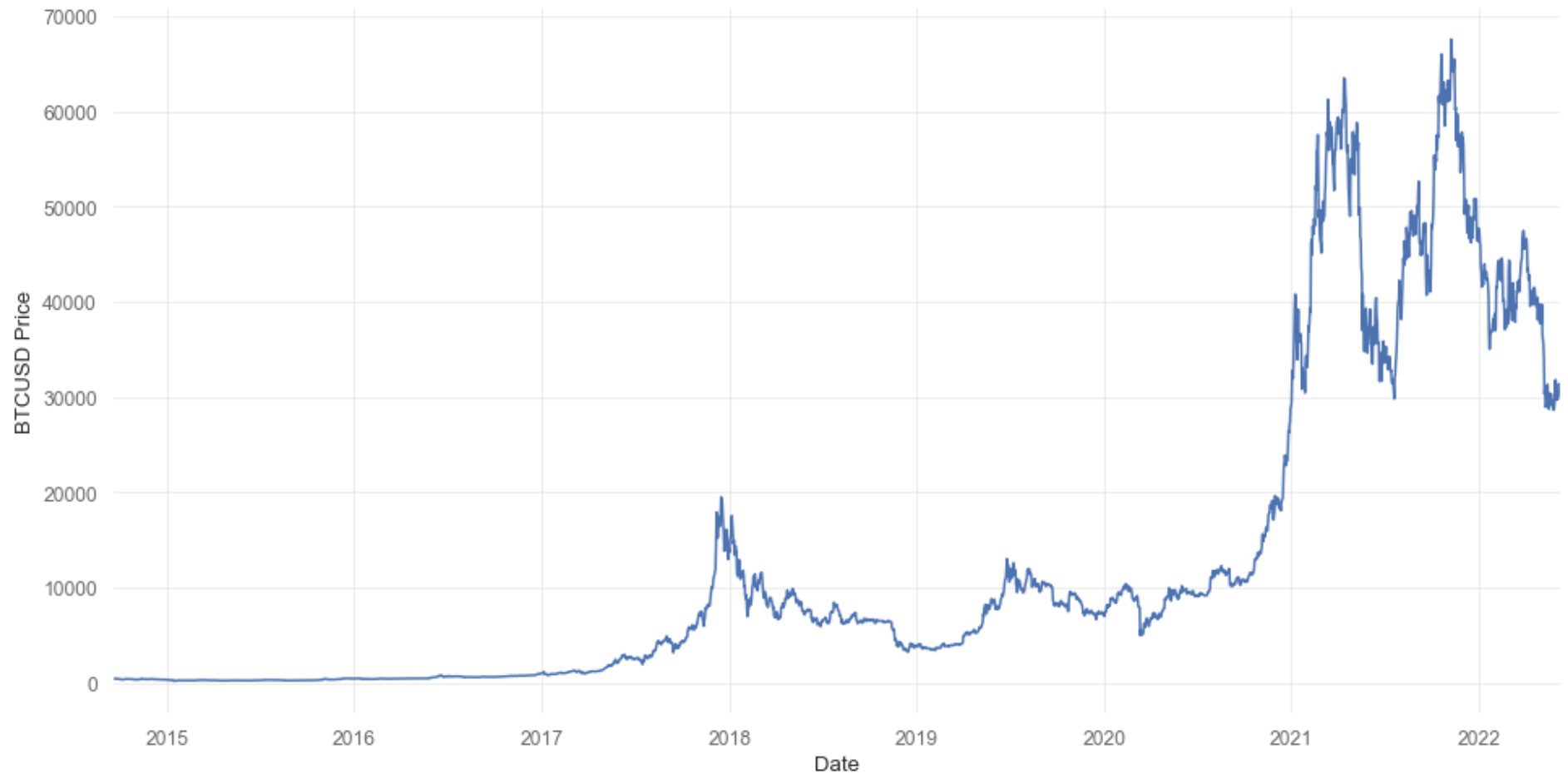
In [32]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2823 entries, 2014-09-17 to 2022-06-09
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Open        2823 non-null   float64
1   High        2823 non-null   float64
2   Low         2823 non-null   float64
3   Close       2823 non-null   float64
4   Adj Close   2823 non-null   float64
5   Volume      2823 non-null   int64
dtypes: float64(5), int64(1)
memory usage: 154.4 KB
```

```
In [33]: # Convert to datetime index  
df.index = pd.to_datetime(df.index)
```

```
In [34]: df.Close.plot(figsize=(16, 8))  
plt.ylabel("BTCUSD Price")  
plt.show()
```



```
In [35]: # ADX settings  
df["ADX"] = ta.ADX(df.High, df.Low, df.Close, 7)
```

```
df["plus_DI"] = ta.PLUS_DI(df.High,df.Low,df.Close,7)
df["minus_DI"] = ta.MINUS_DI(df.High,df.Low,df.Close,7)
```

In [36]:

```
# SAR settings
df["SAR"] = ta.SAR(df['High'].values, df['Low'].values,
                    acceleration=0.02, maximum=0.2)
```

In [37]:

df

Out[37]:

| | Open | High | Low | Close | Adj Close | Volume | ADX | plus_DI | minus_DI | SAR |
|------------|--------------|--------------|--------------|--------------|--------------|-------------|-----------|-----------|-----------|--------------|
| Date | | | | | | | | | | |
| 2014-09-17 | 465.864014 | 468.174011 | 452.421997 | 457.334015 | 457.334015 | 21056800 | NaN | NaN | NaN | NaN |
| 2014-09-18 | 456.859985 | 456.859985 | 413.104004 | 424.440002 | 424.440002 | 34483200 | NaN | NaN | NaN | 468.174011 |
| 2014-09-19 | 424.102997 | 427.834991 | 384.532013 | 394.795990 | 394.795990 | 37919700 | NaN | NaN | NaN | 467.072611 |
| 2014-09-20 | 394.673004 | 423.295990 | 389.882996 | 408.903992 | 408.903992 | 36863600 | NaN | NaN | NaN | 463.770987 |
| 2014-09-21 | 408.084991 | 412.425995 | 393.181000 | 398.821014 | 398.821014 | 26580100 | NaN | NaN | NaN | 460.601428 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2022-06-05 | 29835.117188 | 30117.744141 | 29574.449219 | 29906.662109 | 29906.662109 | 17264085441 | 17.849687 | 22.441956 | 21.737286 | 28997.820798 |
| 2022-06-06 | 29910.283203 | 31693.291016 | 29894.187500 | 31370.671875 | 31370.671875 | 31947336829 | 20.324830 | 35.936536 | 17.233580 | 29192.943347 |
| 2022-06-07 | 31371.742188 | 31489.683594 | 29311.683594 | 31155.478516 | 31155.478516 | 40770974039 | 19.968671 | 27.801182 | 19.386764 | 32249.863281 |
| 2022-06-08 | 31151.480469 | 31253.691406 | 29944.404297 | 30214.355469 | 30214.355469 | 30242059107 | 19.663391 | 23.992010 | 16.730491 | 32191.099687 |
| 2022-06-09 | 30177.673828 | 30432.619141 | 30088.888672 | 30432.619141 | 30432.619141 | 23963238400 | 19.401722 | 23.025713 | 16.056657 | 32133.511366 |

2823 rows × 10 columns

In [38]:

```
# Entry
df["signal"] = np.nan

# SAR
condition_1 = (df['SAR'] < df['Close'])
```

```
# ADX trending
condition_2 = (df.ADX > 25) & (df.plus_DI > df.minus_DI)

# Combine the conditions and store in the signal column 1 when all the conditions are true
df.loc[(condition_1 & condition_2), "signal"] = 1
```

In [39]:

```
# Exit
# ADX trend change
condition_1 = (df.plus_DI < df.minus_DI)

# Combine the conditions and store in the signal column 0 when all the conditions are true
df.loc[condition_1, "signal"] = 0

df.signal.fillna(method="ffill", inplace=True)

df.iloc[320:360, :]
```

Out[39]:

| | Open | High | Low | Close | Adj Close | Volume | ADX | plus_DI | minus_DI | SAR | signal |
|------------|------------|------------|------------|------------|------------|----------|-----------|-----------|-----------|------------|--------|
| Date | | | | | | | | | | | |
| 2015-08-03 | 282.806000 | 285.471008 | 280.233002 | 281.226990 | 281.226990 | 21474100 | 26.165469 | 22.481661 | 25.779983 | 293.023344 | 0.0 |
| 2015-08-04 | 281.225006 | 285.714996 | 281.225006 | 285.217987 | 285.217987 | 21908700 | 23.202770 | 20.674376 | 23.046946 | 292.184130 | 0.0 |
| 2015-08-05 | 284.846985 | 285.501007 | 281.488007 | 281.881989 | 281.881989 | 20128000 | 20.663313 | 18.616448 | 20.752852 | 291.378485 | 0.0 |
| 2015-08-06 | 281.906006 | 281.906006 | 278.403015 | 278.576996 | 278.576996 | 18792100 | 20.984002 | 16.902974 | 26.948538 | 290.605066 | 0.0 |
| 2015-08-07 | 278.740997 | 280.391998 | 276.365997 | 279.584991 | 279.584991 | 42484800 | 22.631632 | 15.046085 | 29.546408 | 289.862583 | 0.0 |
| 2015-08-08 | 279.742004 | 279.928009 | 260.709991 | 260.997009 | 260.997009 | 58533000 | 29.131915 | 9.335000 | 49.253389 | 289.149800 | 0.0 |
| 2015-08-09 | 261.115997 | 267.002991 | 260.467987 | 265.083008 | 265.083008 | 23789600 | 34.746508 | 8.113270 | 43.291952 | 287.443411 | 0.0 |
| 2015-08-10 | 265.477997 | 267.032013 | 262.596008 | 264.470001 | 264.470001 | 20979400 | 39.527317 | 7.412774 | 39.226293 | 285.285377 | 0.0 |
| 2015-08-11 | 264.342010 | 270.385986 | 264.093994 | 270.385986 | 270.385986 | 25433900 | 40.000695 | 13.585478 | 33.950235 | 283.299986 | 0.0 |
| 2015-08-12 | 270.597992 | 270.673004 | 265.468994 | 266.376007 | 266.376007 | 26815400 | 40.103725 | 12.658403 | 30.050137 | 281.473426 | 0.0 |
| 2015-08-13 | 266.183014 | 266.231995 | 262.841003 | 264.079987 | 264.079987 | 27685500 | 41.350445 | 11.602146 | 33.745985 | 279.792991 | 0.0 |
| 2015-08-14 | 264.131989 | 267.466003 | 261.477997 | 265.679993 | 265.679993 | 27091200 | 42.977866 | 9.959749 | 32.191134 | 278.246991 | 0.0 |

| | Open | High | Low | Close | Adj Close | Volume | ADX | plus_DI | minus_DI | SAR | signal |
|------------|------------|------------|------------|------------|------------|----------|-----------|-----------|-----------|------------|--------|
| Date | | | | | | | | | | | |
| 2015-08-15 | 265.528992 | 266.666992 | 261.295990 | 261.550995 | 261.550995 | 19321100 | 44.452253 | 8.674714 | 28.474962 | 276.824670 | 0.0 |
| 2015-08-16 | 261.865997 | 262.440002 | 257.040985 | 258.506989 | 258.506989 | 29717000 | 47.337199 | 7.534635 | 35.090367 | 275.516136 | 0.0 |
| 2015-08-17 | 258.489990 | 260.505005 | 257.117004 | 257.976013 | 257.976013 | 21617900 | 49.810010 | 6.873301 | 32.010398 | 273.668621 | 0.0 |
| 2015-08-18 | 257.925995 | 257.993011 | 211.078995 | 211.078995 | 211.078995 | 42147200 | 55.876949 | 2.842602 | 70.786421 | 272.005857 | 0.0 |
| 2015-08-19 | 225.671005 | 237.408997 | 222.766006 | 226.684006 | 226.684006 | 60869200 | 61.077183 | 2.053931 | 51.146948 | 264.694634 | 0.0 |
| 2015-08-20 | 226.899002 | 237.365005 | 226.899002 | 235.350006 | 235.350006 | 32275000 | 65.534525 | 1.815538 | 45.210496 | 258.260757 | 0.0 |
| 2015-08-21 | 235.354996 | 236.432007 | 231.723999 | 232.569000 | 232.569000 | 23173800 | 69.355105 | 1.713278 | 42.664012 | 252.598945 | 0.0 |
| 2015-08-22 | 232.662003 | 234.957001 | 222.703995 | 230.389999 | 230.389999 | 23205900 | 72.873665 | 1.463061 | 47.184214 | 247.616551 | 0.0 |
| 2015-08-23 | 230.376007 | 232.705002 | 225.580002 | 228.169006 | 228.169006 | 18406600 | 75.889573 | 1.331171 | 42.930720 | 243.232045 | 0.0 |
| 2015-08-24 | 228.112000 | 228.139008 | 210.442993 | 210.494995 | 210.494995 | 59220700 | 78.762893 | 1.055103 | 51.737147 | 239.373679 | 0.0 |
| 2015-08-25 | 210.067993 | 226.320999 | 199.567001 | 221.608994 | 221.608994 | 61089200 | 81.351069 | 0.772867 | 48.771871 | 235.323383 | 0.0 |
| 2015-08-26 | 222.076004 | 231.182999 | 220.203995 | 225.830994 | 225.830994 | 31808000 | 80.680635 | 5.712662 | 43.234894 | 199.567001 | 0.0 |
| 2015-08-27 | 226.050003 | 228.643005 | 223.684006 | 224.768997 | 224.768997 | 21905400 | 80.105977 | 5.390195 | 40.794378 | 199.567001 | 0.0 |
| 2015-08-28 | 224.701004 | 235.218994 | 220.925995 | 231.395996 | 231.395996 | 31336600 | 75.600263 | 11.870066 | 34.286412 | 200.199321 | 0.0 |
| 2015-08-29 | 231.548996 | 233.222000 | 227.330002 | 229.779999 | 229.779999 | 17142500 | 71.738223 | 11.024242 | 31.843270 | 201.600108 | 0.0 |
| 2015-08-30 | 229.895004 | 232.067993 | 226.246994 | 228.761002 | 228.761002 | 19412600 | 68.680807 | 10.187528 | 30.838531 | 202.944864 | 0.0 |
| 2015-08-31 | 229.113998 | 231.955994 | 225.914993 | 230.056000 | 230.056000 | 20710700 | 66.146439 | 9.330145 | 28.705684 | 204.235829 | 0.0 |
| 2015-09-01 | 230.255997 | 231.216003 | 226.860001 | 228.121002 | 228.121002 | 20575200 | 63.974123 | 8.713248 | 26.807703 | 205.475155 | 0.0 |
| 2015-09-02 | 228.026993 | 230.576996 | 226.475006 | 229.283997 | 229.283997 | 18760400 | 62.244123 | 8.123179 | 25.627859 | 206.664909 | 0.0 |
| 2015-09-03 | 229.324005 | 229.604996 | 226.667007 | 227.182999 | 227.182999 | 17482000 | 60.761266 | 7.688122 | 24.255297 | 207.807072 | 0.0 |
| 2015-09-04 | 227.214996 | 230.899994 | 227.050995 | 230.298004 | 230.298004 | 20962400 | 57.768204 | 9.652148 | 22.420026 | 208.903549 | 0.0 |
| 2015-09-05 | 230.199005 | 236.143005 | 229.442993 | 235.018997 | 235.018997 | 20671400 | 49.756372 | 18.789571 | 19.433780 | 209.956167 | 0.0 |
| 2015-09-06 | 234.869995 | 242.912003 | 234.681000 | 239.839996 | 239.839996 | 25473700 | 46.636892 | 28.960403 | 16.318521 | 211.527377 | 1.0 |

| | Open | High | Low | Close | Adj Close | Volume | ADX | plus_DI | minus_DI | SAR | signal |
|------------|------------|------------|------------|------------|------------|----------|-----------|-----------|-----------|------------|--------|
| Date | | | | | | | | | | | |
| 2015-09-07 | 239.934006 | 242.106003 | 238.722000 | 239.847000 | 239.847000 | 21192200 | 43.963052 | 26.892665 | 15.153399 | 214.038147 | 1.0 |
| 2015-09-08 | 239.845993 | 245.781006 | 239.677994 | 243.606995 | 243.606995 | 26879200 | 43.494356 | 31.244962 | 13.174256 | 216.348056 | 1.0 |
| 2015-09-09 | 243.414993 | 244.416000 | 237.820999 | 238.167999 | 238.167999 | 23635700 | 41.193146 | 26.827570 | 15.292592 | 219.291351 | 1.0 |
| 2015-09-10 | 238.335999 | 241.292999 | 235.791000 | 238.477005 | 238.477005 | 21215500 | 37.263096 | 23.582472 | 17.905727 | 221.940316 | 1.0 |
| 2015-09-11 | 238.328995 | 241.169006 | 238.328995 | 240.106995 | 240.106995 | 19224700 | 33.894481 | 21.981272 | 16.689966 | 224.324385 | 1.0 |

```
In [40]: df["signal_change"] = df.signal.diff()
df["signal_change"].value_counts()
```

```
Out[40]: 0.0    2641
1.0      87
-1.0     87
Name: signal_change, dtype: int64
```

```
In [41]: # Visualize entry position relative to close price
entry = df[df["signal_change"] == 1.0]["Close"].hvplot.scatter(
    color="green",
    marker="^",
    size=200,
    legend=False,
    ylabel="Price in $",
    width=1000,
    height=400
)

# Visualize exit position relative to close price
exit = df[df["signal_change"] == -1.0]["Close"].hvplot.scatter(
    color="blue",
    marker="v",
    size=200,
    legend=False,
    ylabel="Price in $",
    width=1000,
    height=400
)
```



```
# Visualize close price for the investment
security_close = df[["Close"]].hvplot(
    line_color="lightgray",
    ylabel="Price in $",
    width=1000,
    height=400
)

# Plot indicators
ichi = df[["Close", "SAR"]].hvplot(
    ylabel="Price in $",
    width=1000,
    height=400
)

# # Overlay plots
saradxplot = security_close * ichi * entry * exit
saradxplot
```

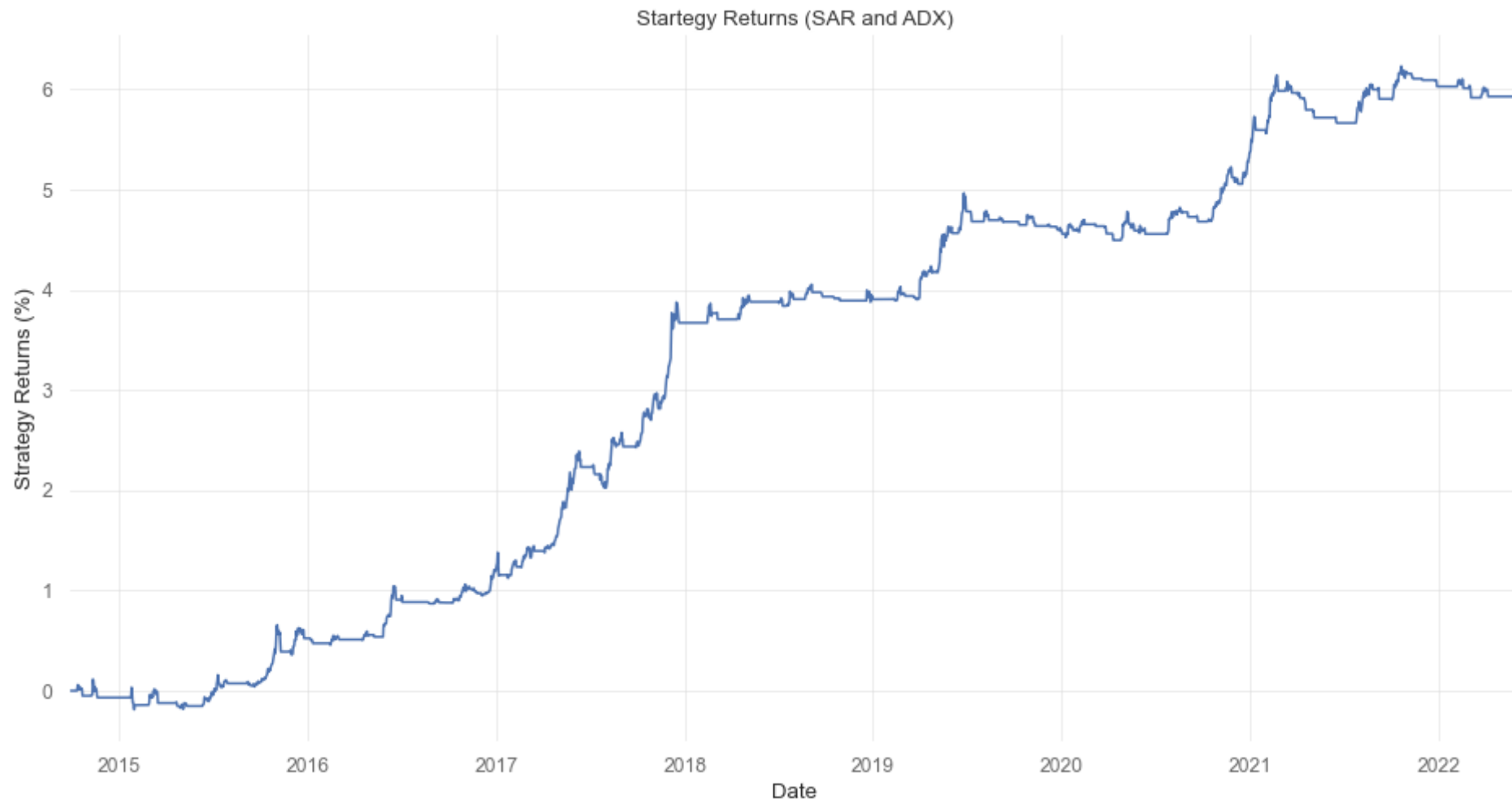
Out[41]:

In [42]:

```
# Calculate daily returns
daily_returns = df.Close.pct_change()

# Calculate strategy returns
strategy_returns = daily_returns * df.signal.shift(1)
strategy_returns.dropna(inplace=True)

# Plot strategy returns
strategy_returns.cumsum().plot(figsize=(16, 8))
plt.xlabel("Date")
plt.ylabel("Strategy Returns (%)")
plt.title("Strategy Returns (SAR and ADX)")
plt.show()
```



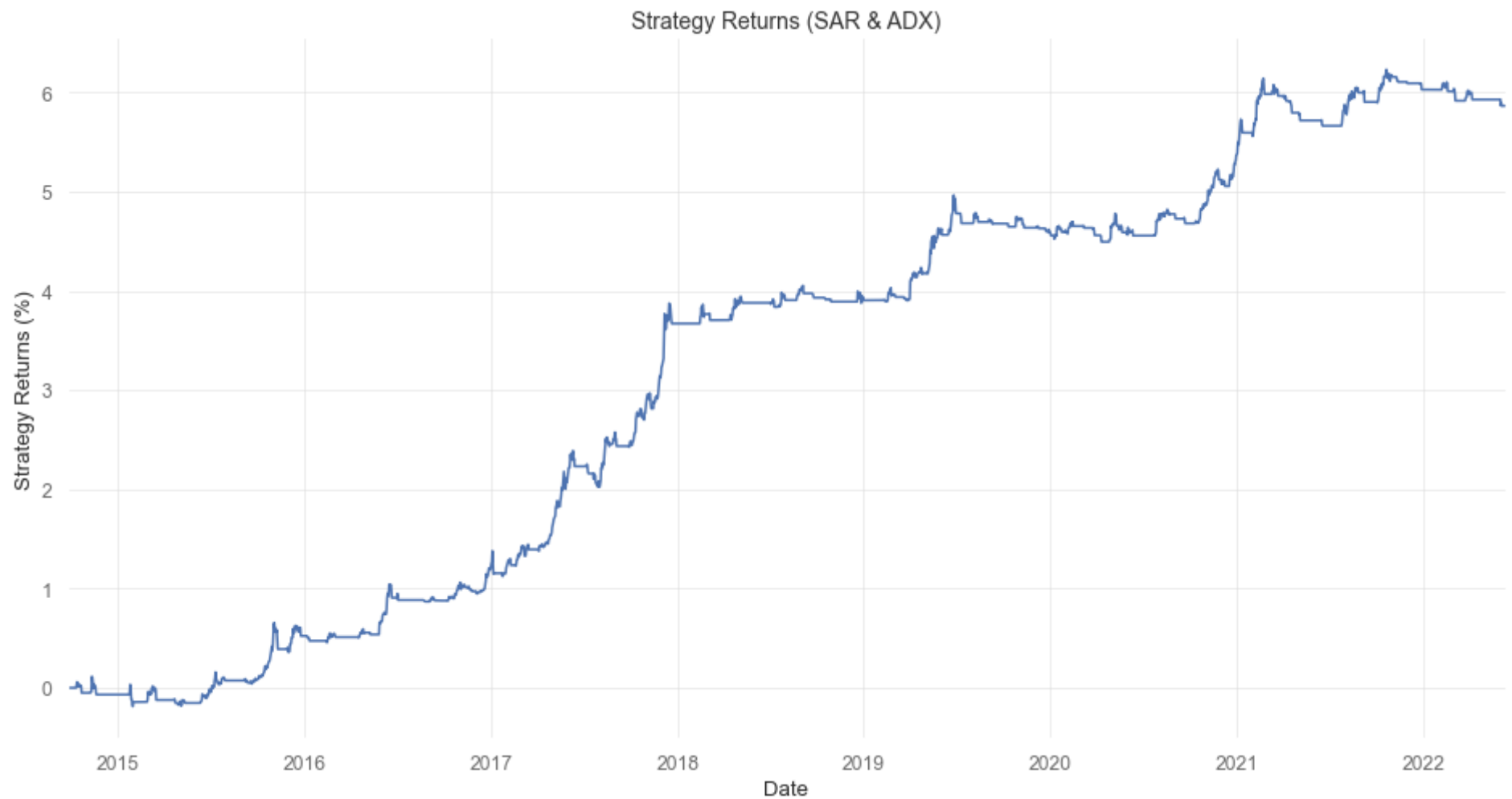
In [43]:

```
# Calculate daily returns
daily_returns = df.Close.pct_change()

# Calculate strategy returns
strategy_returns = daily_returns * df.signal.shift(1)
strategy_returns.dropna(inplace=True)

# Plot strategy returns
strategy_returns.cumsum().plot(figsize=(16, 8))
plt.xlabel("Date")
plt.ylabel("Strategy Returns (%)")
```

```
plt.title("Strategy Returns (SAR & ADX)", fontsize=14)  
plt.show()
```



In [44]:

```
# Check Sharpe ratio calculation  
def annualized_sharpe_ratio(returns, N=252):  
    return ((N) * returns.mean()) / (returns.std() * np.sqrt(N))  
  
# Sharpe ratio  
excess_daily_strategy_return = strategy_returns  
sharpe = annualized_sharpe_ratio(excess_daily_strategy_return)  
print("The Sharpe ratio of strategy is %.2f" % sharpe)
```

The Sharpe ratio of strategy is 1.37

In [45]:

```
# Calculate the cumulative returns
df["cumulative_returns"] = (strategy_returns+1).cumprod()

# Plot the cumulative returns
plt.figure(figsize=(16, 8))
plt.plot(df["cumulative_returns"])
plt.title("Cumaltive Returns (SAR & ADX)", fontsize=14)
plt.xlabel("Date")
plt.ylabel("Returns (%)")
plt.show()
```



```
In [46]: # strategy_returns.value_counts()
```

```
In [47]: # Calculate the running maximum
running_max = np.maximum.accumulate(df["cumulative_returns"].dropna())
# Ensure the value never drops below 1
running_max[running_max < 1] = 1
# Calculate the percentage drawdown
drawdown = ((df["cumulative_returns"])/running_max - 1) * 100

# Calculate the maximum drawdown
```

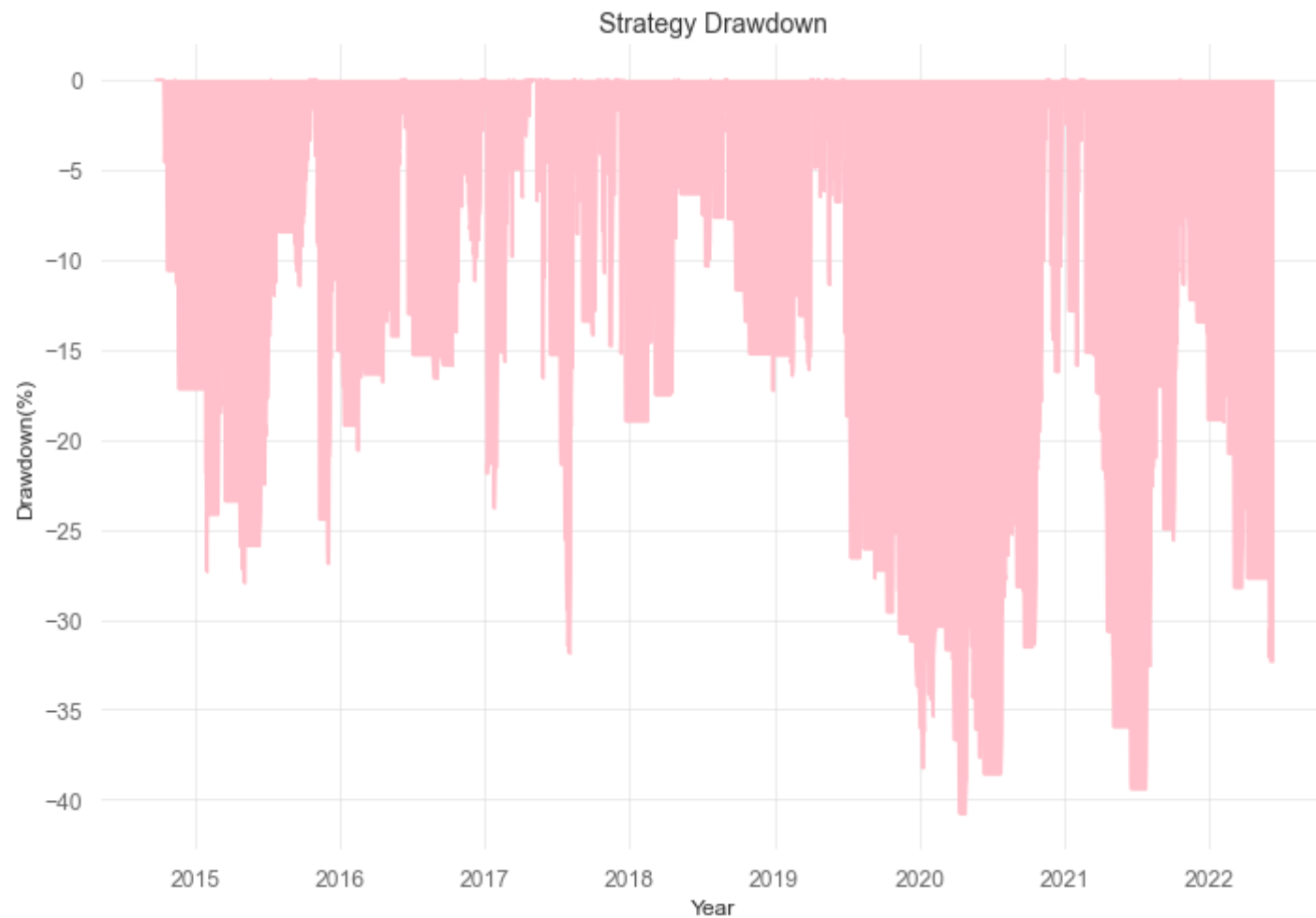
```
print("Maximum drawdown of the strategy is {:.2f}%".format(drawdown.min()))

fig = plt.figure(figsize=(10, 7))

# Plot max drawdown
plt.plot(drawdown, color="pink")
# Fill in-between the drawdown
plt.fill_between(drawdown.index, drawdown.values, color="pink")
plt.title("Strategy Drawdown", fontsize=14)
plt.ylabel("Drawdown(%)", fontsize=12)
plt.xlabel("Year", fontsize=12)

plt.tight_layout()
plt.show()
```

Maximum drawdown of the strategy is -40.77%



```
In [48]: # Extend pandas functionality with metrics  
qs.extend_pandas()
```

```
In [49]: # View basic performance metrics  
qs.reports.basic(strategy_returns)
```

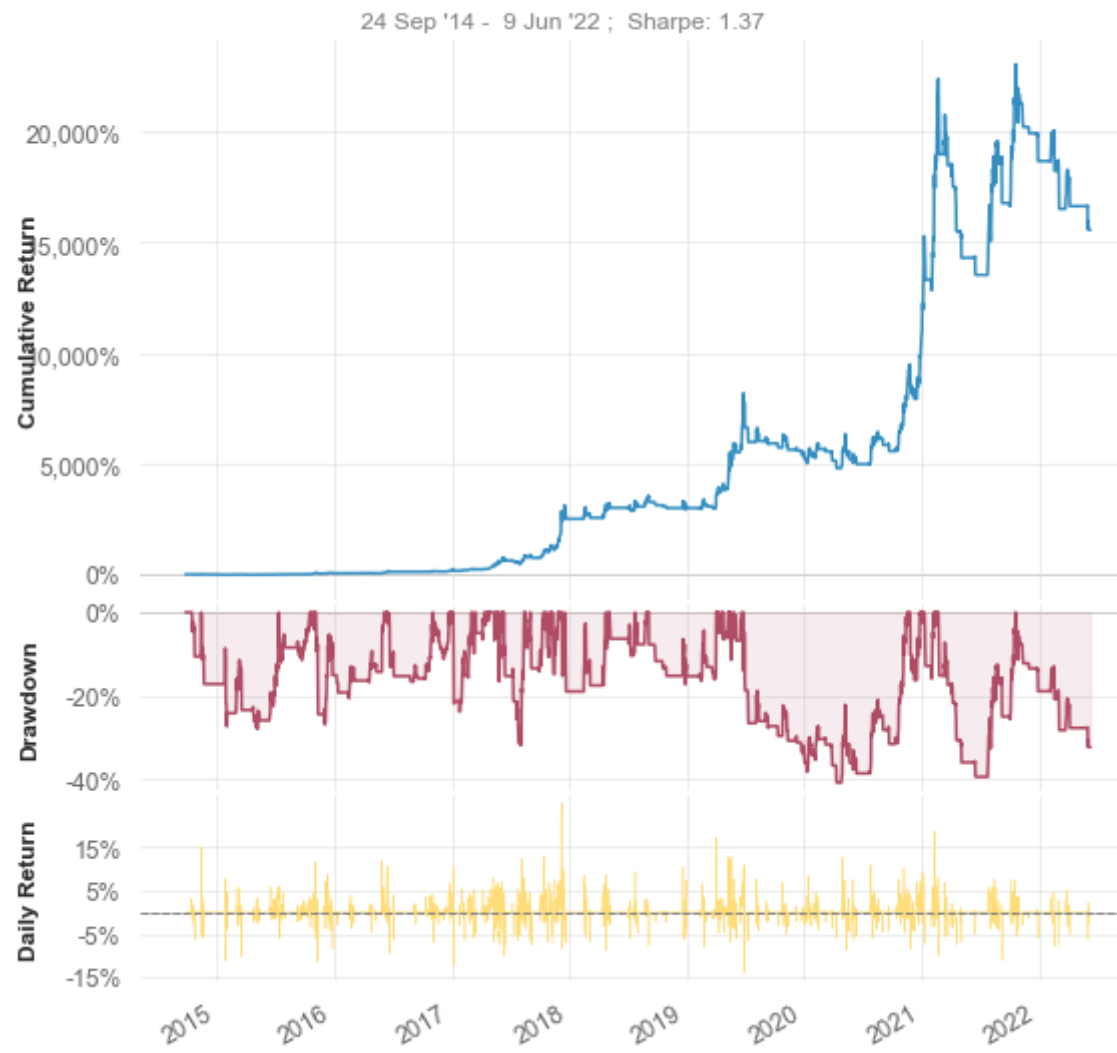
Performance Metrics

Strategy

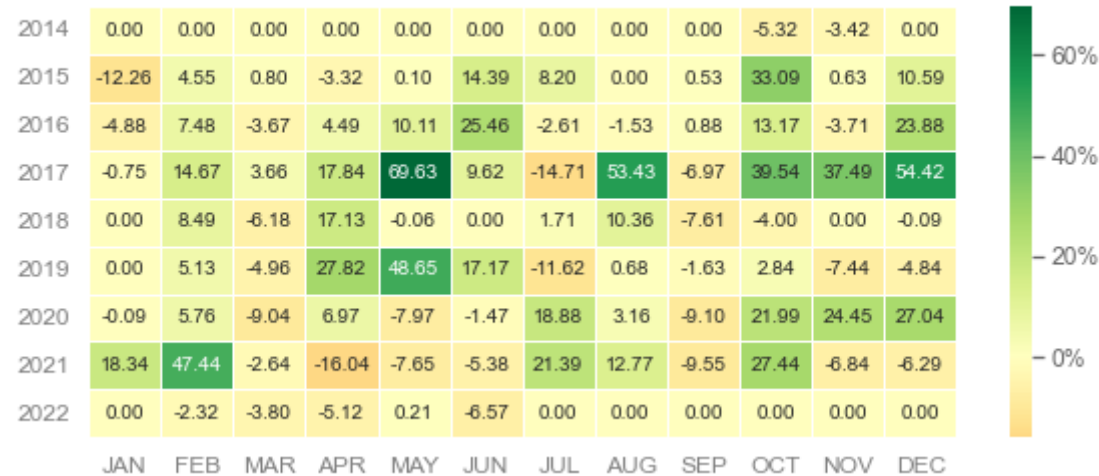
| | |
|--------------------|------------|
| Start Period | 2014-09-25 |
| End Period | 2022-06-09 |
| Risk-Free Rate | 0.0% |
| Time in Market | 44.0% |
| Cumulative Return | 15,573.42% |
| CAGR % | 92.63% |
| Sharpe | 1.37 |
| Prob. Sharpe Ratio | 100.0% |
| Sortino | 2.28 |
| Sortino/√2 | 1.61 |
| Omega | 1.47 |
| Max Drawdown | -40.77% |
| Longest DD Days | 508 |
| Gain/Pain Ratio | 0.47 |
| Gain/Pain (1M) | 2.85 |
| Payoff Ratio | 1.13 |
| Profit Factor | 1.47 |
| Common Sense Ratio | 2.0 |
| CPC Index | 0.94 |
| Tail Ratio | 1.36 |
| Outlier Win Ratio | 10.66 |
| Outlier Loss Ratio | 3.03 |
| MTD | -6.57% |
| 3M | -5.69% |
| 6M | -21.78% |
| YTD | -16.53% |
| 1Y | 8.79% |
| 3Y (ann.) | 33.55% |
| 5Y (ann.) | 86.04% |
| 10Y (ann.) | 92.63% |
| All-time (ann.) | 92.63% |
| Avg. Drawdown | -8.97% |
| Avg. Drawdown Days | 42 |
| Recovery Factor | 382.02 |
| Ulcer Index | 0.2 |
| Serenity Index | 55.38 |

Strategy Visualization

Portfolio Summary



Monthly Returns (%)



```
In [50]: # View basic performance metrics
         qs.reports.full(strategy_returns, "BTC-USD")
```

Performance Metrics

| | Strategy | Benchmark |
|--------------------|------------|------------|
| ----- | ----- | ----- |
| Start Period | 2014-09-25 | 2014-09-25 |
| End Period | 2022-06-09 | 2022-06-09 |
| Risk-Free Rate | 0.0% | 0.0% |
| Time in Market | 44.0% | 100.0% |
| Cumulative Return | 15,573.42% | 7,294.20% |
| CAGR % | 92.63% | 74.75% |
| Sharpe | 1.37 | 0.94 |
| Prob. Sharpe Ratio | 100.0% | 99.91% |
| Smart Sharpe | 1.27 | 0.87 |
| Sortino | 2.28 | 1.38 |
| Smart Sortino | 2.12 | 1.28 |
| Sortino/√2 | 1.61 | 0.97 |
| Smart Sortino/√2 | 1.5 | 0.9 |
| Omega | 1.47 | 1.47 |

| | | |
|---------------------------|---------|---------|
| Max Drawdown | -40.77% | -83.4% |
| Longest DD Days | 508 | 1079 |
| Volatility (ann.) | 38.32% | 61.4% |
| R^2 | 0.39 | 0.39 |
| Information Ratio | -0.01 | -0.01 |
| Calmar | 2.27 | 0.9 |
| Skew | 1.29 | -0.14 |
| Kurtosis | 14.24 | 7.22 |
| Expected Daily % | 0.18% | 0.15% |
| Expected Monthly % | 5.52% | 4.68% |
| Expected Yearly % | 75.35% | 61.31% |
| Kelly Criterion | 18.12% | 13.54% |
| Risk of Ruin | 0.0% | 0.0% |
| Daily Value-at-Risk | -3.76% | -6.13% |
| Expected Shortfall (cVaR) | -3.76% | -6.13% |
| Max Consecutive Wins | 13 | 13 |
| Max Consecutive Losses | 5 | 7 |
| Gain/Pain Ratio | 0.47 | 0.2 |
| Gain/Pain (1M) | 2.85 | 1.3 |
| Payoff Ratio | 1.13 | 1.13 |
| Profit Factor | 1.47 | 1.2 |
| Common Sense Ratio | 2.0 | 1.26 |
| CPC Index | 0.94 | 0.73 |
| Tail Ratio | 1.36 | 1.05 |
| Outlier Win Ratio | 12.44 | 3.88 |
| Outlier Loss Ratio | 3.79 | 3.46 |
| MTD | -6.57% | -4.28% |
| 3M | -5.69% | -21.44% |
| 6M | -21.78% | -39.74% |
| YTD | -16.53% | -34.28% |
| 1Y | 8.79% | -9.08% |
| 3Y (ann.) | 33.55% | 36.42% |
| 5Y (ann.) | 86.04% | 65.23% |
| 10Y (ann.) | 92.63% | 74.75% |
| All-time (ann.) | 92.63% | 74.75% |
| Best Day | 25.25% | 25.25% |
| Worst Day | -14.09% | -37.17% |
| Best Month | 69.63% | 69.63% |
| Worst Month | -16.04% | -36.41% |

| | | |
|--------------------|-----------|---------|
| Best Year | 832.48% | 1368.9% |
| Worst Year | -16.53% | -73.56% |
| Avg. Drawdown | -8.97% | -11.8% |
| Avg. Drawdown Days | 42 | 46 |
| Recovery Factor | 382.02 | 87.46 |
| Ulcer Index | 0.2 | 0.42 |
| Serenity Index | 55.38 | 9.12 |
| Avg. Up Month | 19.83% | 25.92% |
| Avg. Down Month | -5.97% | -12.02% |
| Win Days % | 56.54% | 54.11% |
| Win Month % | 55.81% | 54.26% |
| Win Quarter % | 58.06% | 53.12% |
| Win Year % | 77.78% | 66.67% |
| Beta | 0.39 | - |
| Alpha | 0.3 | - |
| Correlation | 62.37% | - |
| Treynor Ratio | 40008.17% | - |
| None | | |

5 Worst Drawdowns

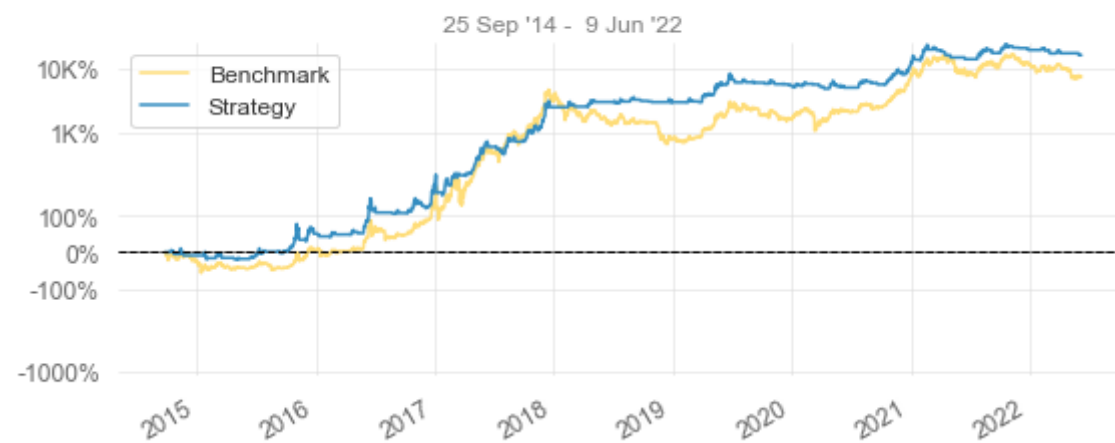
| | Start | Valley | End | Days | Max Drawdown | 99% Max Drawdown |
|---|------------|------------|------------|------|--------------|------------------|
| 1 | 2019-06-27 | 2020-04-11 | 2020-11-16 | 508 | -40.766000 | -40.710642 |
| 2 | 2021-02-22 | 2021-06-17 | 2021-10-19 | 239 | -39.391593 | -38.924037 |
| 3 | 2021-10-21 | 2022-06-03 | 2022-06-09 | 231 | -32.293601 | -32.077771 |
| 4 | 2017-06-12 | 2017-08-02 | 2017-08-13 | 62 | -31.870229 | -31.679460 |
| 5 | 2014-11-13 | 2015-05-06 | 2015-07-12 | 241 | -27.947153 | -26.332894 |

Strategy Visualization

Cumulative Returns vs Benchmark



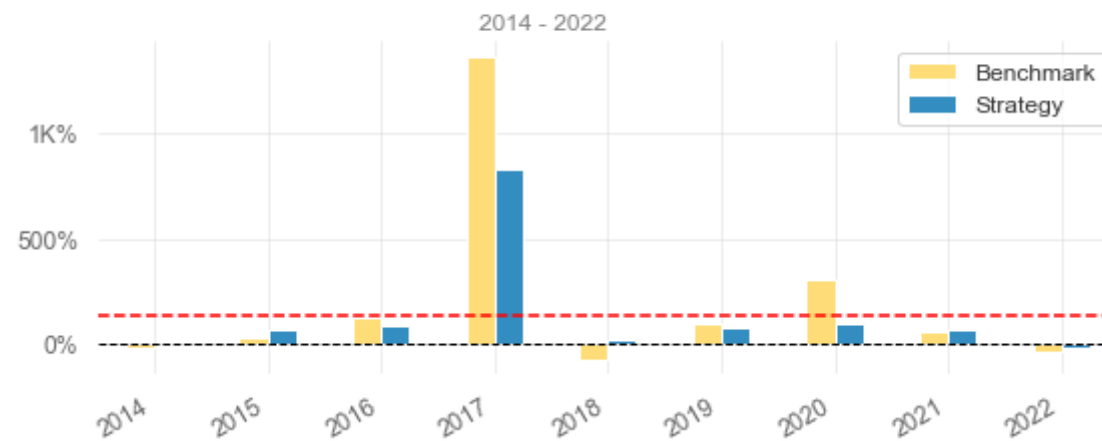
Cumulative Returns vs Benchmark (Log Scaled)



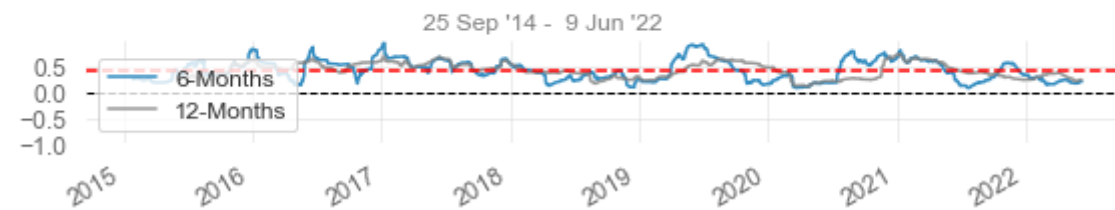
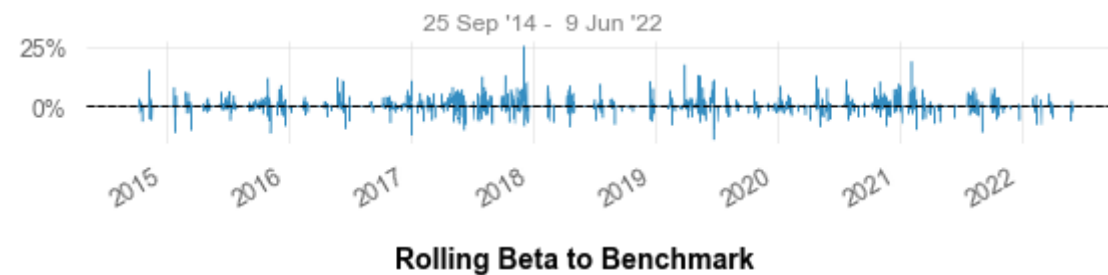
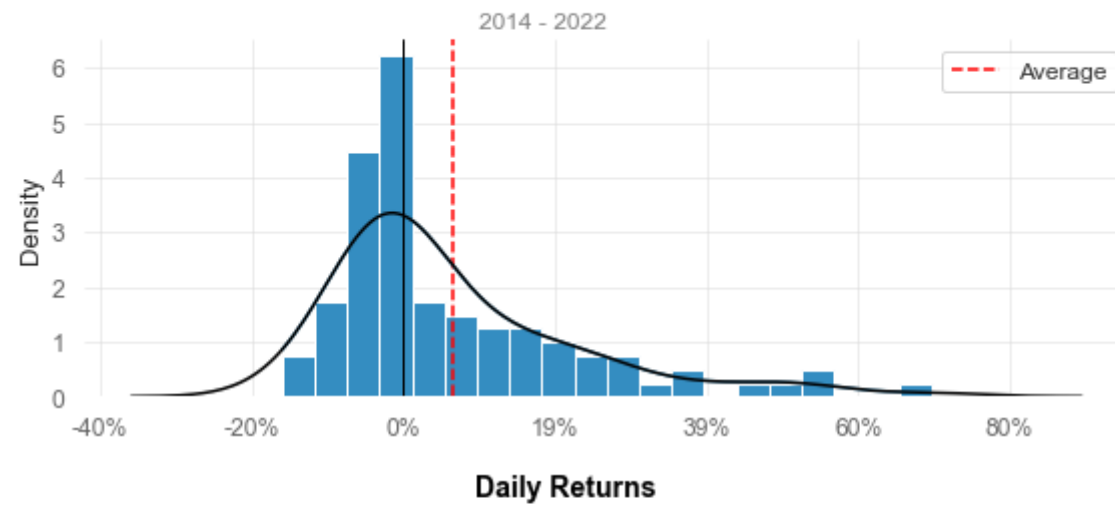
Cumulative Returns vs Benchmark (Volatility Matched)



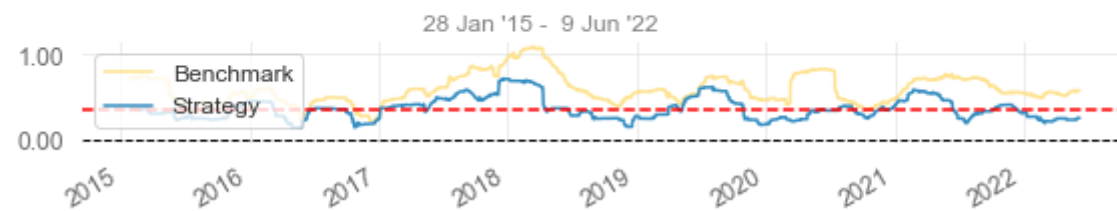
EOY Returns vs Benchmark



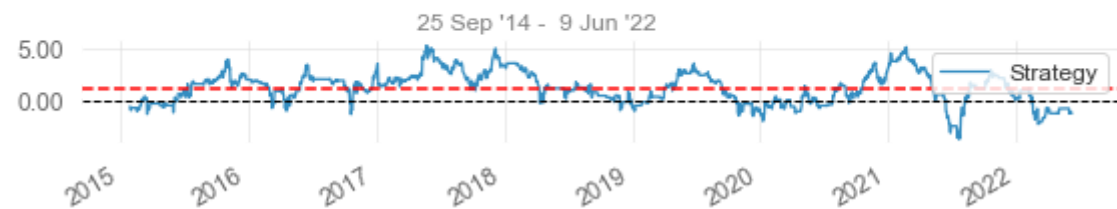
Distribution of Monthly Returns



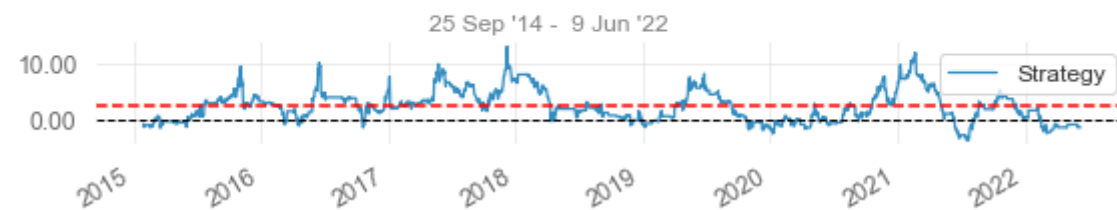
Rolling Volatility (6-Months)



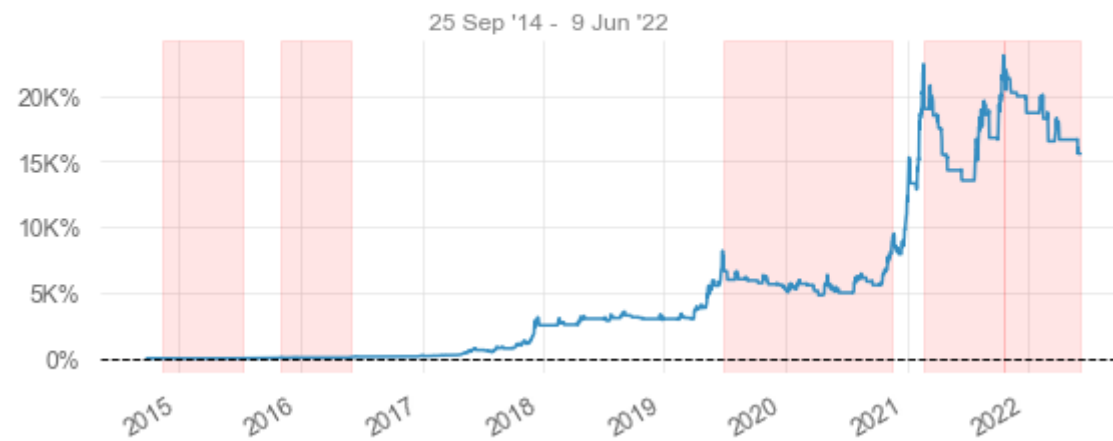
Rolling Sharpe (6-Months)



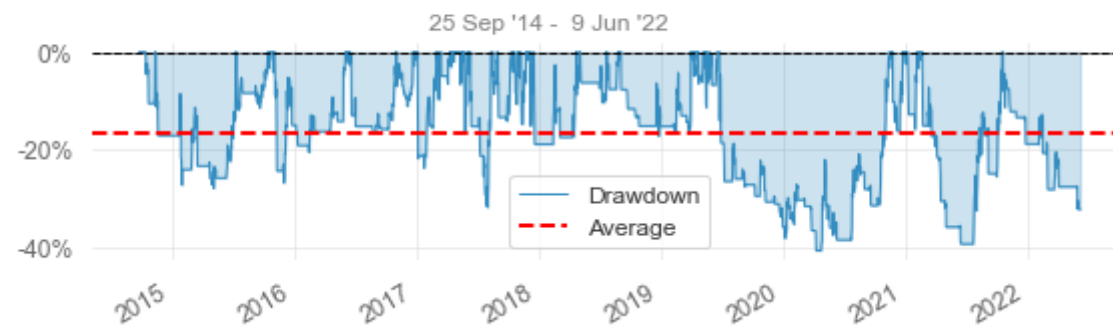
Rolling Sortino (6-Months)



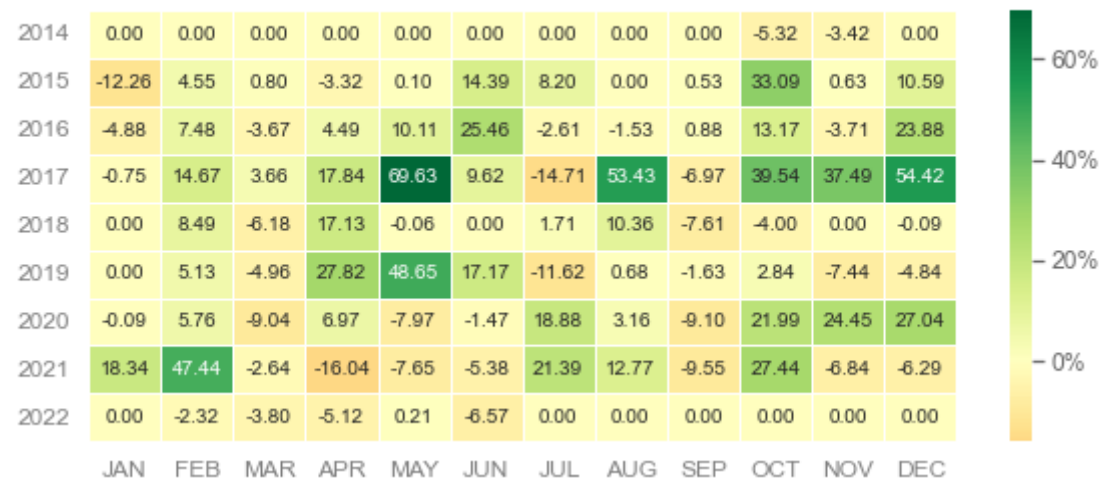
Worst 5 Drawdown Periods



Underwater Plot



Monthly Returns (%)



Return Quantiles

