

## Backtest RSI Score Trend

```
In [1]: from datetime import datetime
from IPython.core.display import display, HTML

from glob import glob
import os
from pathlib import Path
```

```
/tmp/ipykernel_2852054/2186252158.py:2: DeprecationWarning: Importing display from IPython.core.display is deprecated since IPython 7.14, please import from IPython display
    from IPython.core.display import display, HTML
```

```
In [2]: rsi_dates_str = """
rsi_date = 20240729
rsi_date = 20230731
rsi_date = 20220729
rsi_date = 20210730
rsi_date = 20200731
rsi_date = 20190731
rsi_date = 20180731
rsi_date = 20170731
rsi_date = 20160729
rsi_date = 20150731
rsi_date = 20140731
rsi_date = 20130731
rsi_date = 20120731
rsi_date = 20110729
rsi_date = 20100730
rsi_date = 20090731
rsi_date = 20080731
rsi_date = 20070731
rsi_date = 20060731
rsi_date = 20050729
rsi_date = 20060731
rsi_date = 20060131
rsi_date = 20060131
rsi_date = 20070131
rsi_date = 20080131
rsi_date = 20090130
rsi_date = 20100129
rsi_date = 20110131
rsi_date = 20120131
rsi_date = 20130131
rsi_date = 20140131
rsi_date = 20150130
rsi_date = 20160129
rsi_date = 20170131
rsi_date = 20180131
rsi_date = 20190131
rsi_date = 20200131
rsi_date = 20210129
rsi_date = 20220131
```

```
rsi_date = 20230131
rsi_date = 20240131
rsi_date = 20240729
""".replace("rsi_date = ","")
```

In [11]: all\_dates = sorted(list(set([i.strip() for i in rsi\_dates\_str.split("\n") if

In [12]: print(all\_dates)

```
['20060131', '20060731', '20070131', '20070731', '20080131', '20080731', '20090130', '20090731', '20100129', '20100730', '20110131', '20110729', '20120131', '20120731', '20130131', '20130731', '20140131', '20140731', '20150130', '20150731', '20160129', '20160729', '20170131', '20170731', '20180131', '20180731', '20190131', '20190731', '20200131', '20200731', '20210129', '20210730', '20220131', '20220729', '20230131', '20230731', '20240131', '20240729']
```

all\_files = glob("images/20\*") files = [dt.split("\\\\")[-1] for dt in all\_files if datetime.fromtimestamp(os.path.getmtime(dt)).date() == datetime.now().date()]

```
In [13]: def _get_img_path(dt, nday=250):
    #     dt = dt.replace("/", "")
    curr_dir = "." # os.getcwd()
    trend_img = f"images/{dt}/Trend/RSI_Score_Trend-{nday}-v4.png"
    spy_img = f"images/{dt}/Trend/SPY-{nday}.png"
    qqq_img = f"images/{dt}/Trend/QQQ-{nday}.png"
    iwm_img = f"images/{dt}/Trend/IWM-{nday}.png"
    return [os.path.join(curr_dir,trend_img),
            os.path.join(curr_dir,spy_img),
            os.path.join(curr_dir,qqq_img),
            os.path.join(curr_dir,iwm_img),]

def trow(dt, width=800, height=1000):
    ret_err = []
    trend_img, spy_img, qqq_img, iwm_img = _get_img_path(dt)
    ret = ""

    for img in _get_img_path(dt):

        if Path(img).exists():
            ret += f"""
<tr>
<img src='{img}' width={width} height={height} />
</tr>"""
        else:
            ret_err.append(img)

    return ret, ret_err
```

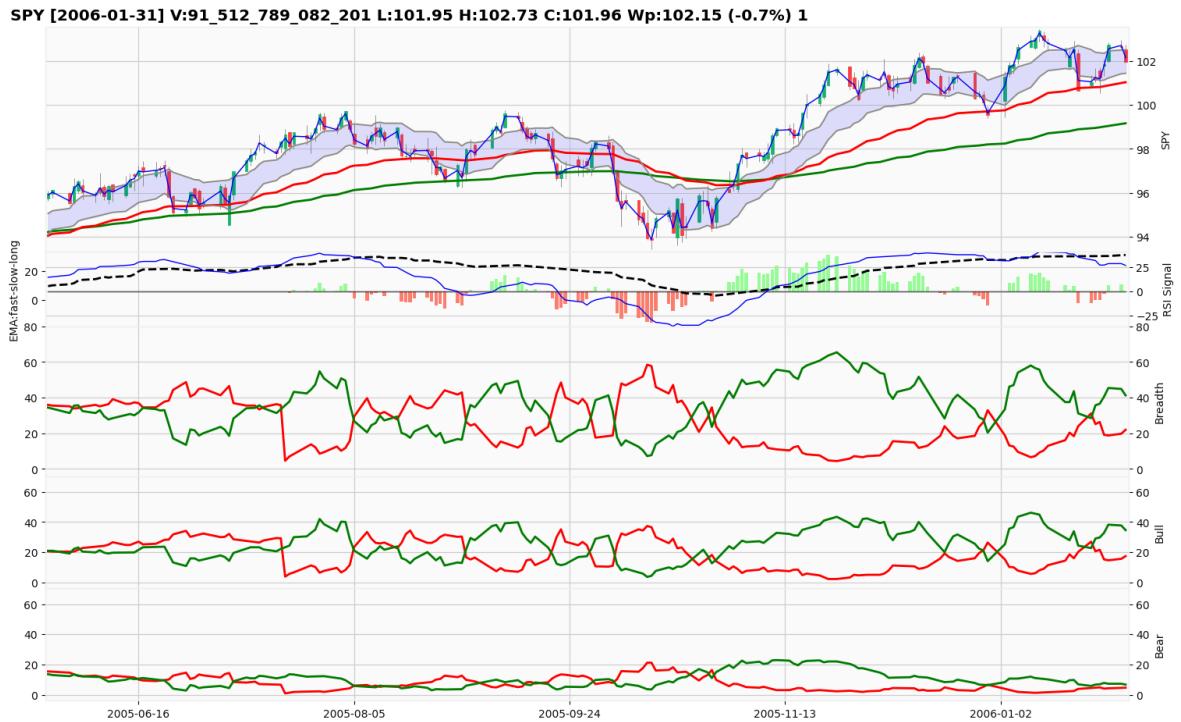
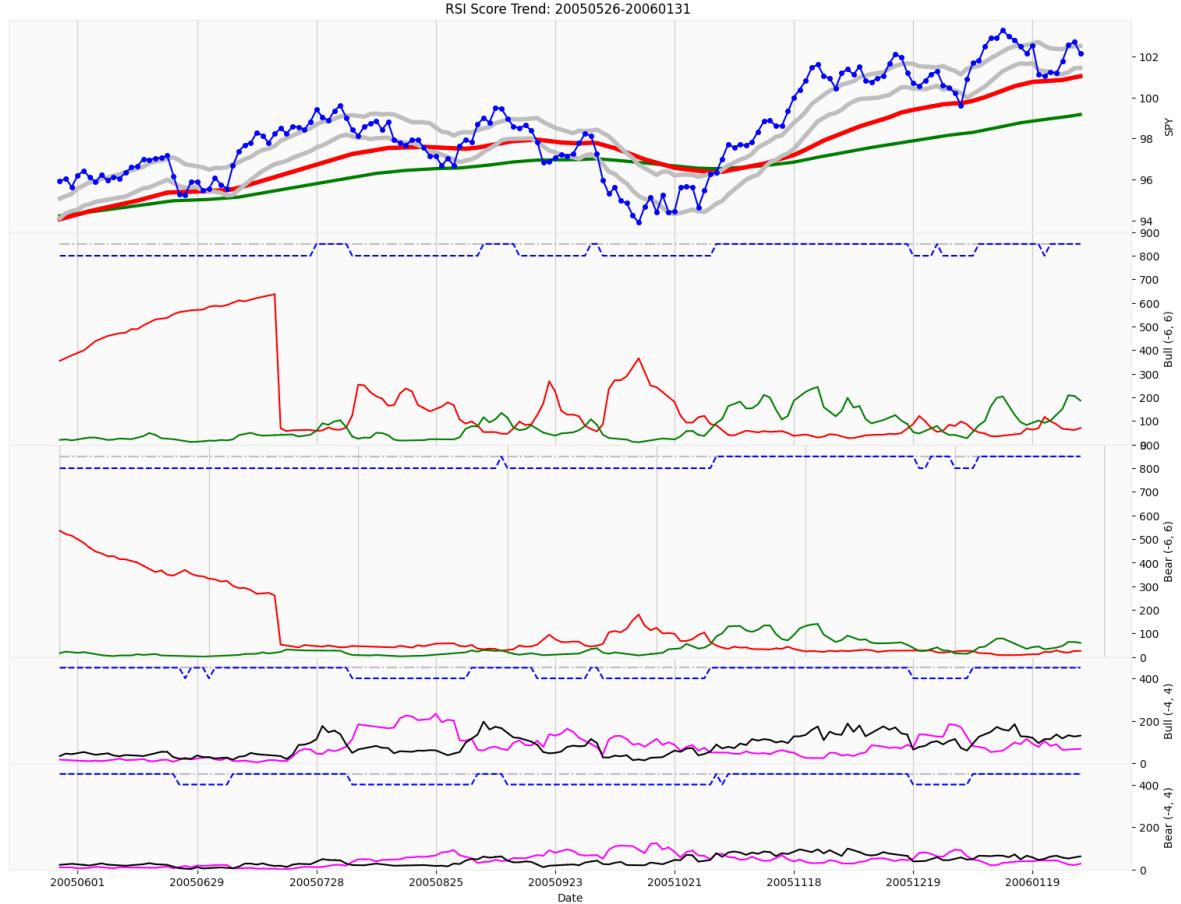
In [14]: dt = "20240729"
print(\_get\_img\_path(dt))

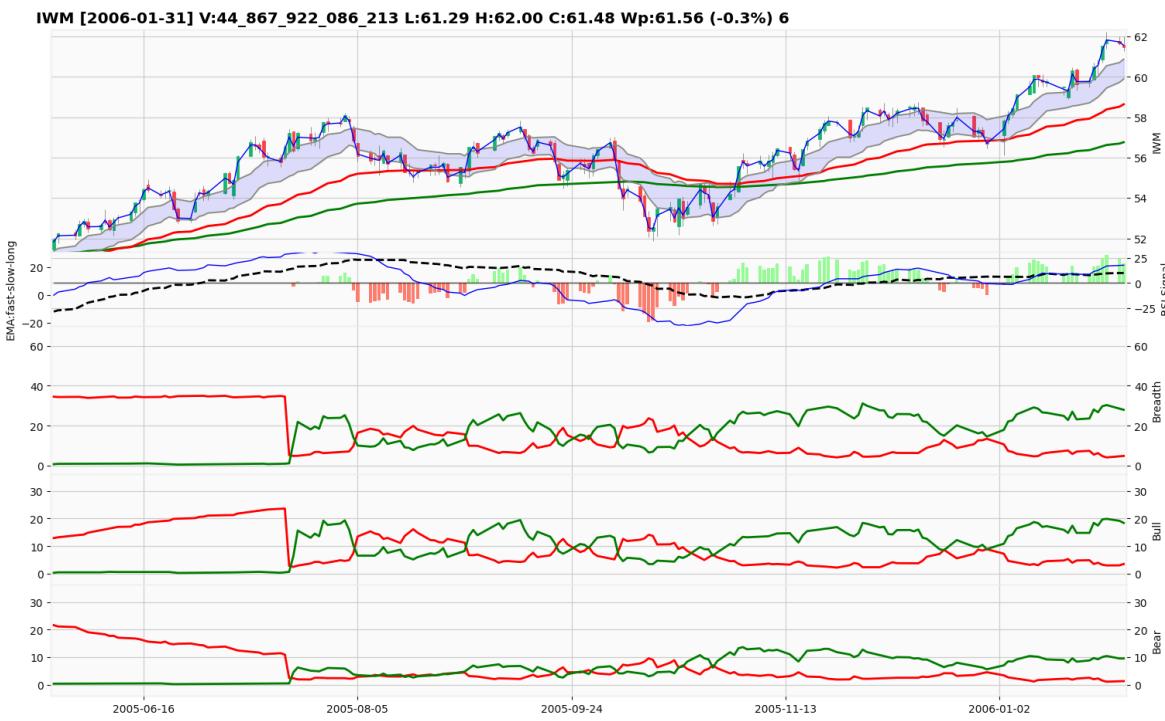
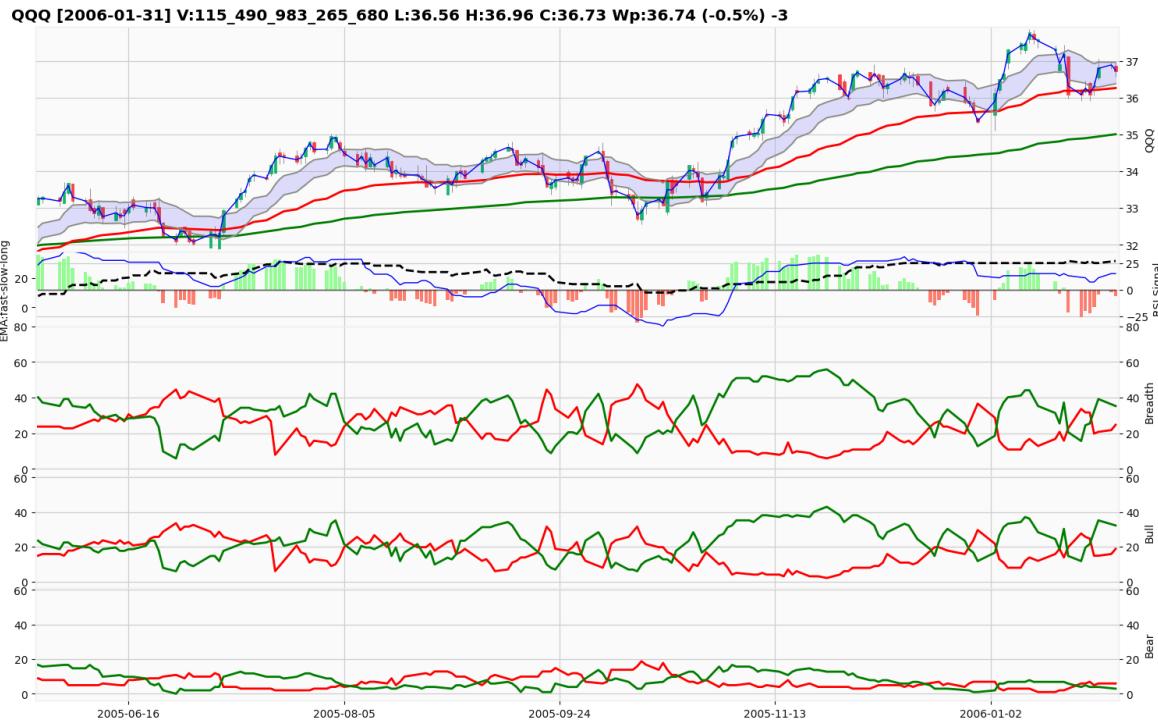
```
 ['./images/20240729/Trend/RSI_Score_Trend-250-v4.png', './images/20240729/Trend/SPY-250.png', './images/20240729/Trend/QQQ-250.png', './images/20240729/Trend/IWM-250.png']
```

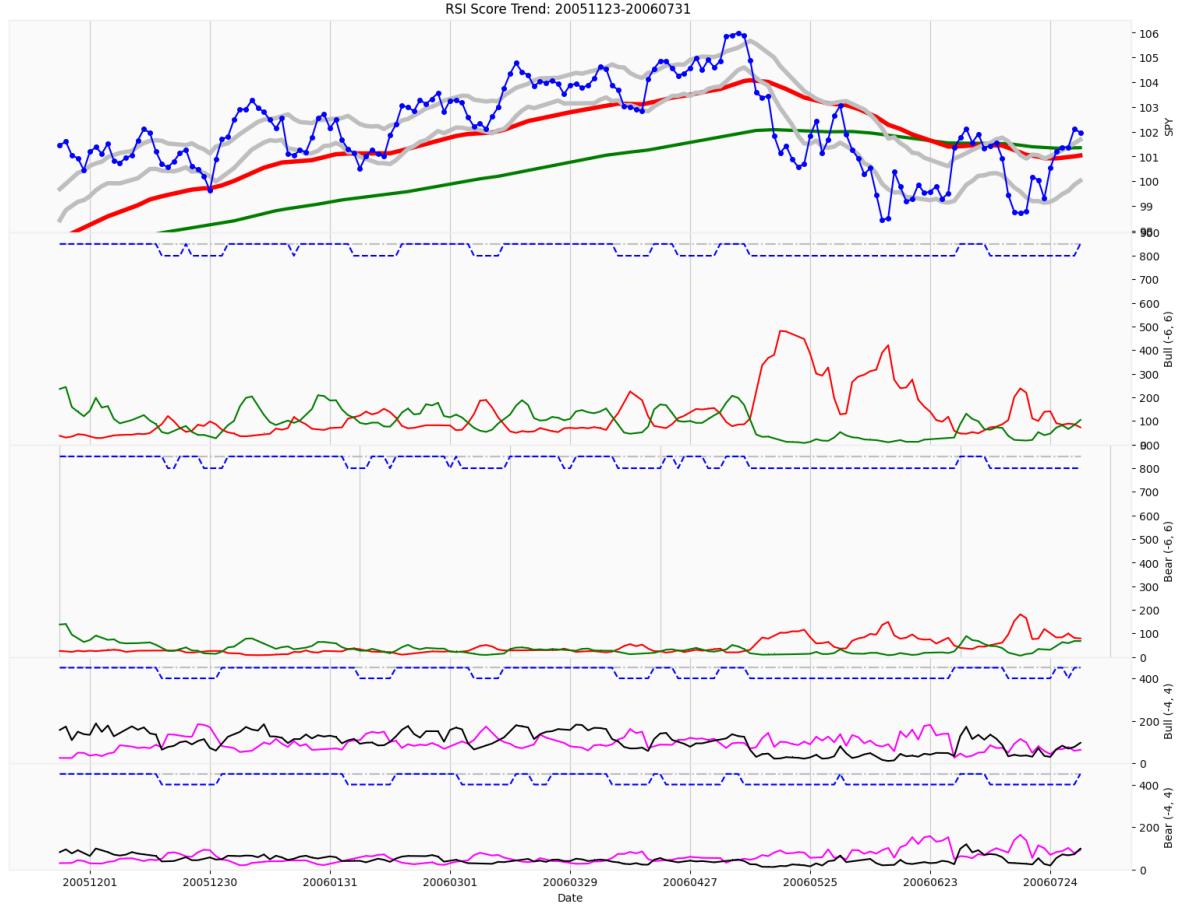
In [15]: errors = []
html = "<table>"

```
for dt in all_dates:
    ret, ret_err = trow(dt)
    if ret_err:
        errors += ret_err
    if ret:
        html += ret
html += "</table>"
display(HTML(html))

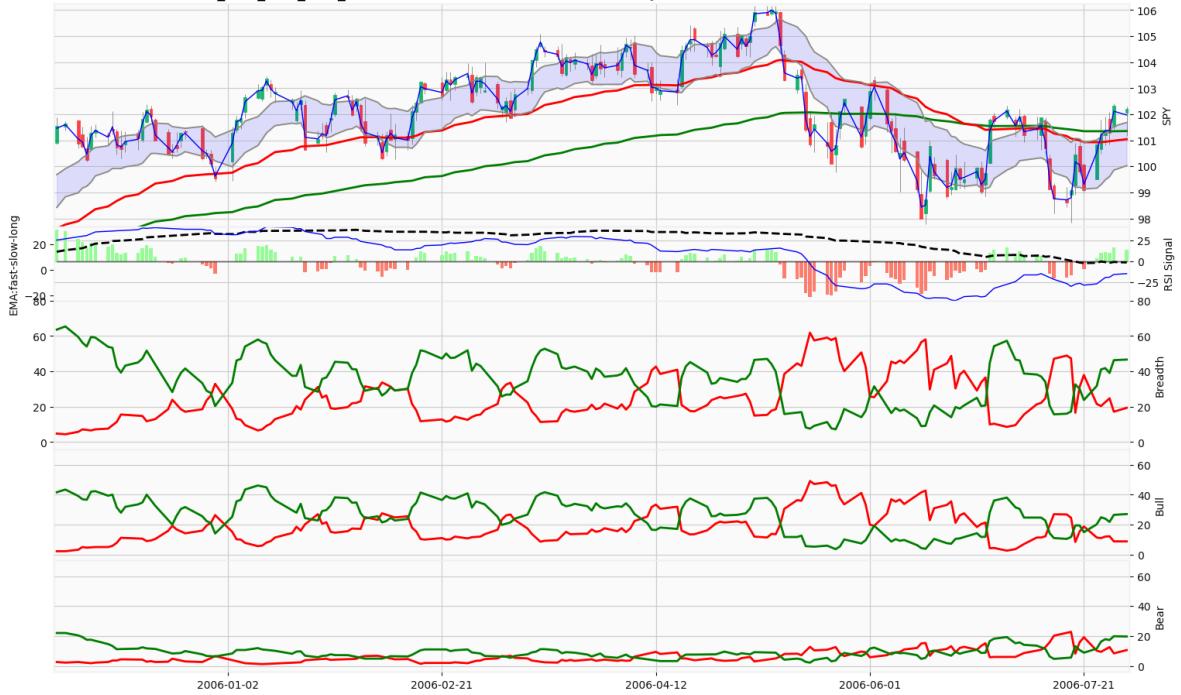
print(errors)
```

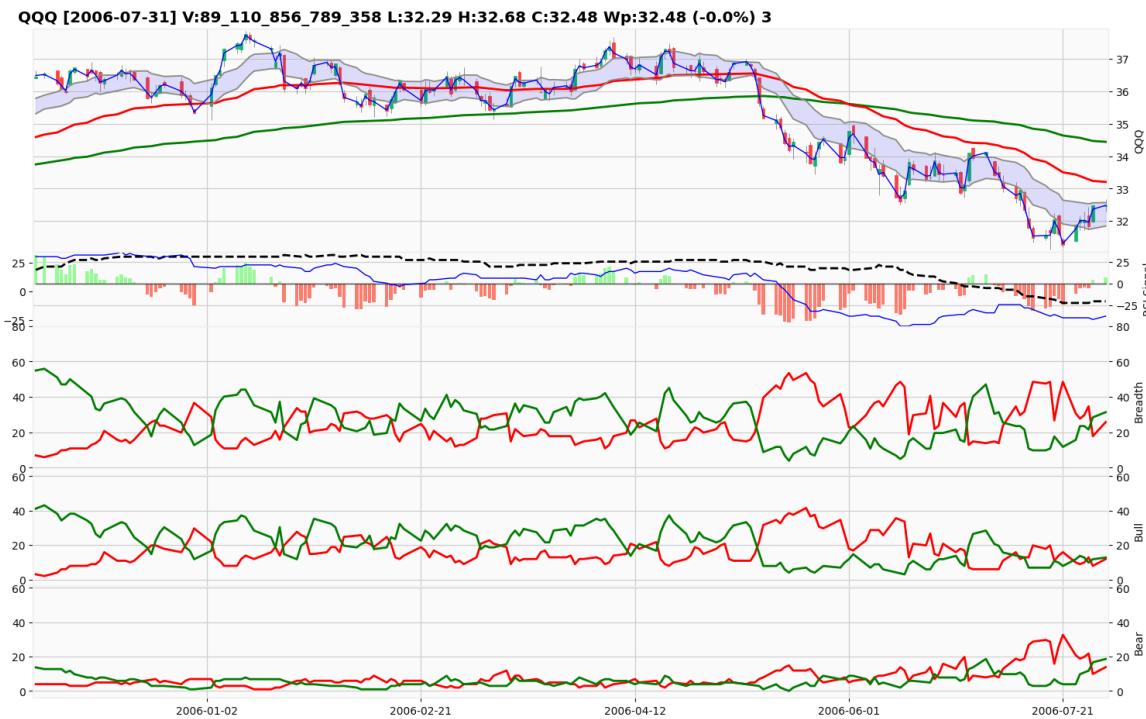


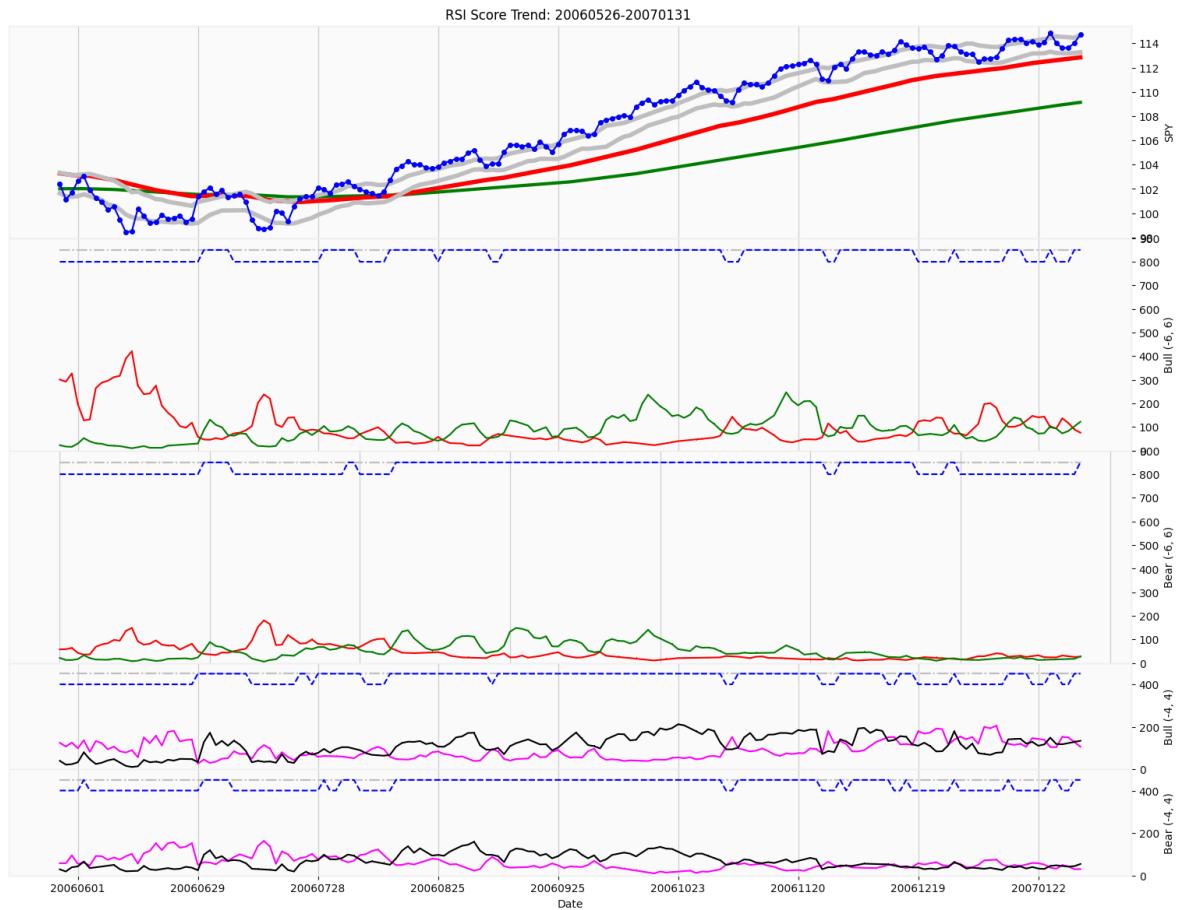




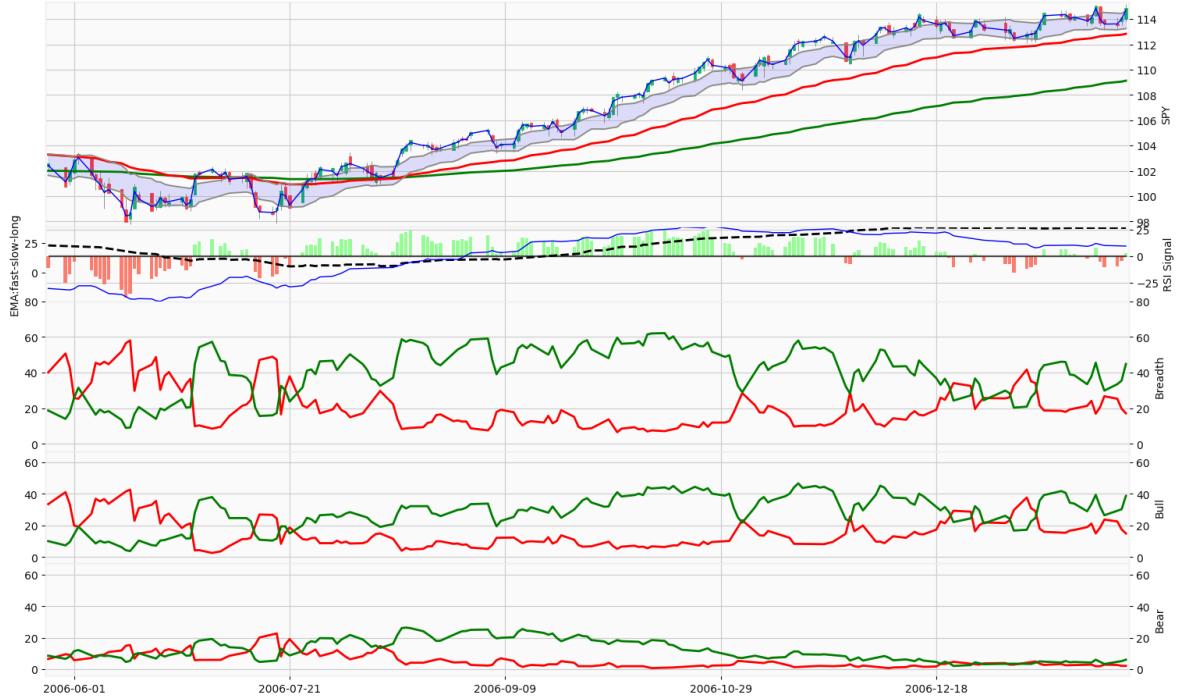
**SPY [2006-07-31] V:62\_041\_398\_196\_474 L:101.15 H:102.29 C:102.22 Wp:101.97 (-0.1%) 4**

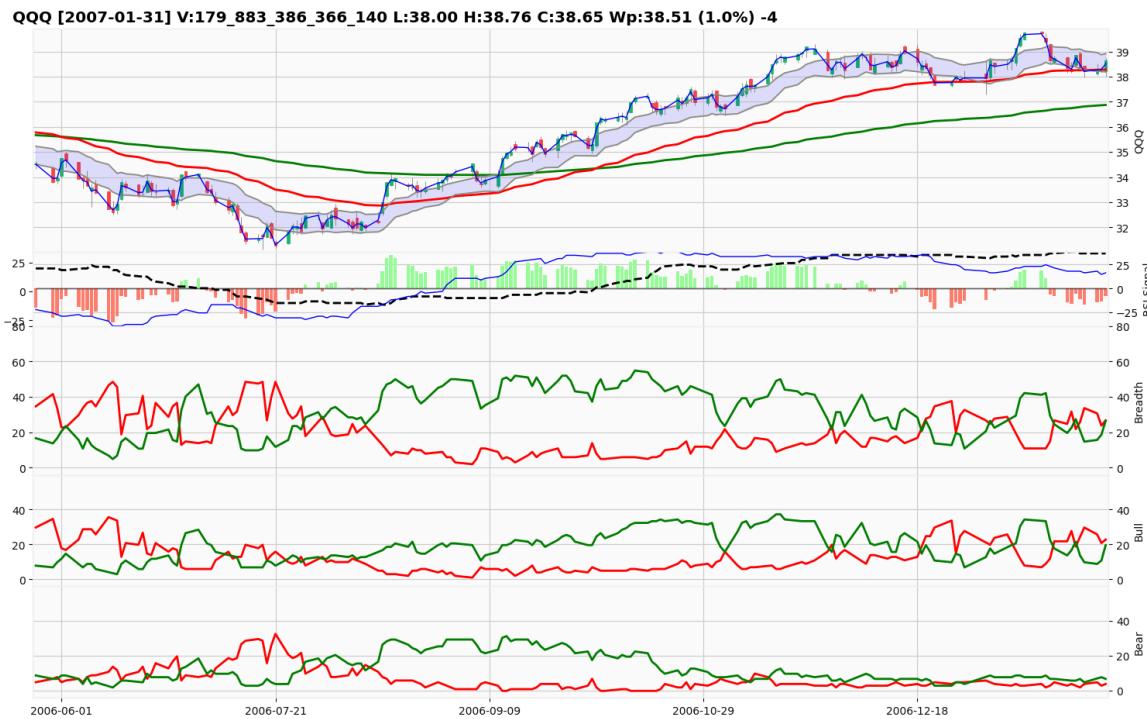


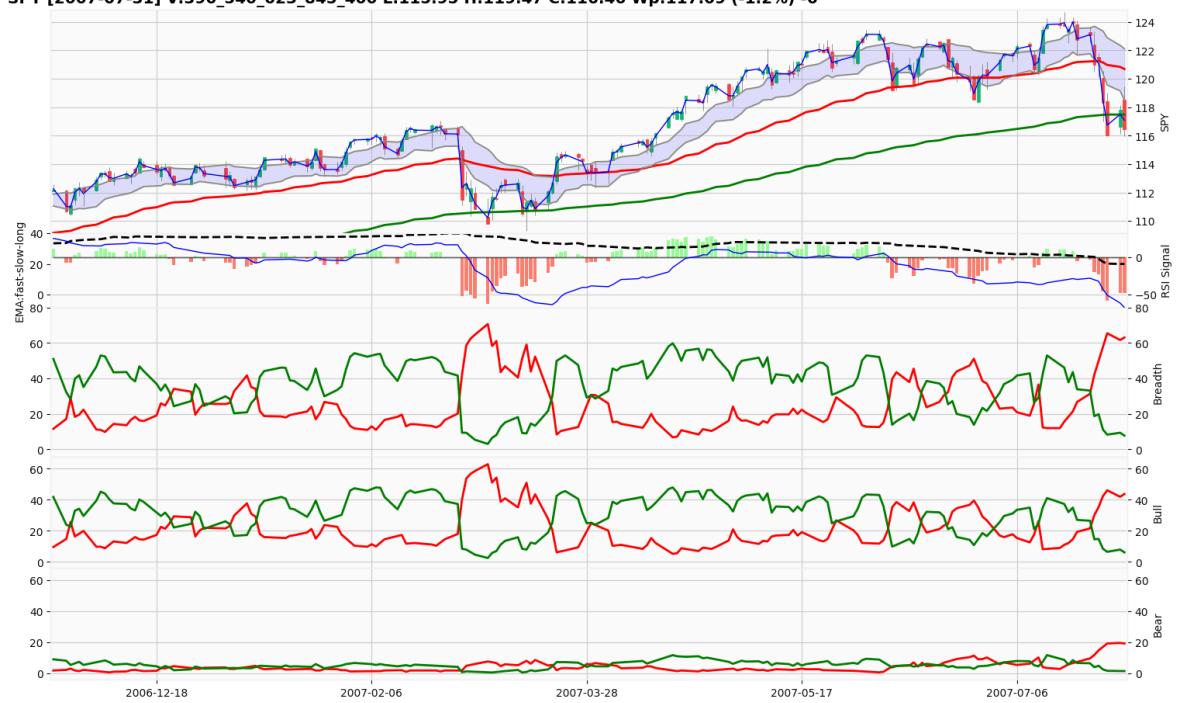
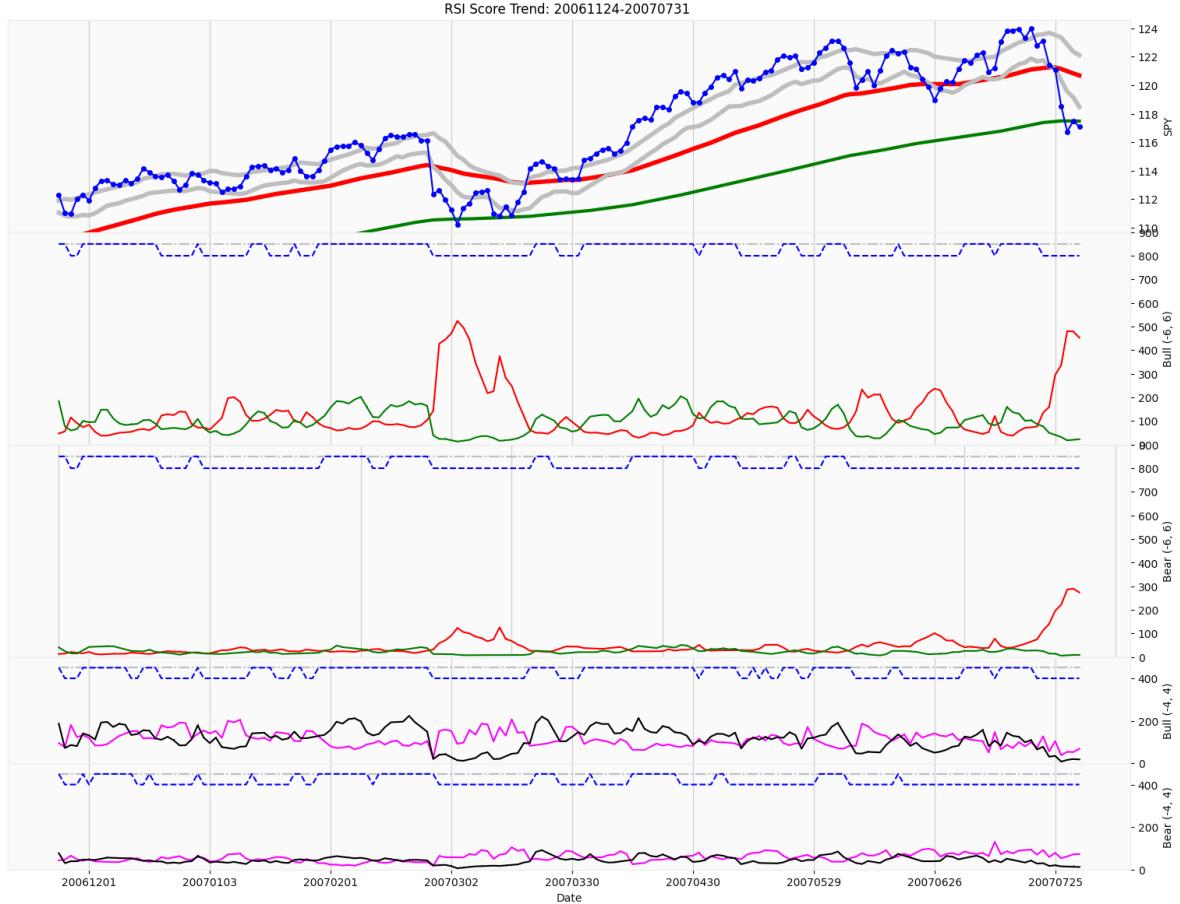


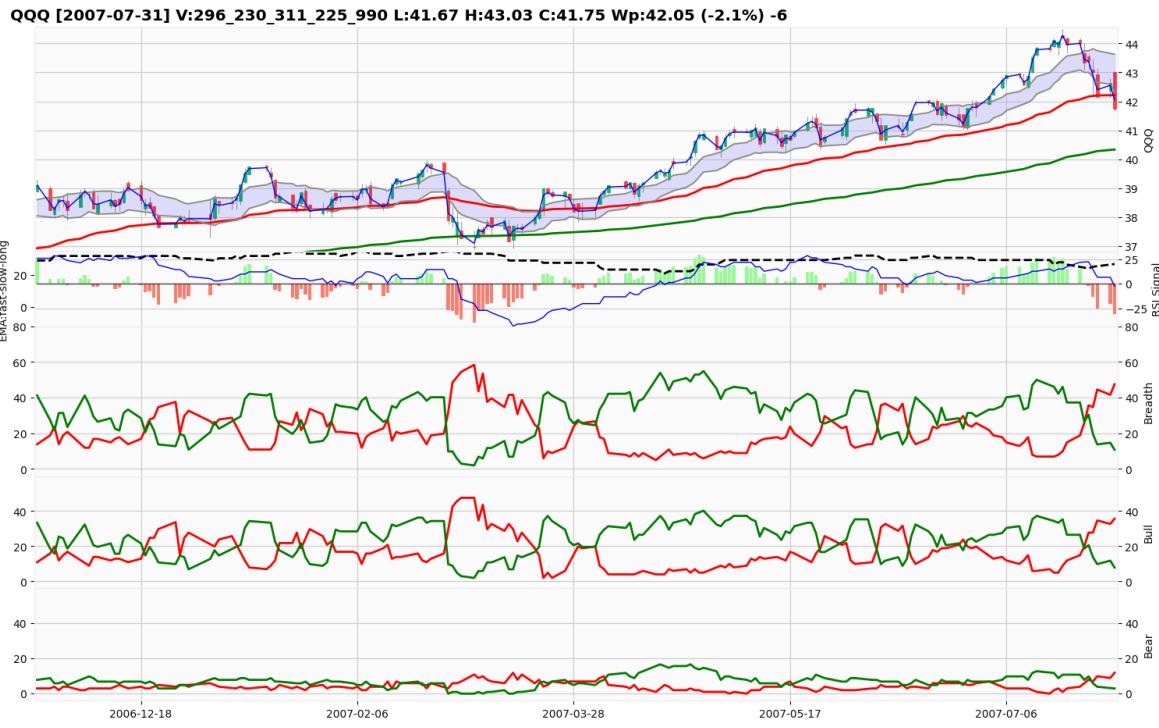


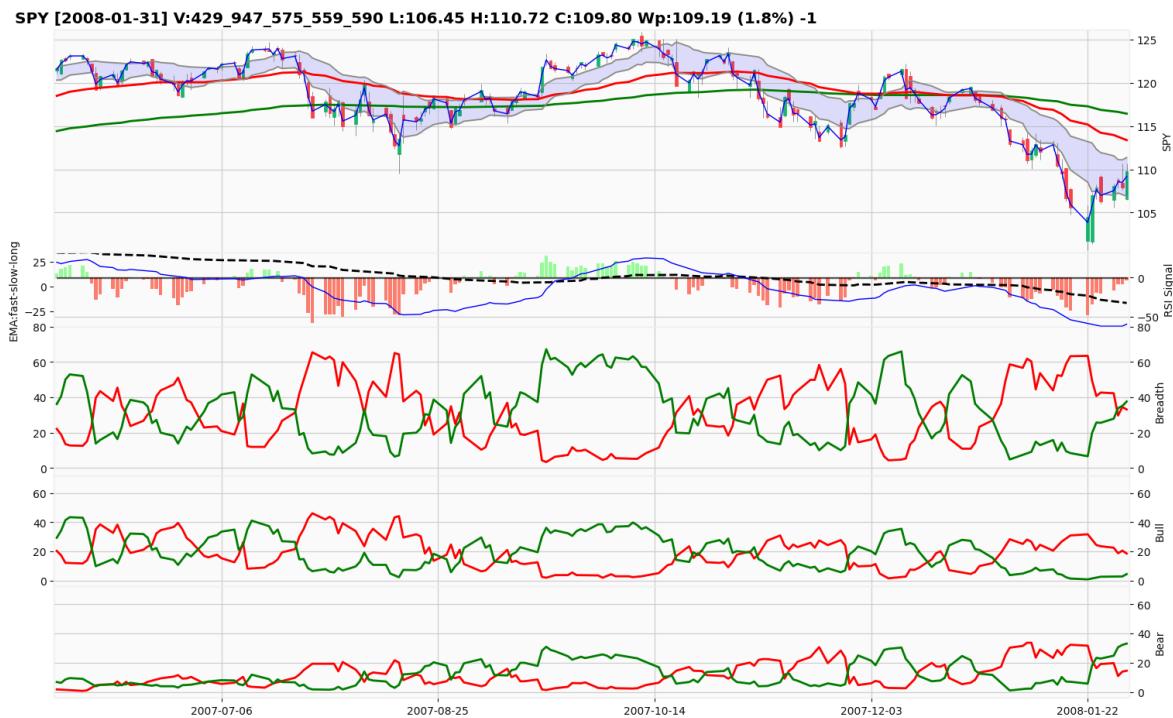
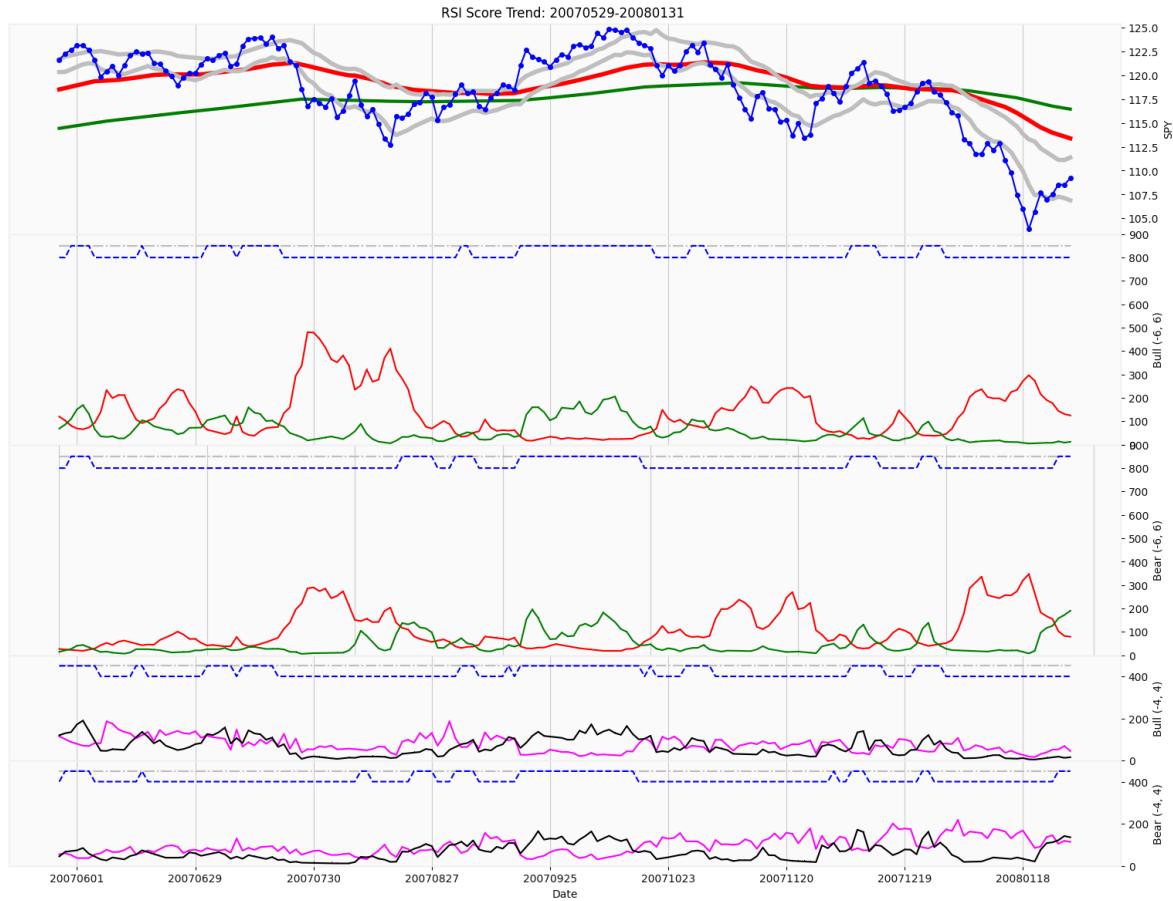
**SPY [2007-01-31] V:114\_973\_957\_681\_700 L:113.85 H:115.19 C:114.90 Wp:114.71 (0.7%) 0**

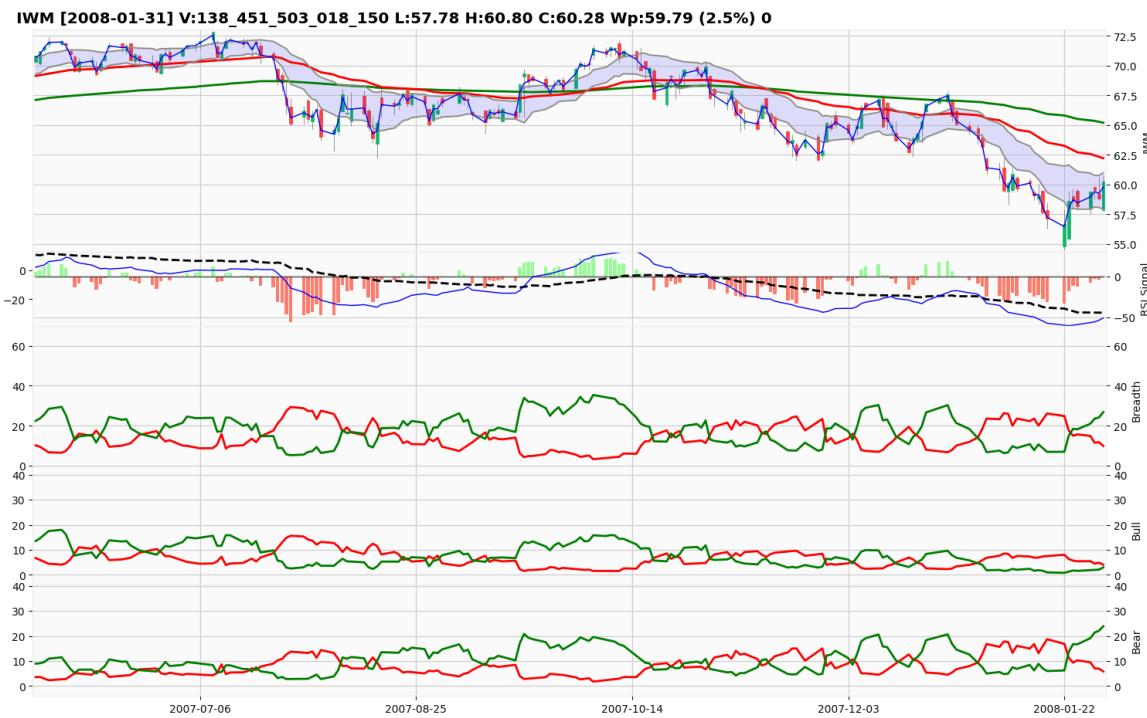
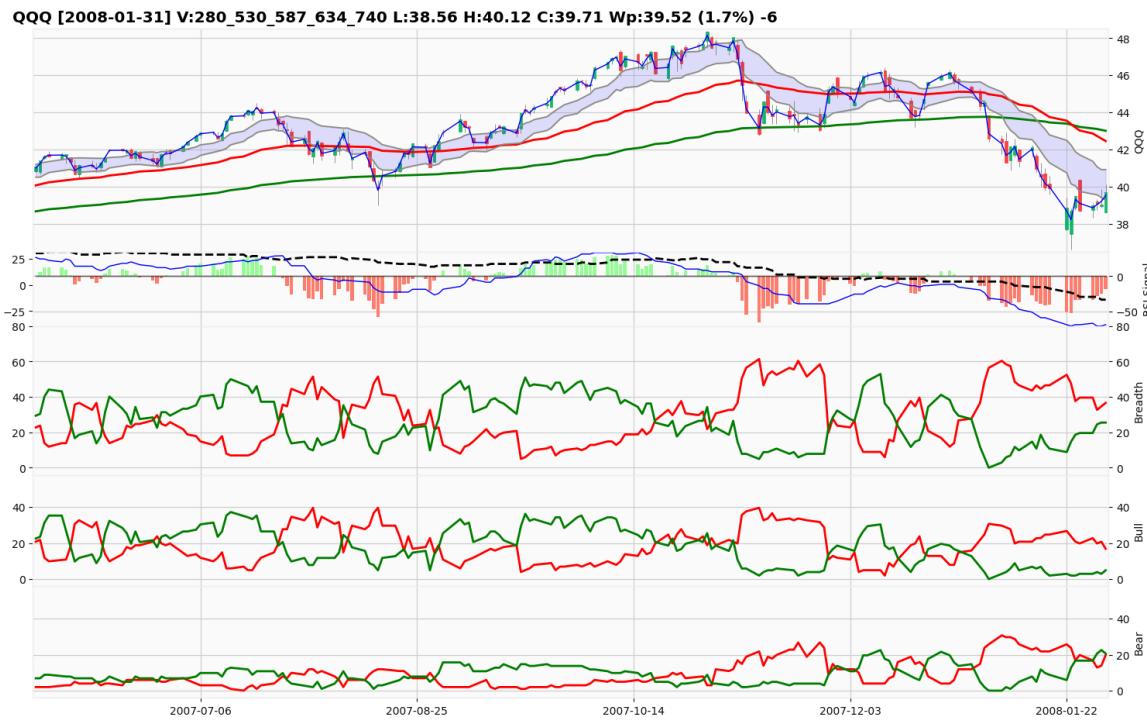


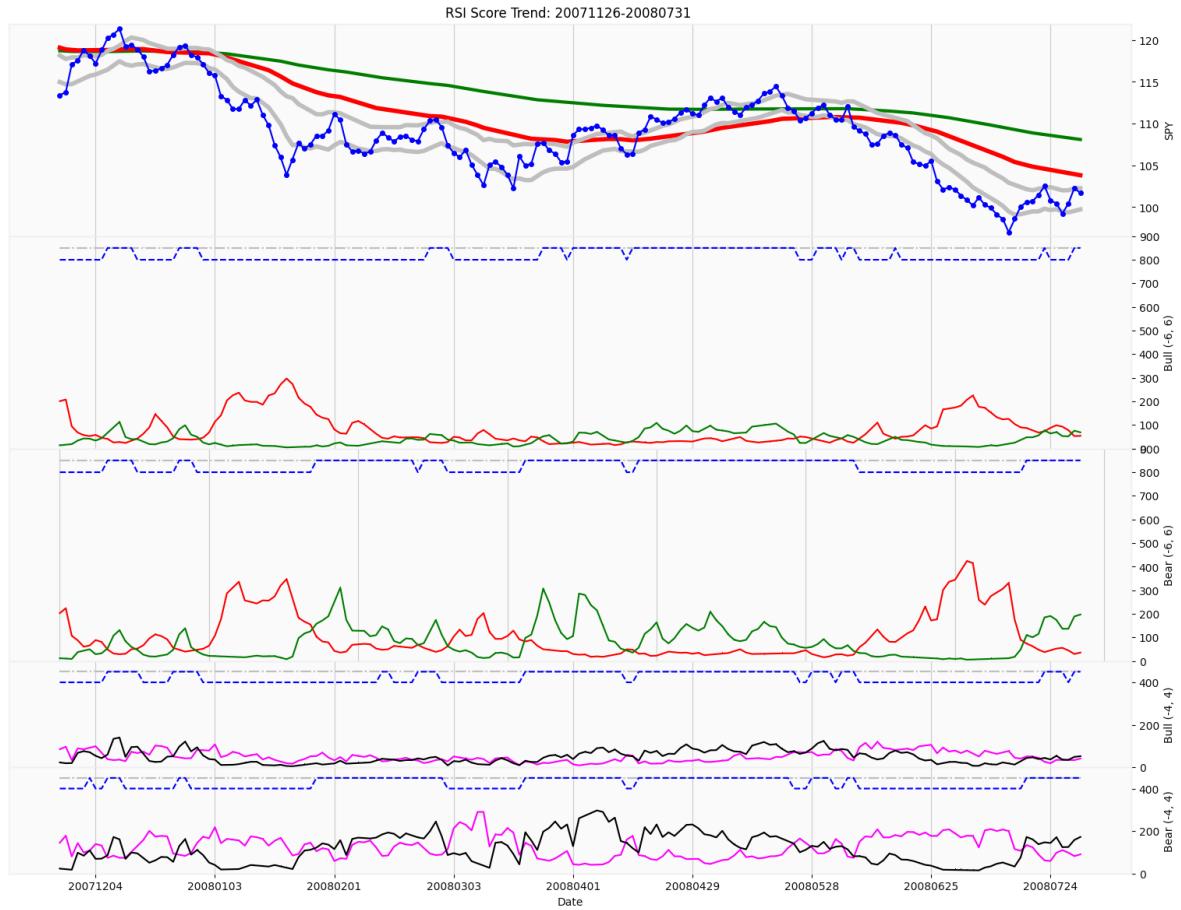








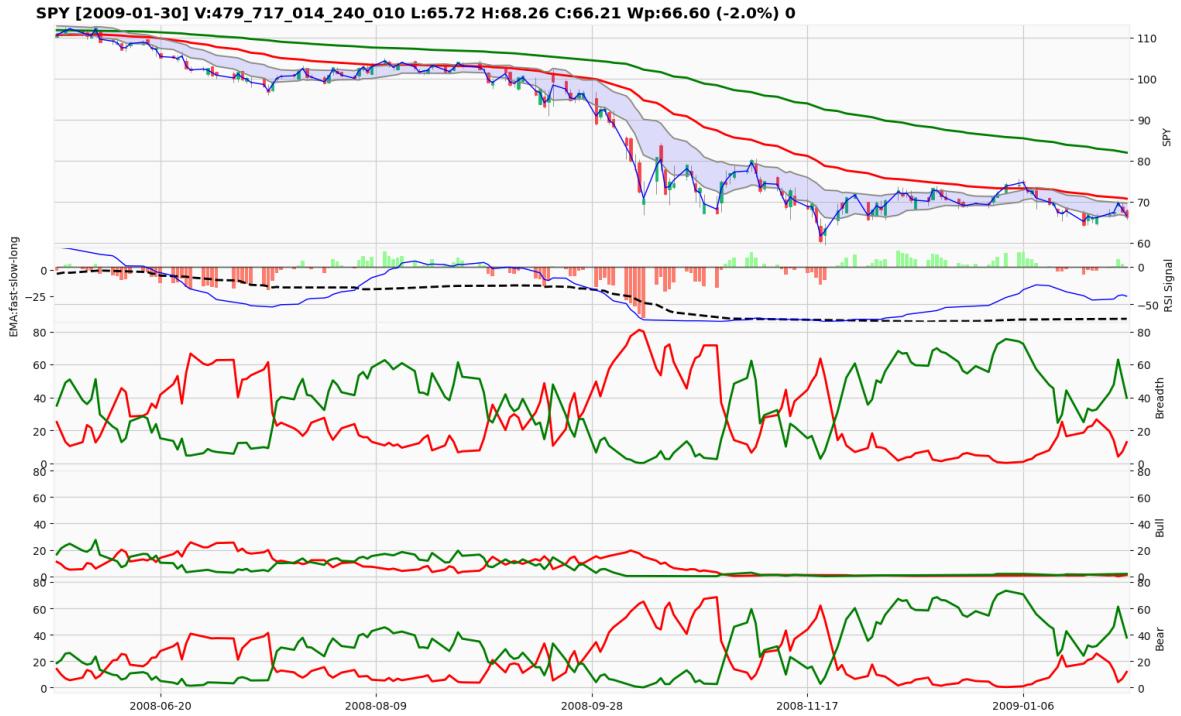
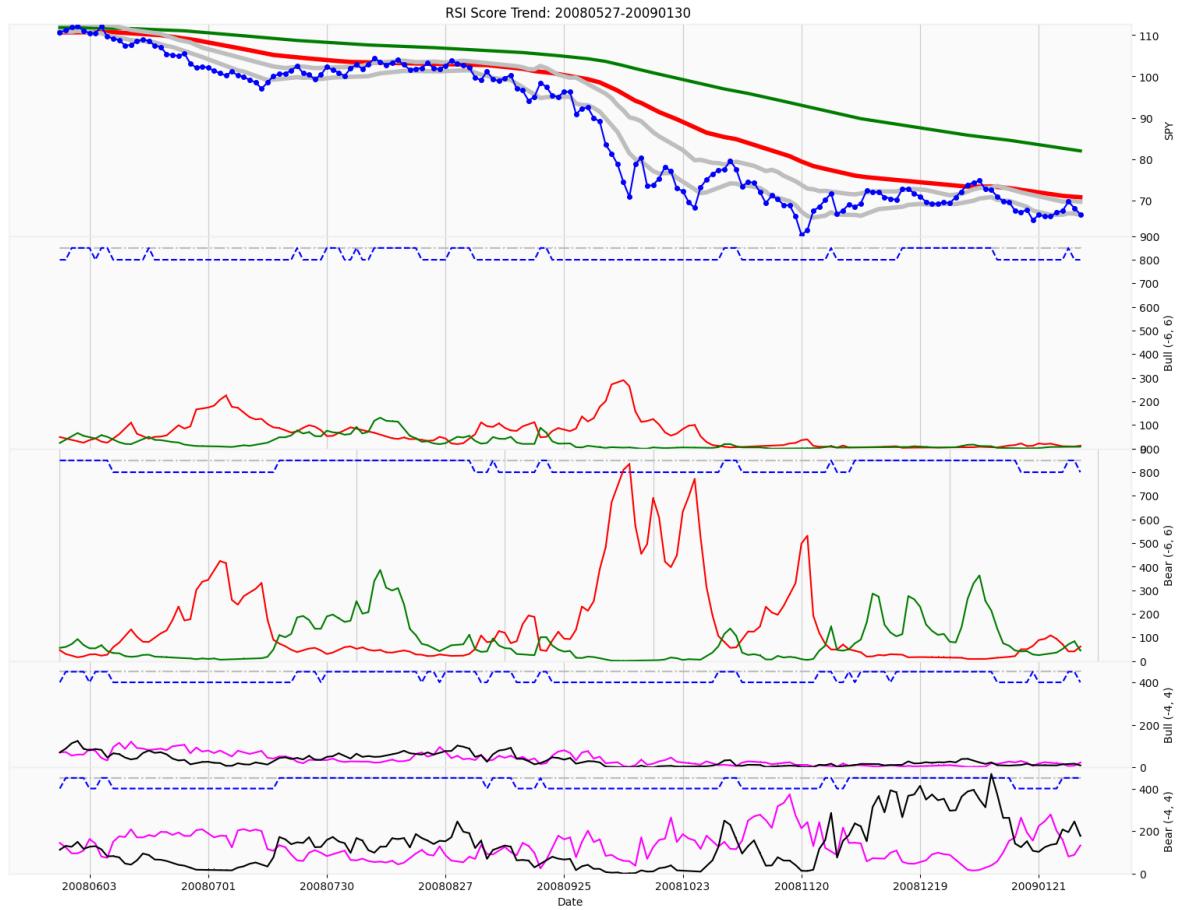




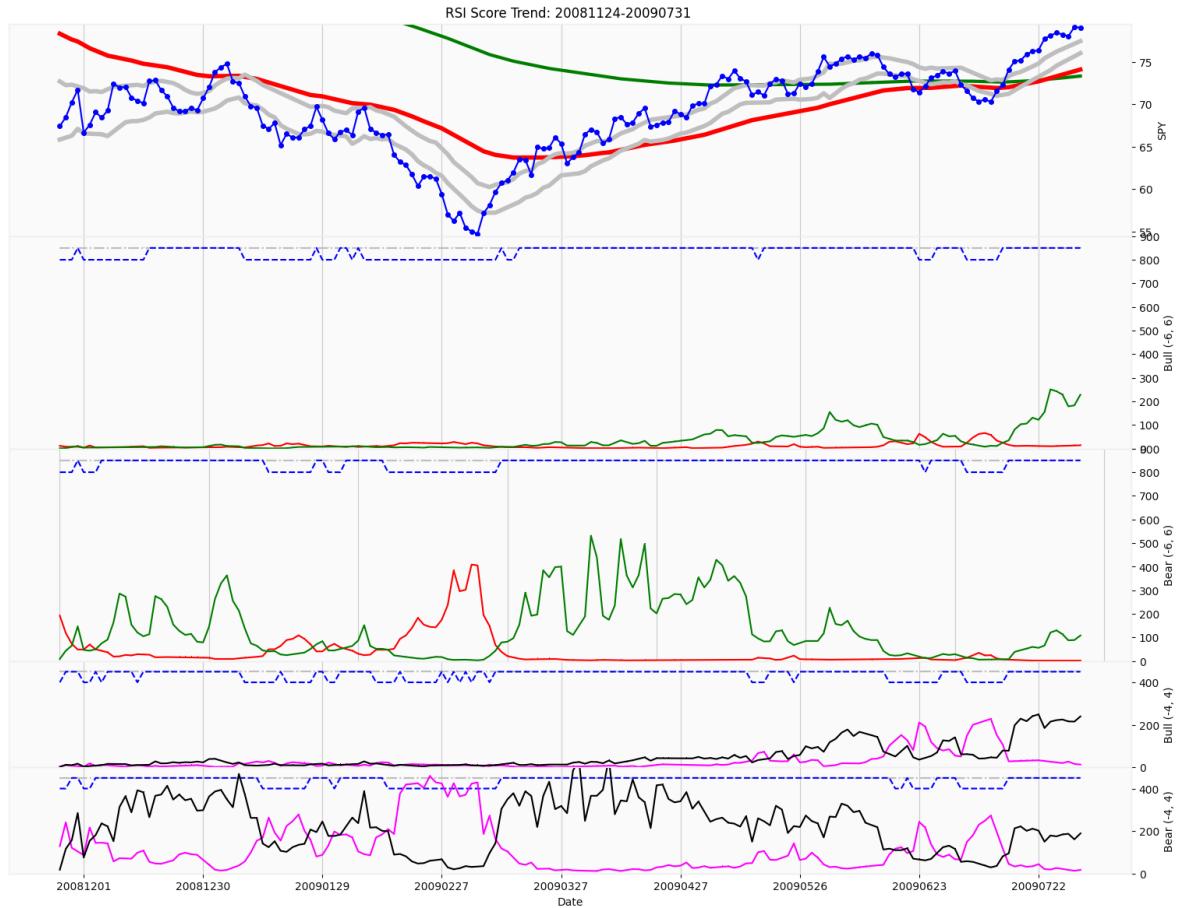
**SPY [2008-07-31] V:347\_032\_556\_835\_260 L:101.21 H:102.76 C:101.38 Wp:101.68 (-1.3%) 4**



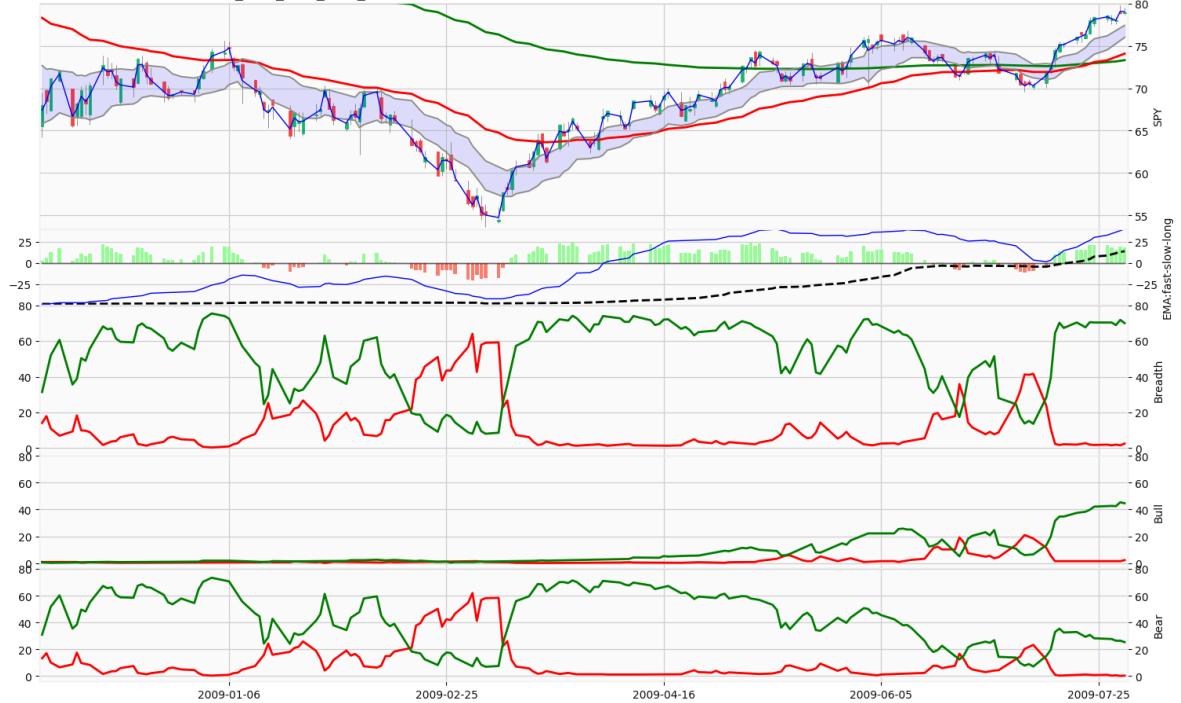


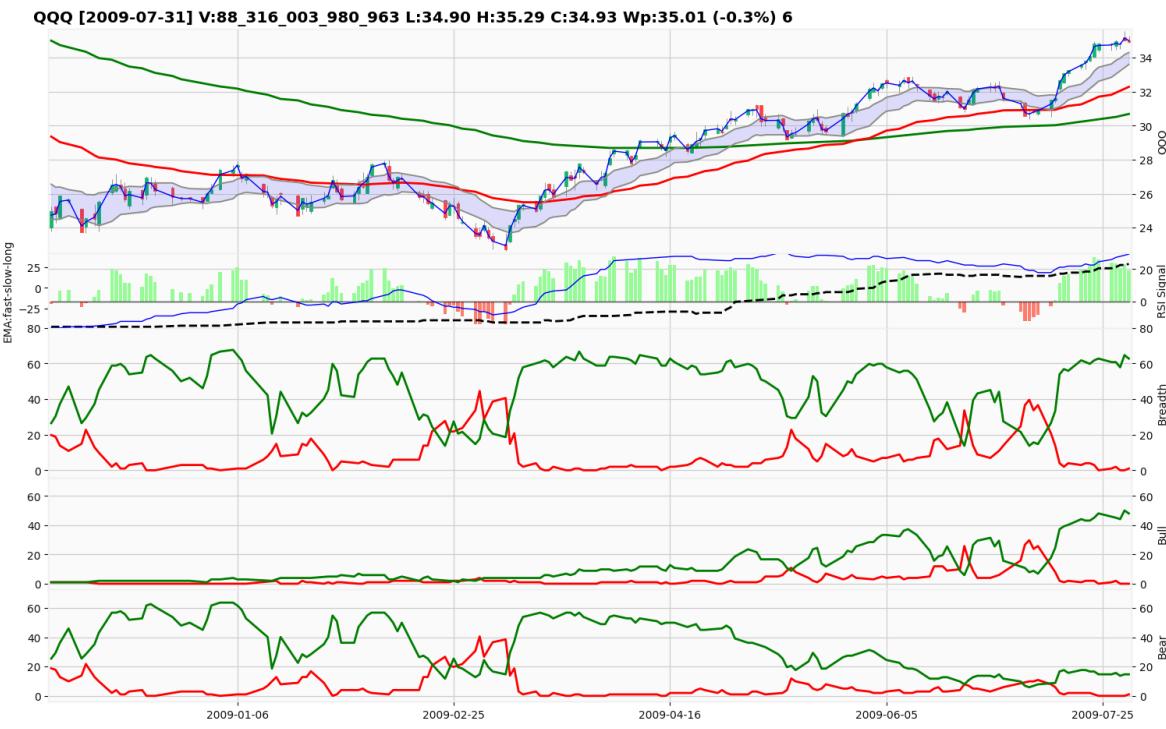


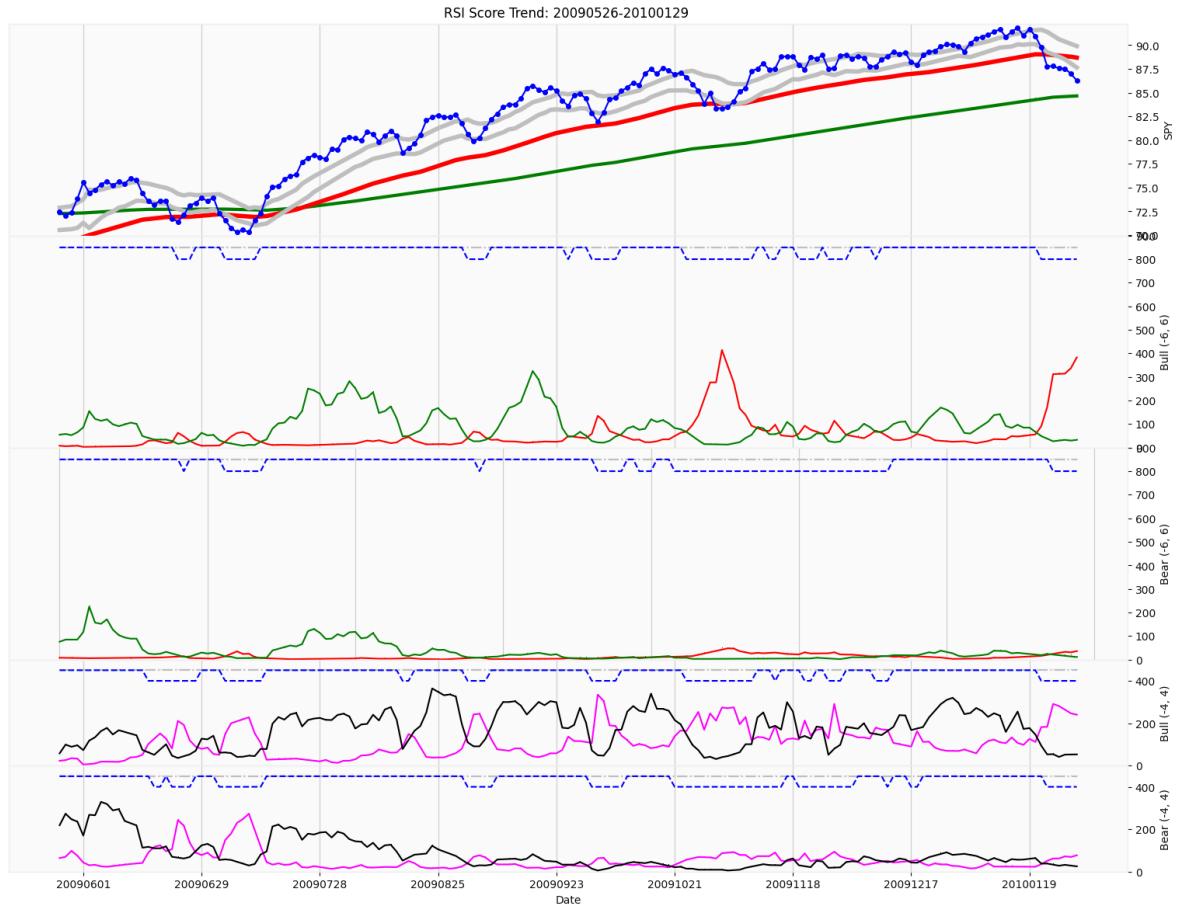




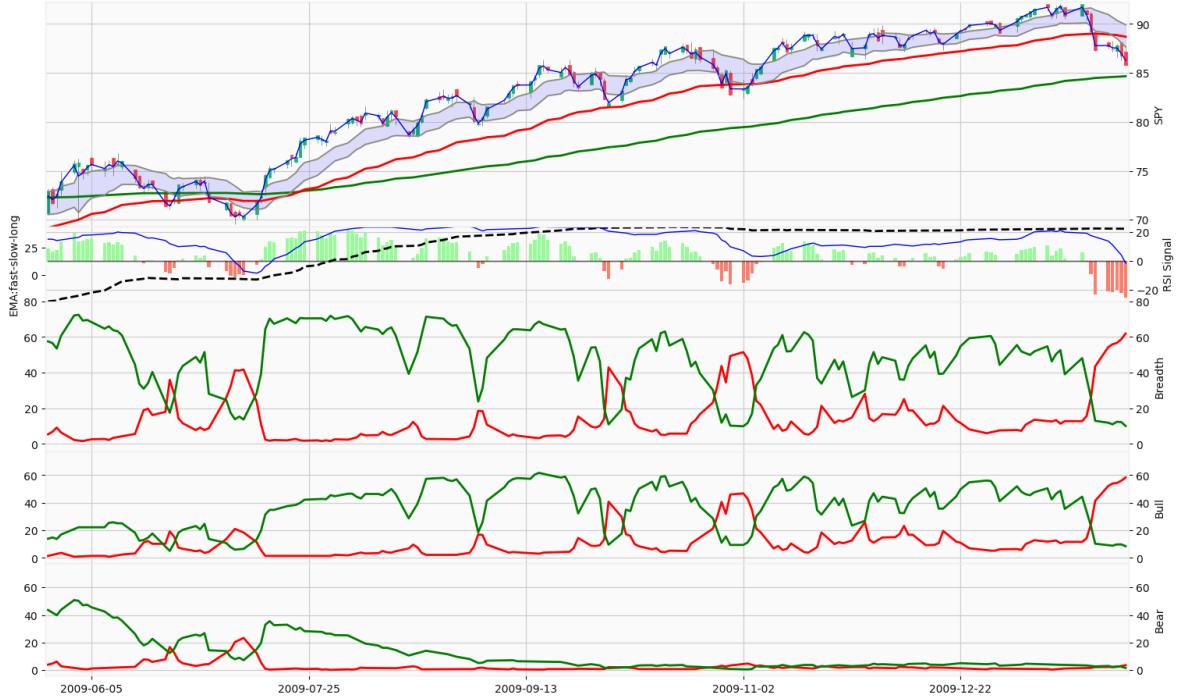
**SPY [2009-07-31] V:259\_406\_611\_704\_900 L:78.64 H:79.51 C:78.98 Wp:79.03 (0.1%) 6**

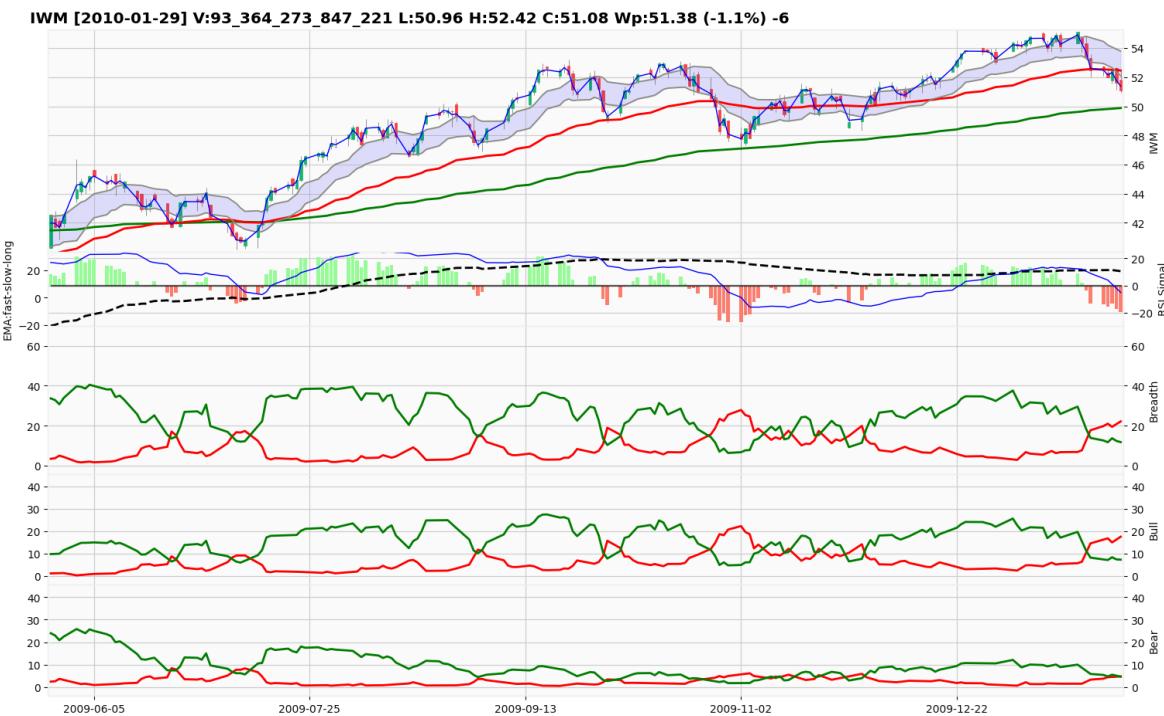
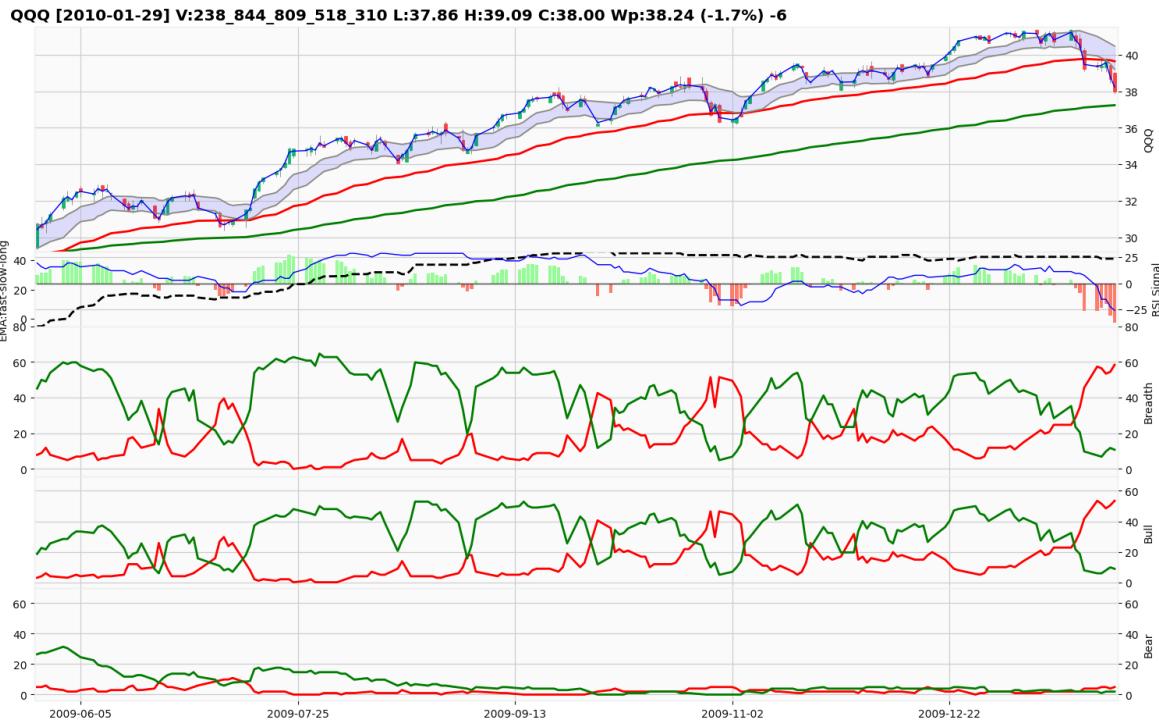


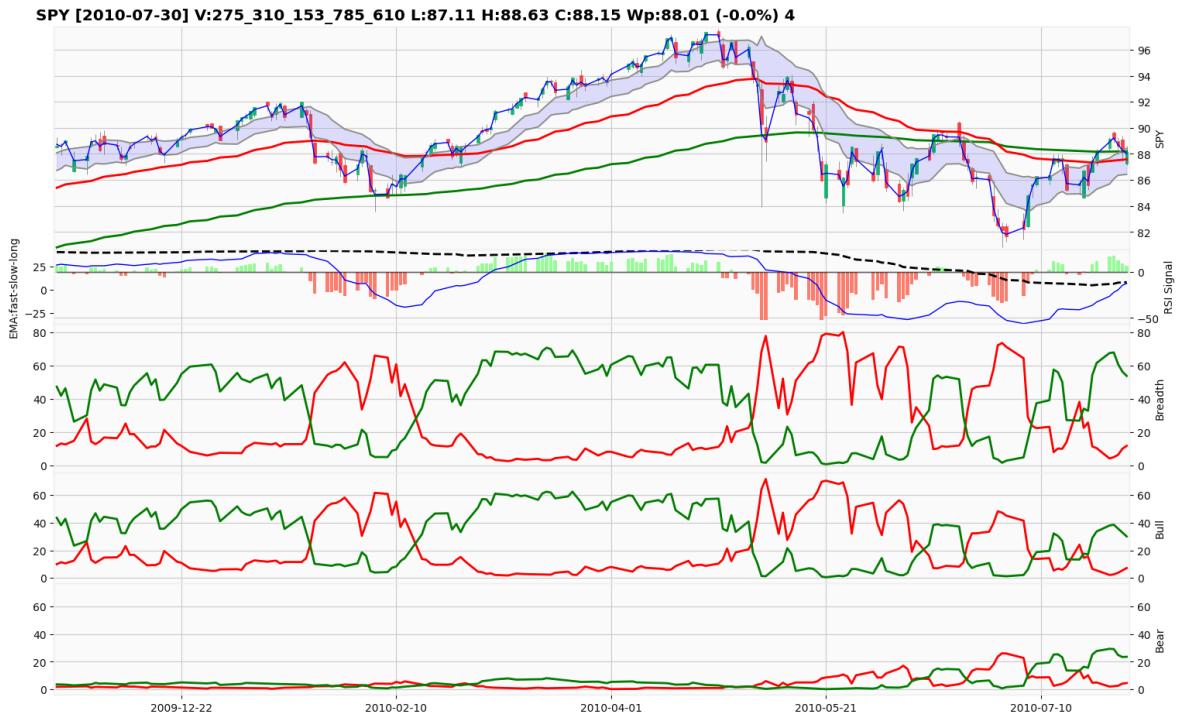
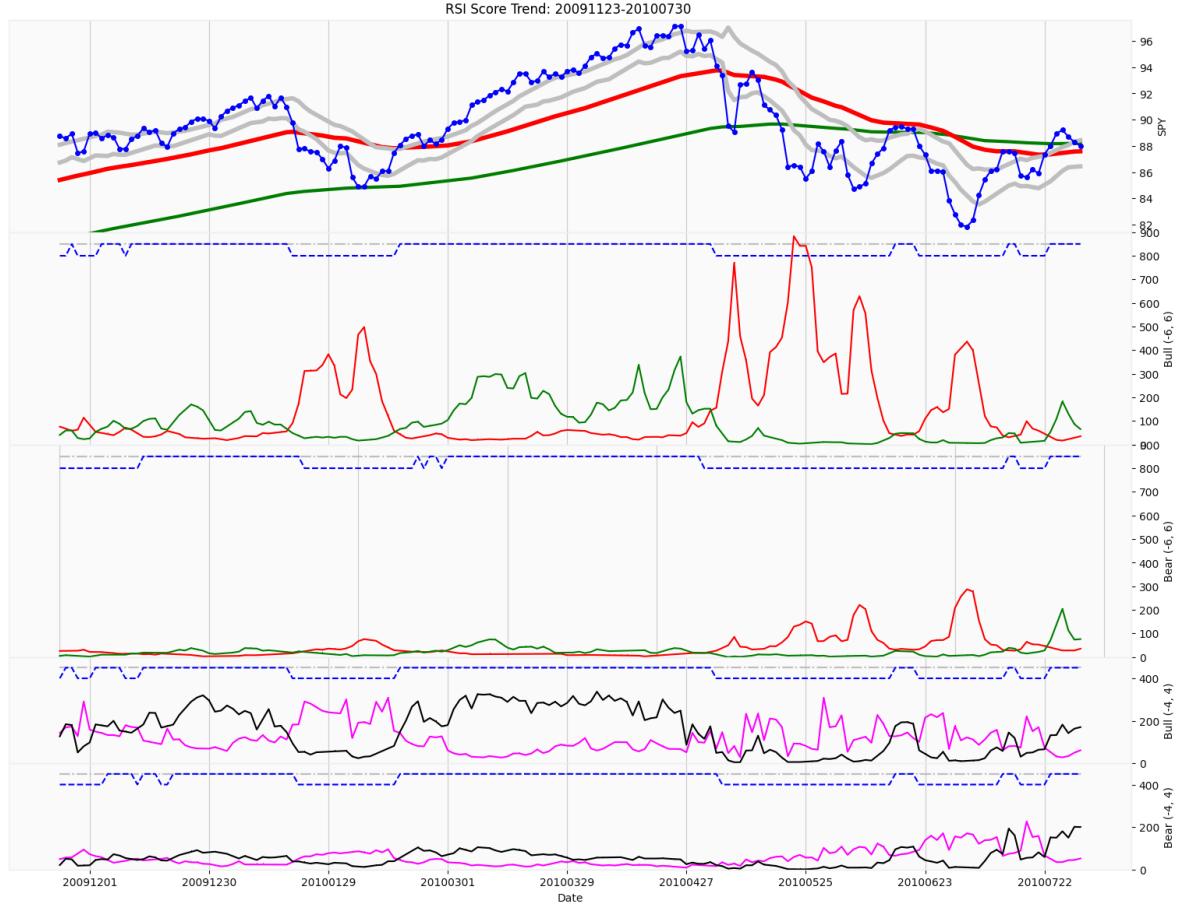


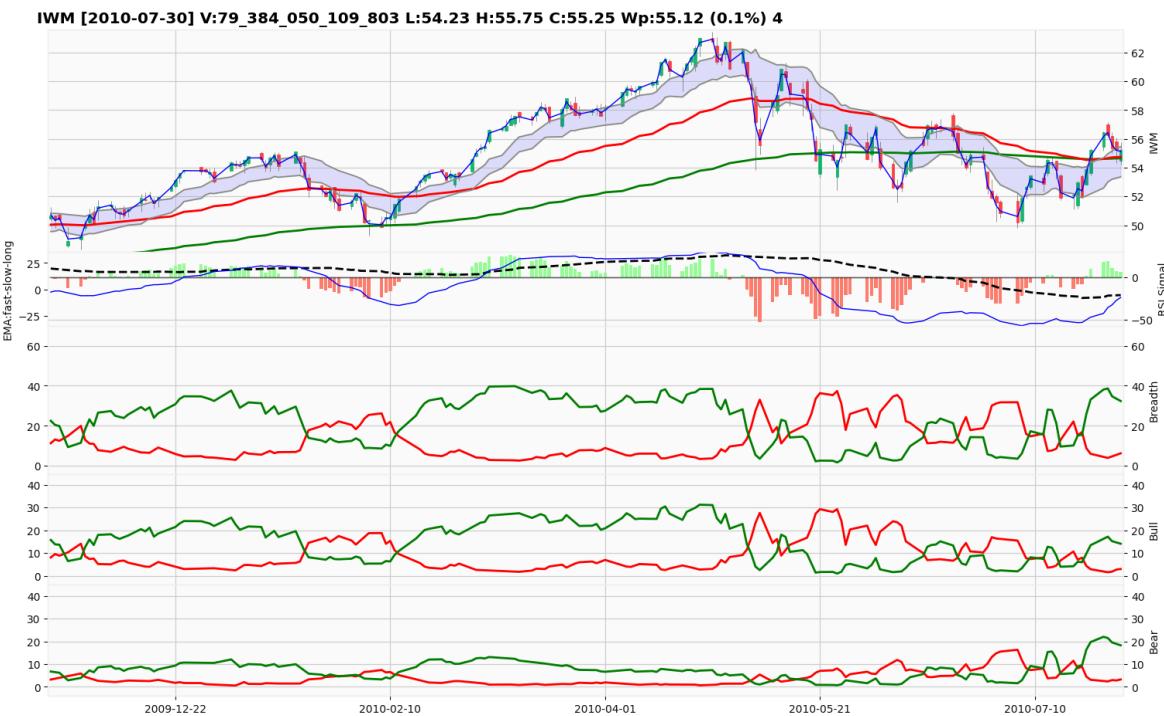
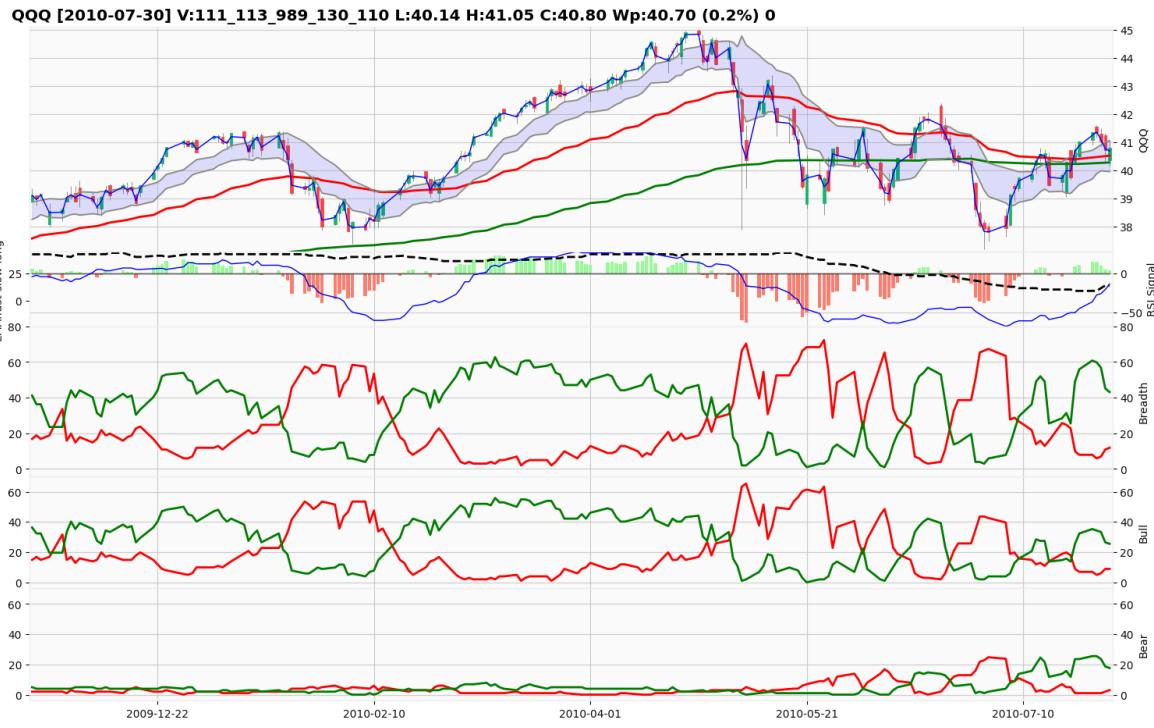


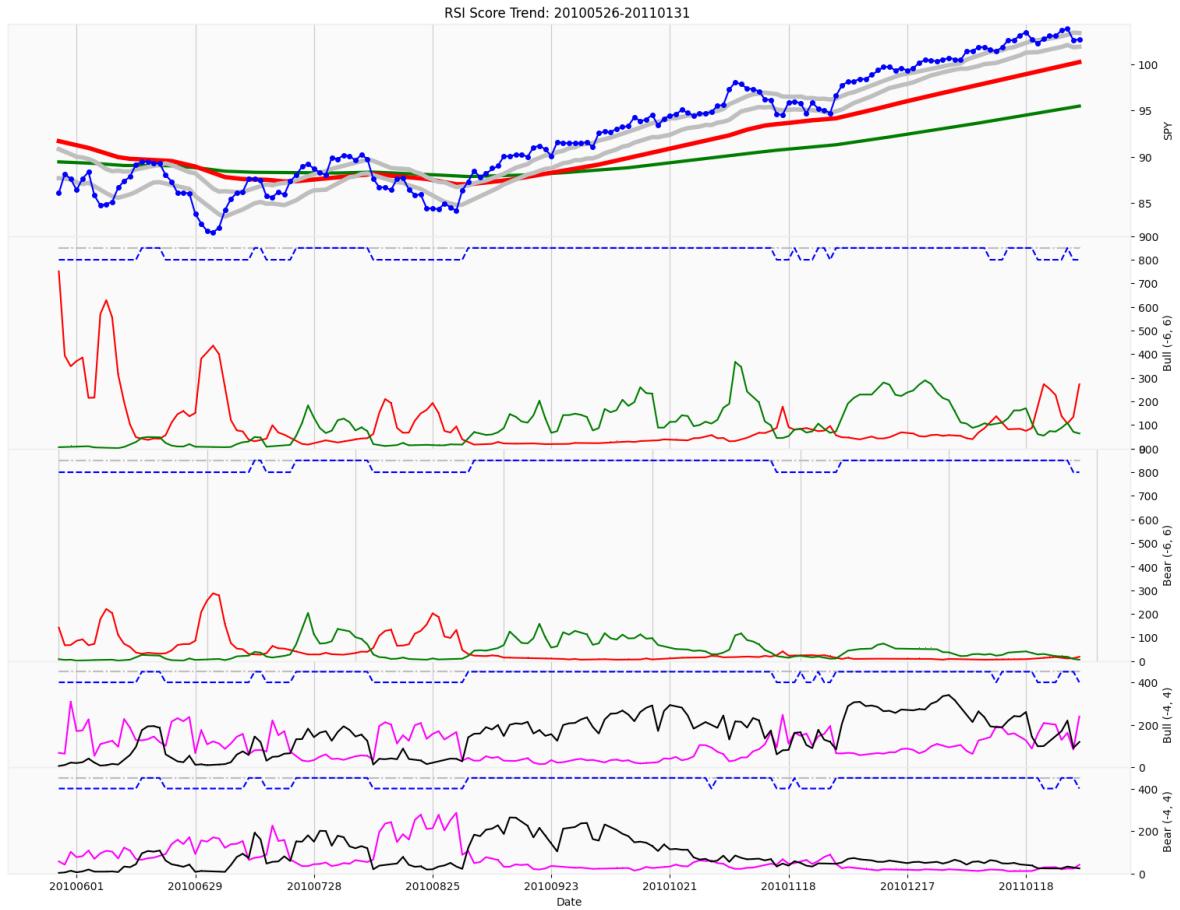
SPY [2010-01-29] V:388\_660\_378\_615\_110 L:85.70 H:87.76 C:85.84 Wp:86.28 (-1.1%) -6

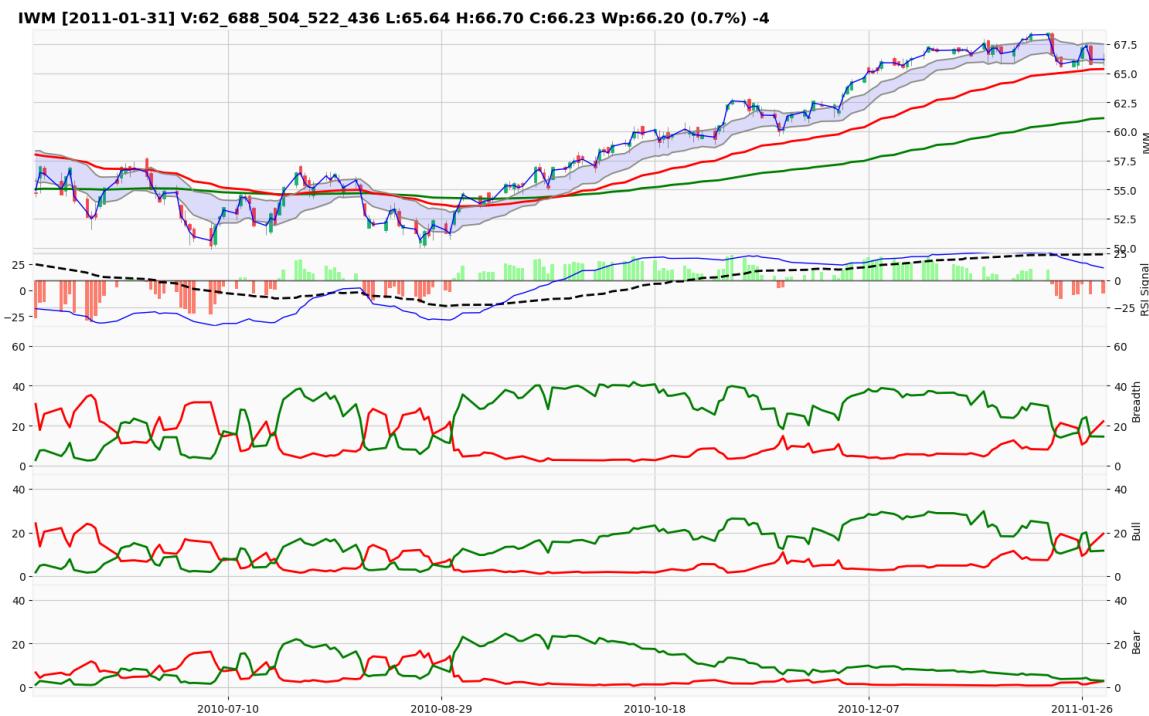
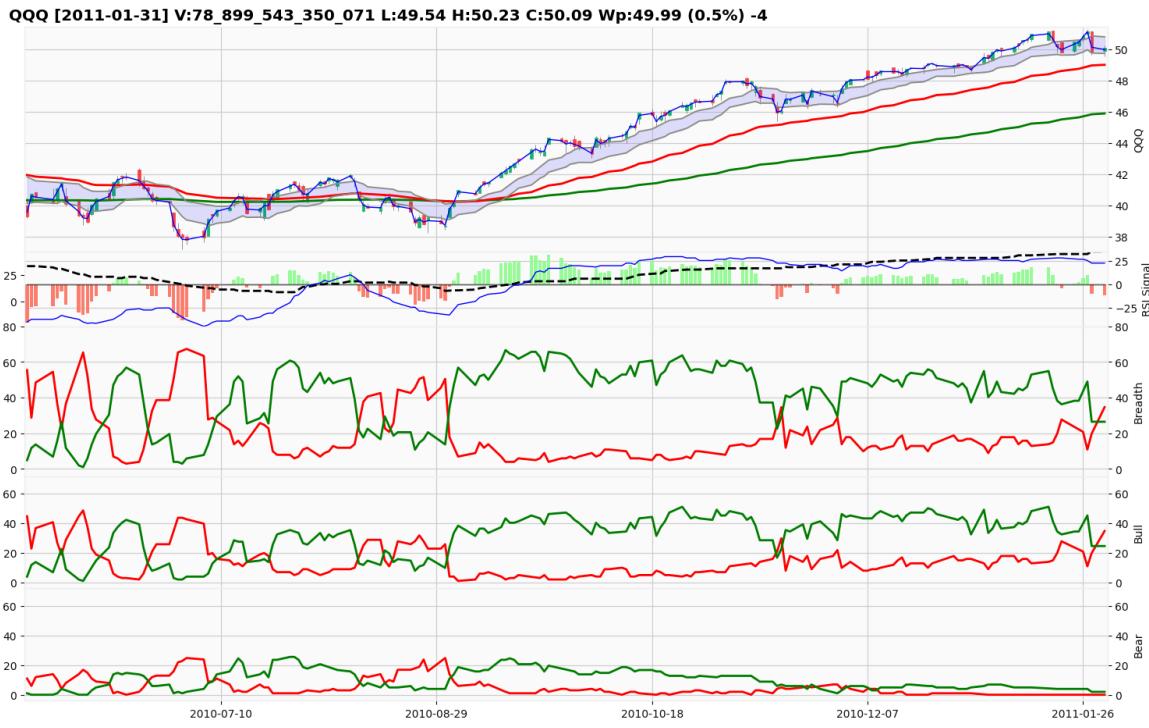


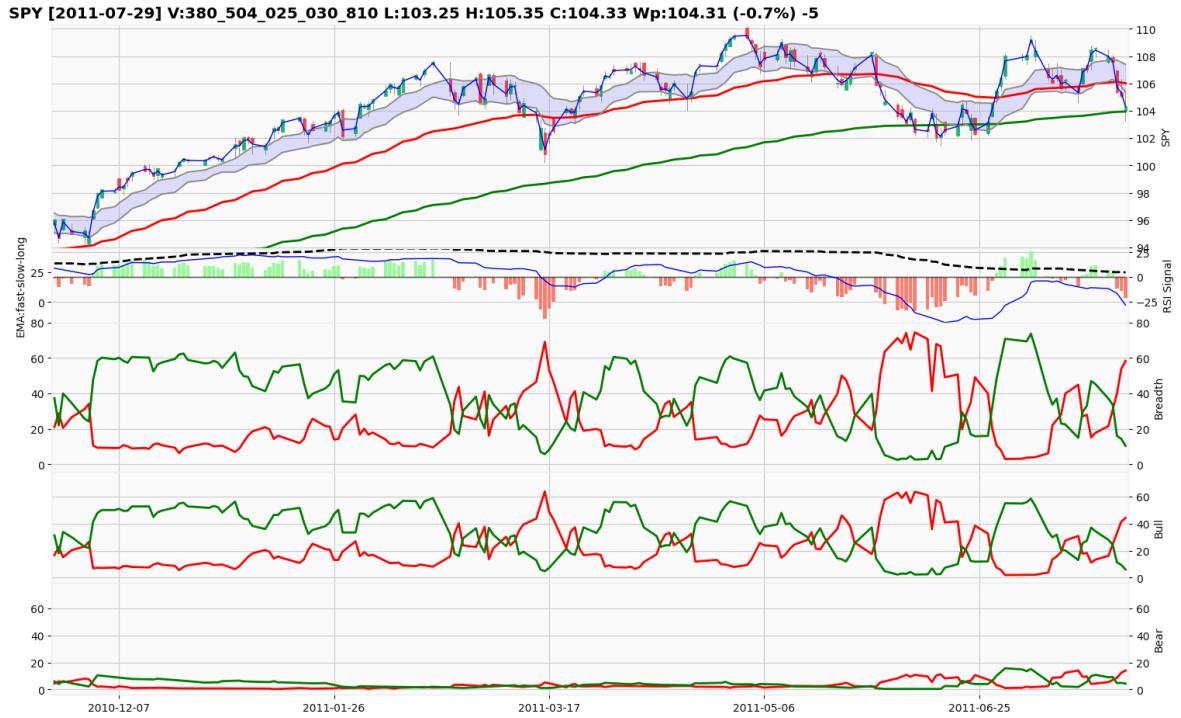
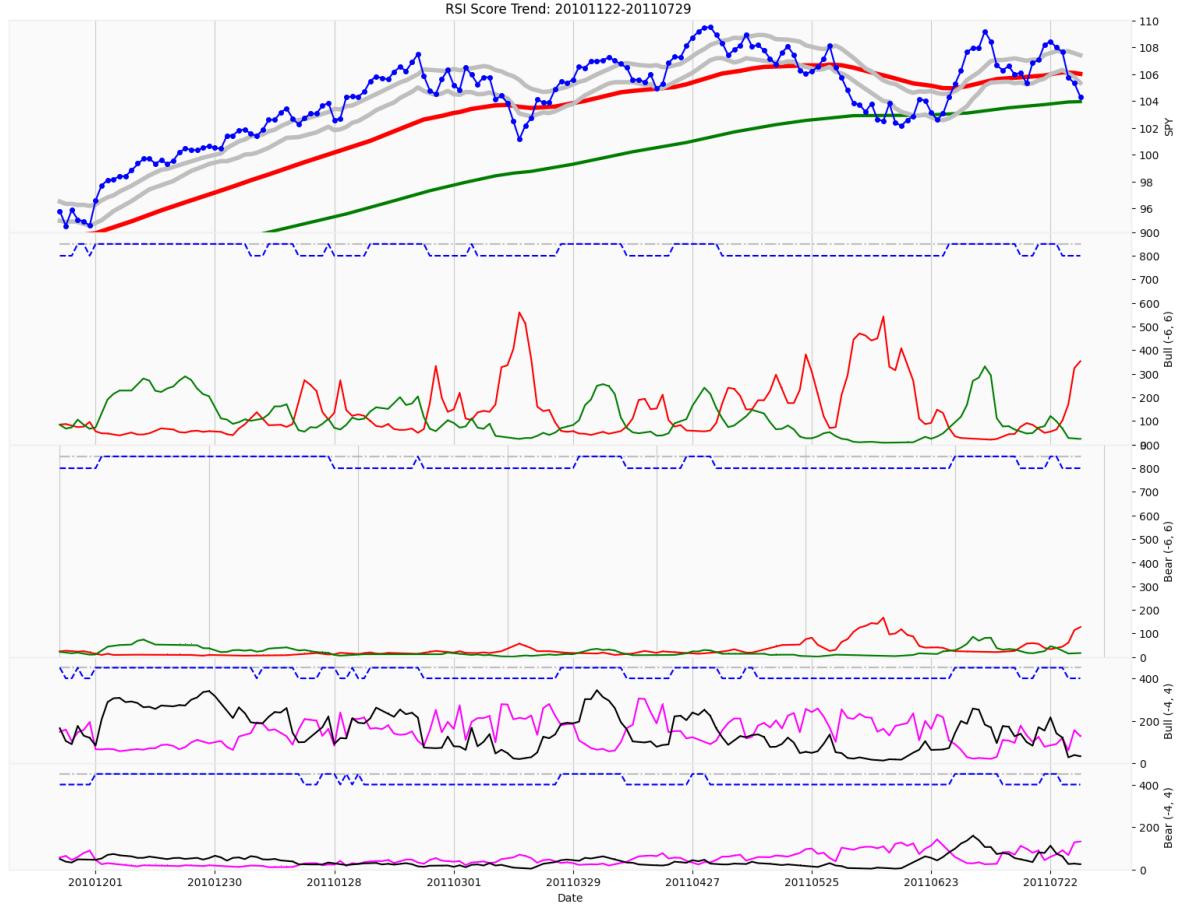










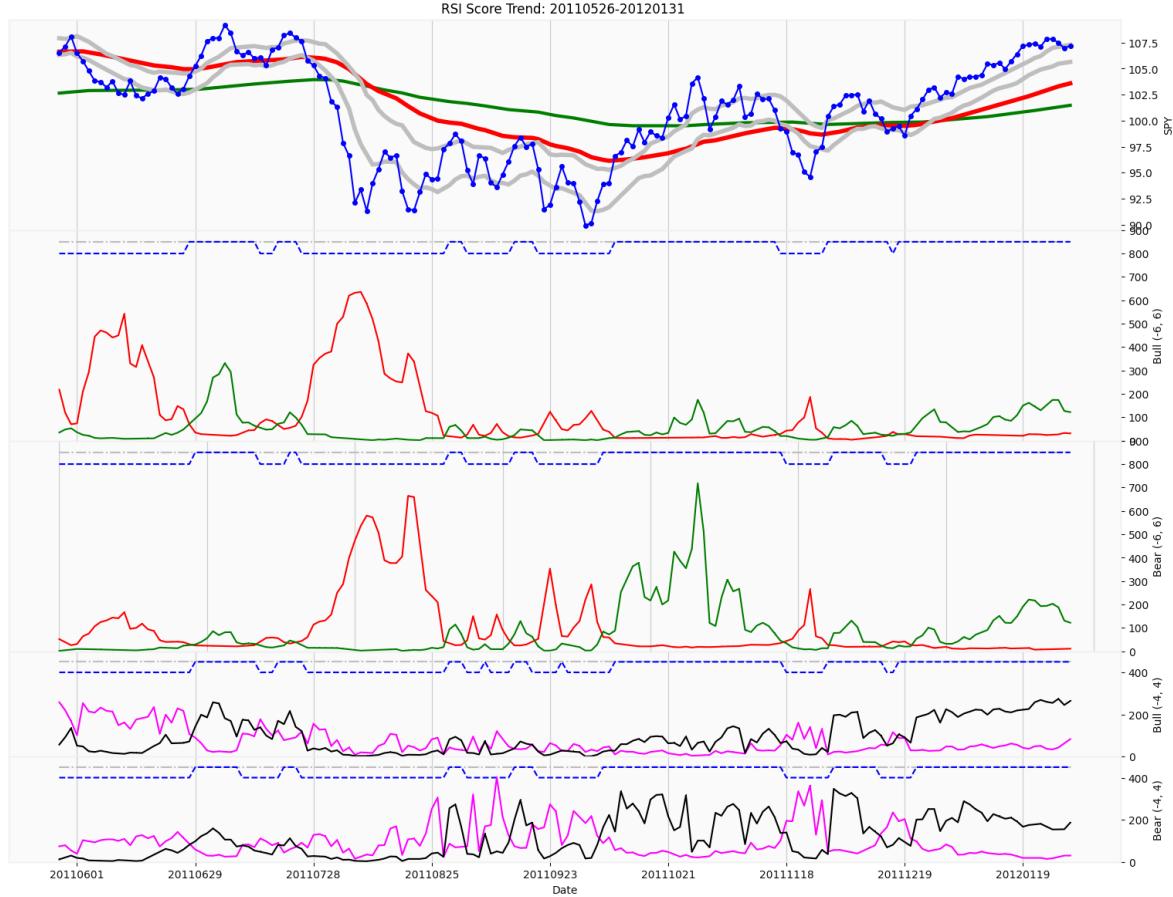


**QQQ [2011-07-29] V:92\_296\_540\_308\_148 L:51.56 H:52.61 C:52.07 Wp:52.08 (-0.3%) 0**



**IWM [2011-07-29] V:109\_493\_913\_814\_150 L:66.73 H:68.48 C:68.10 Wp:67.85 (-0.1%) -6**



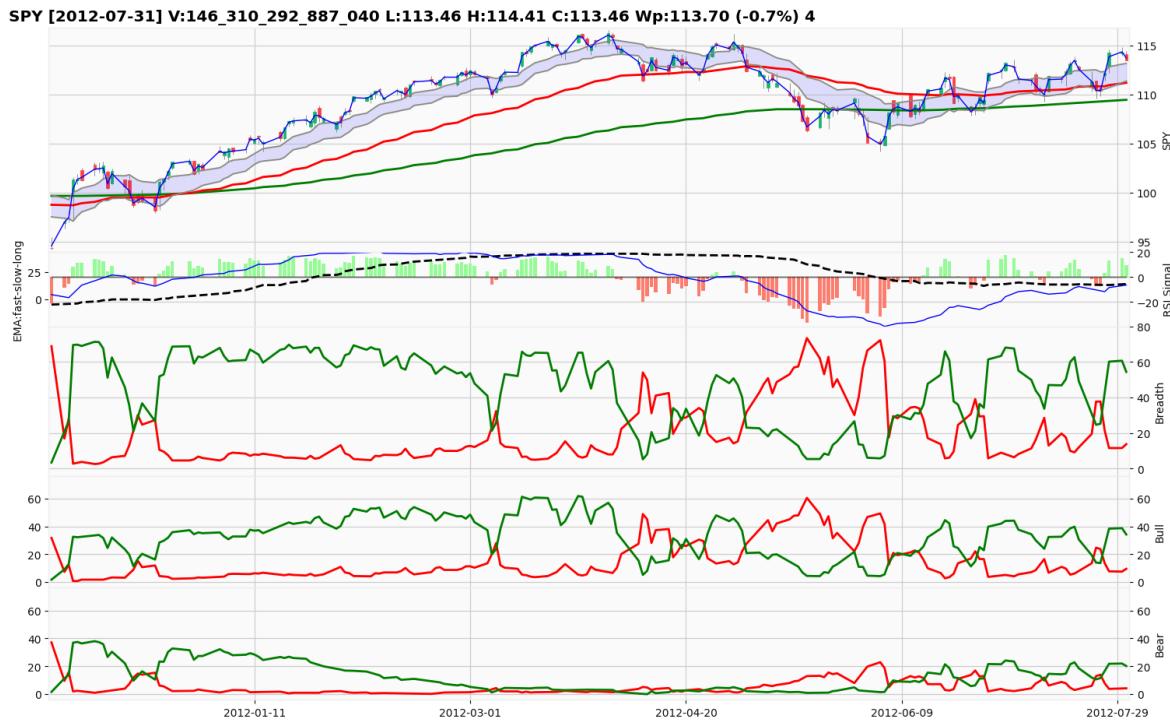
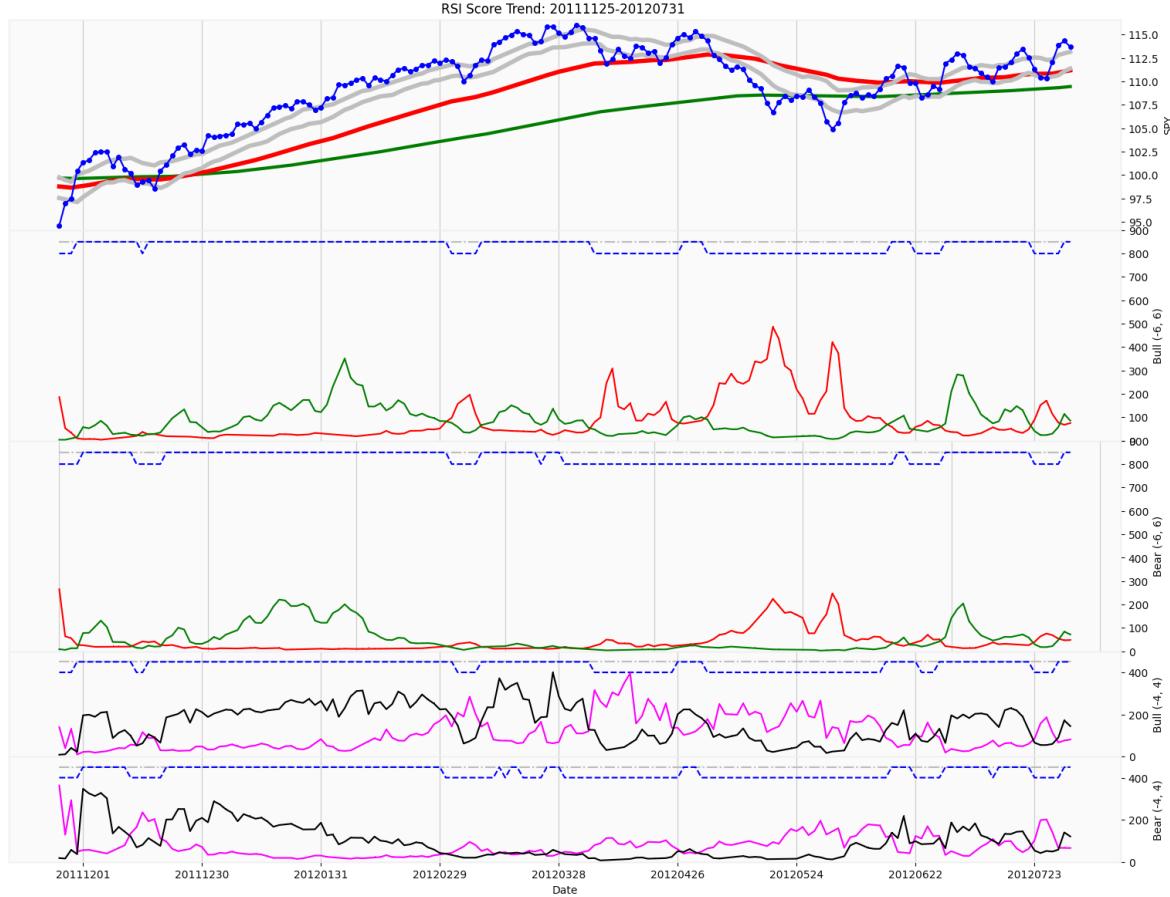


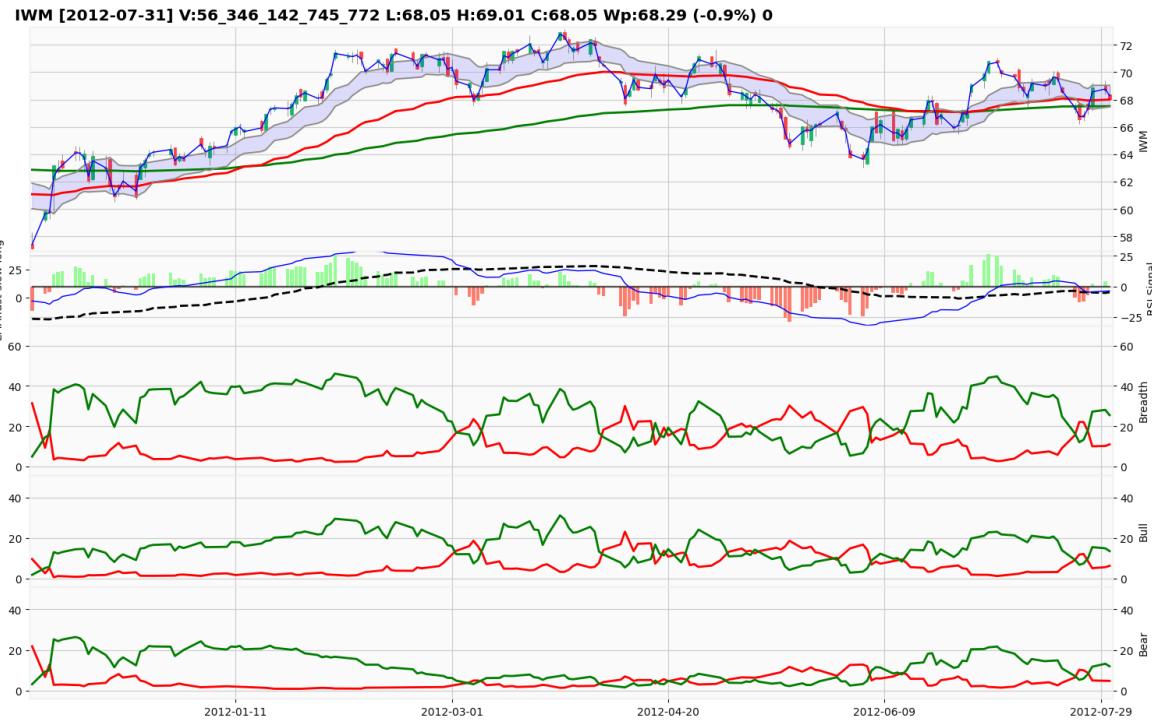
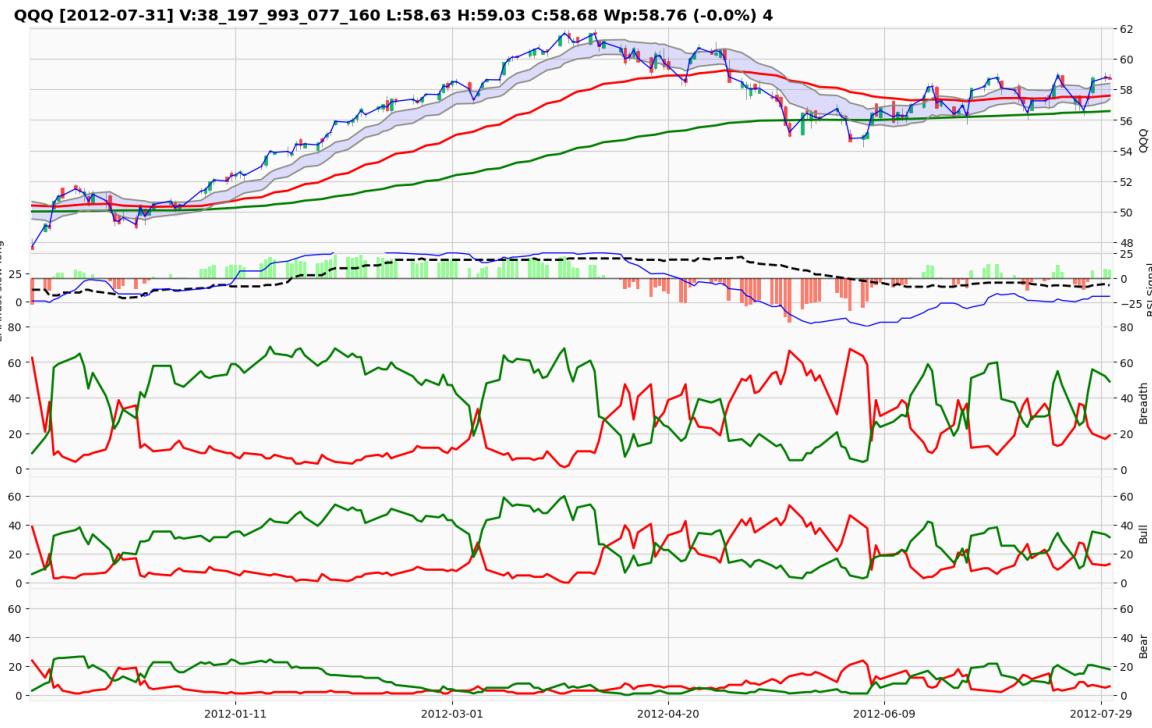
QQQ [2012-01-31] V:56\_992\_316\_046\_799 L:54.25 H:54.90 C:54.60 Wp:54.59 (0.1%) 4

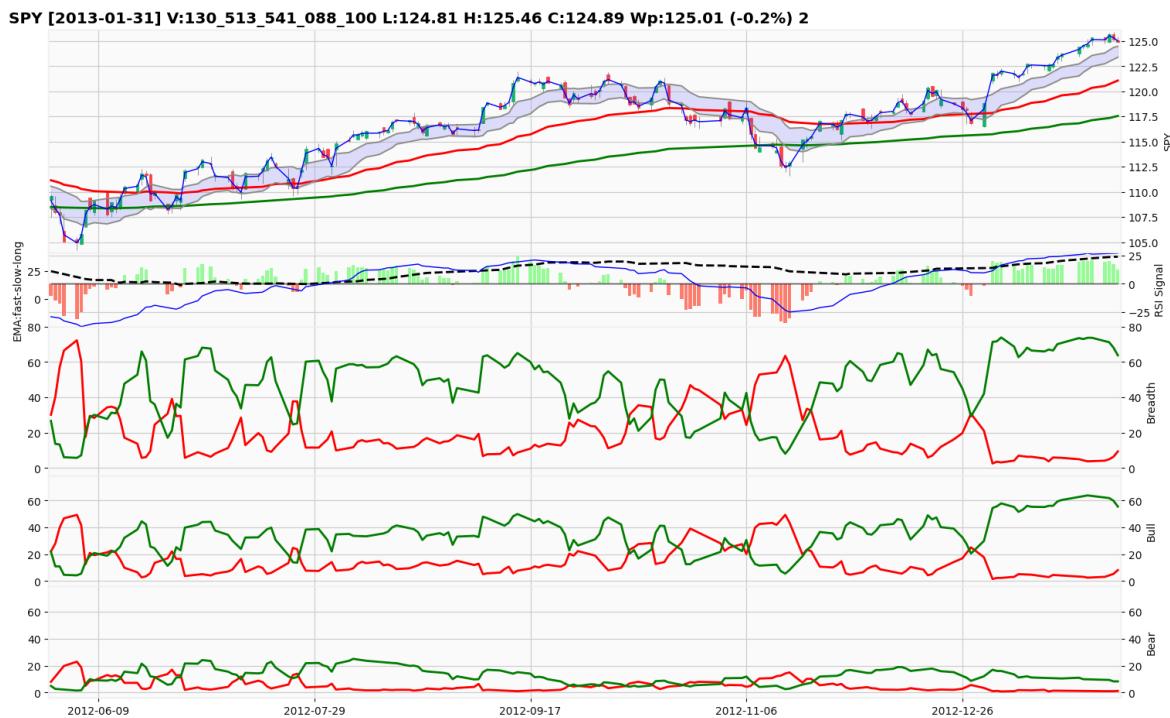
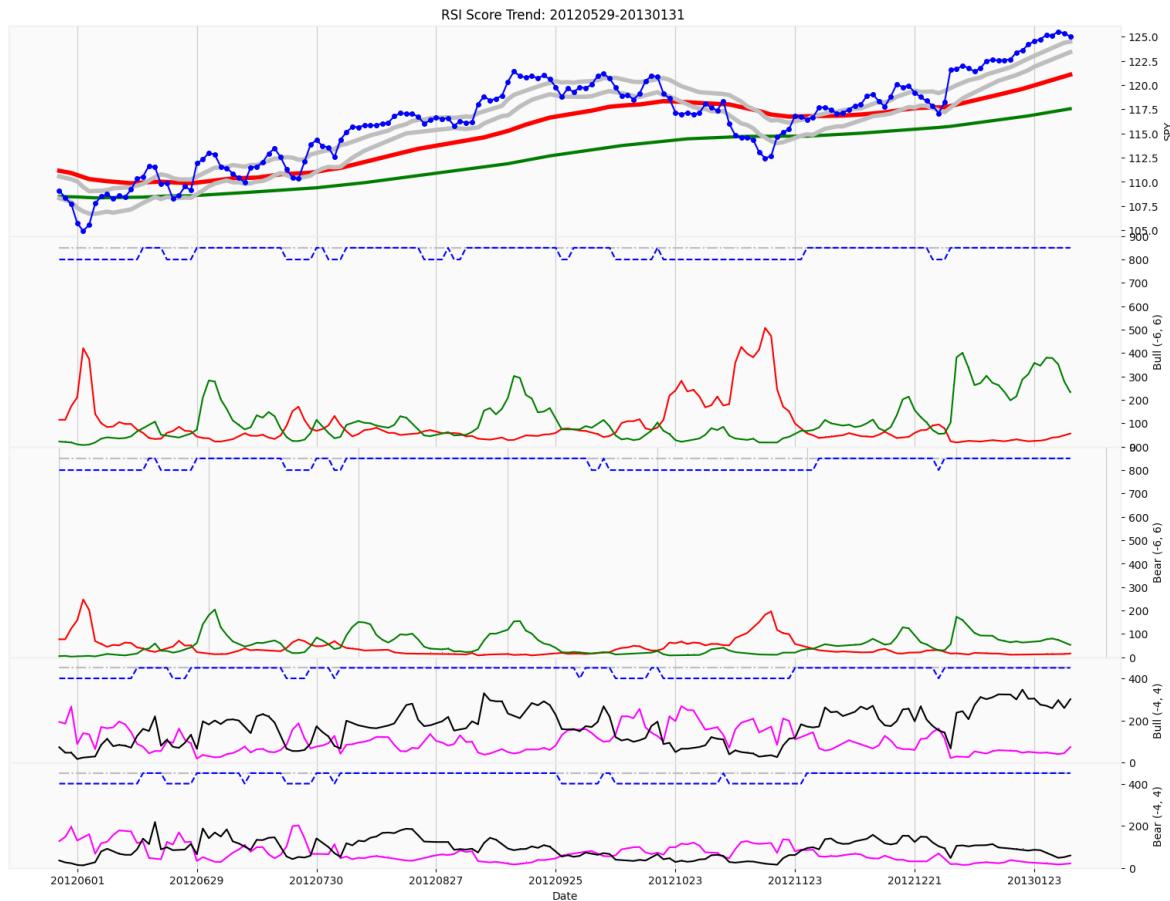


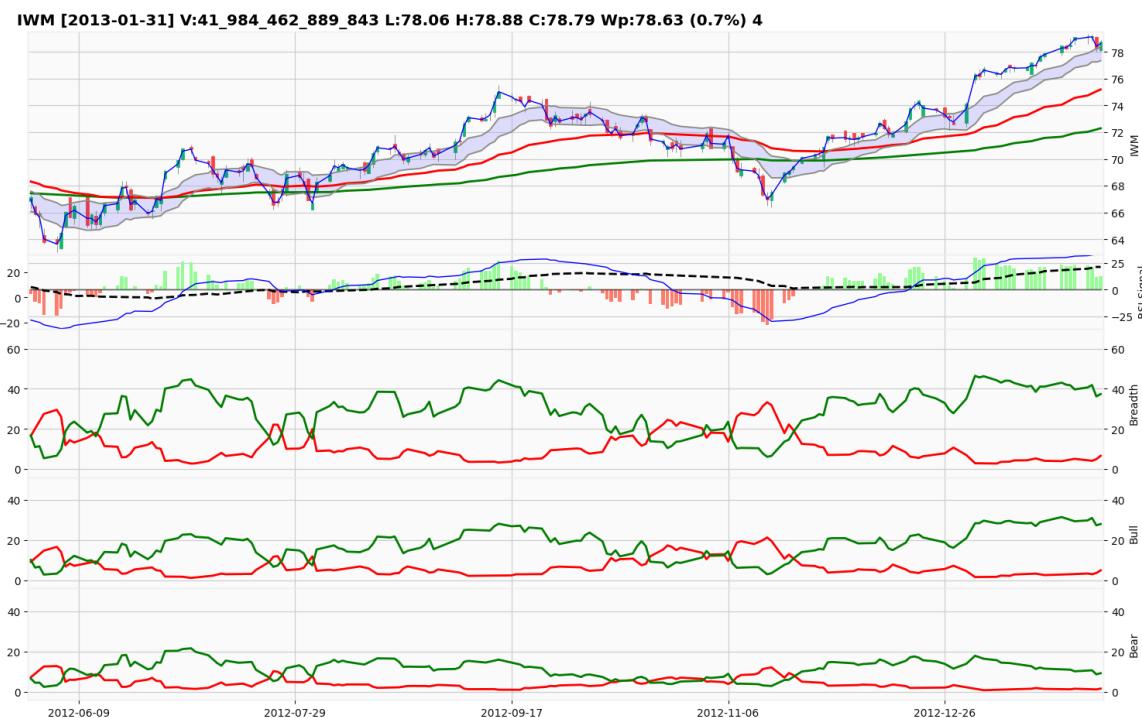
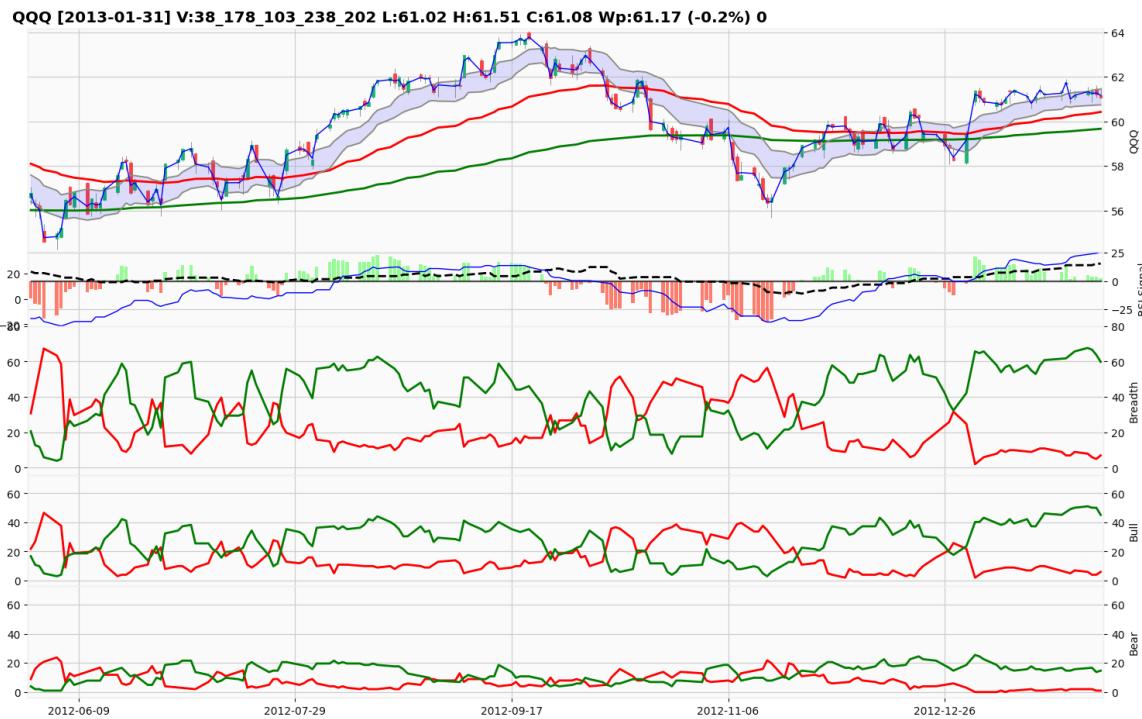
IWM [2012-01-31] V:60\_385\_482\_329\_824 L:67.70 H:68.77 C:68.08 Wp:68.16 (-0.1%) 4

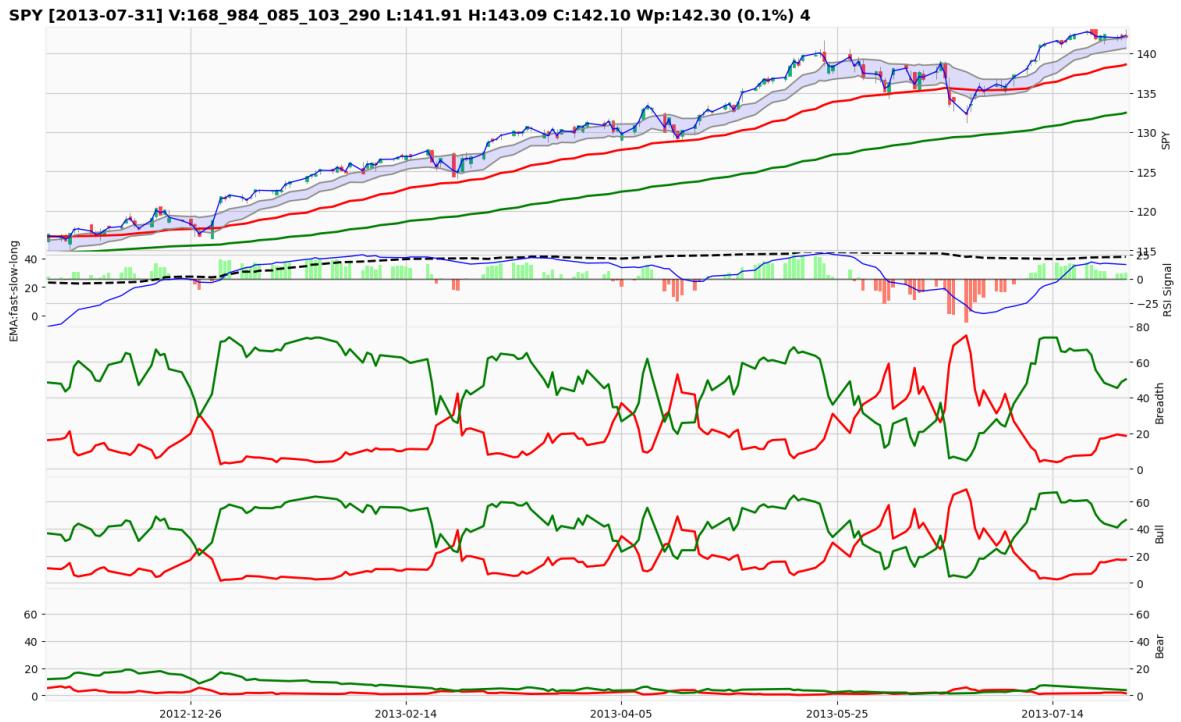
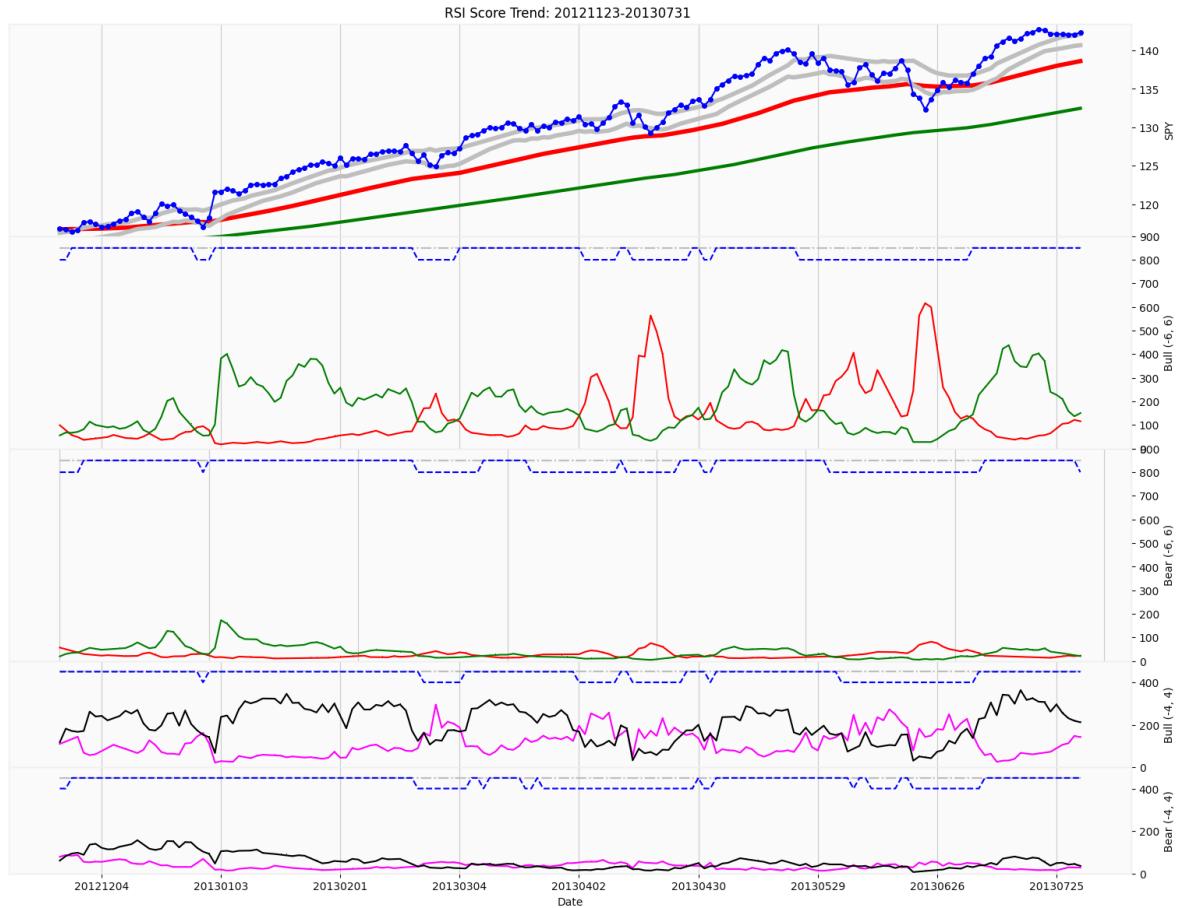


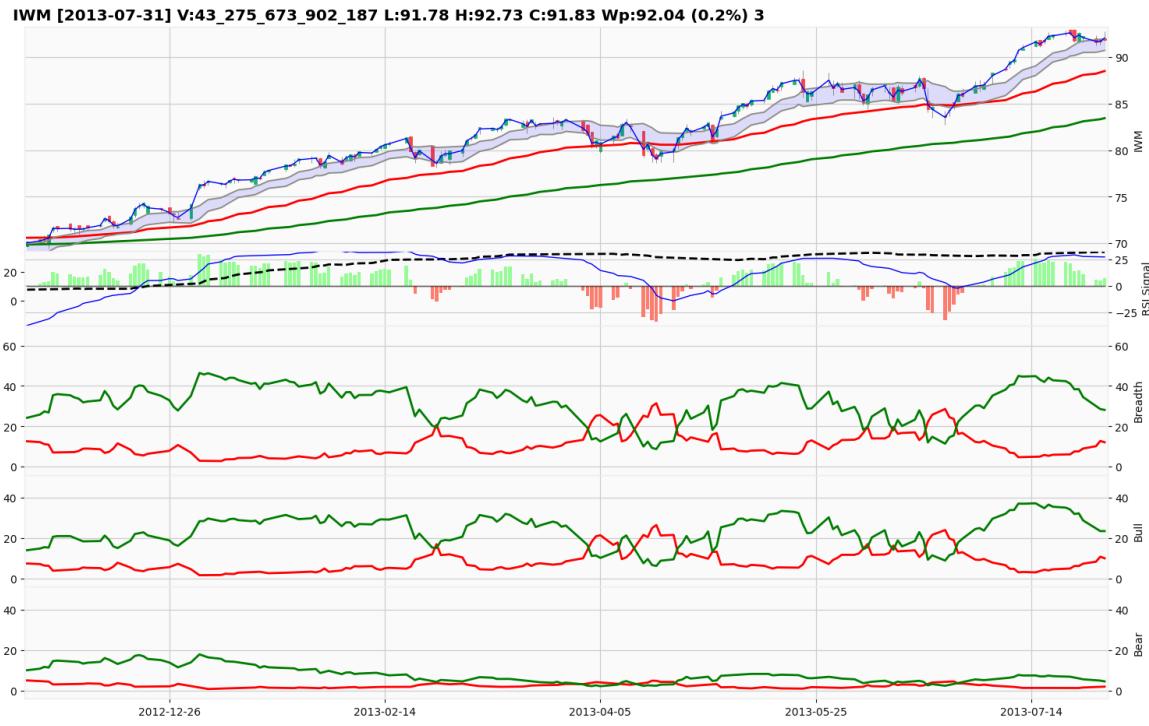
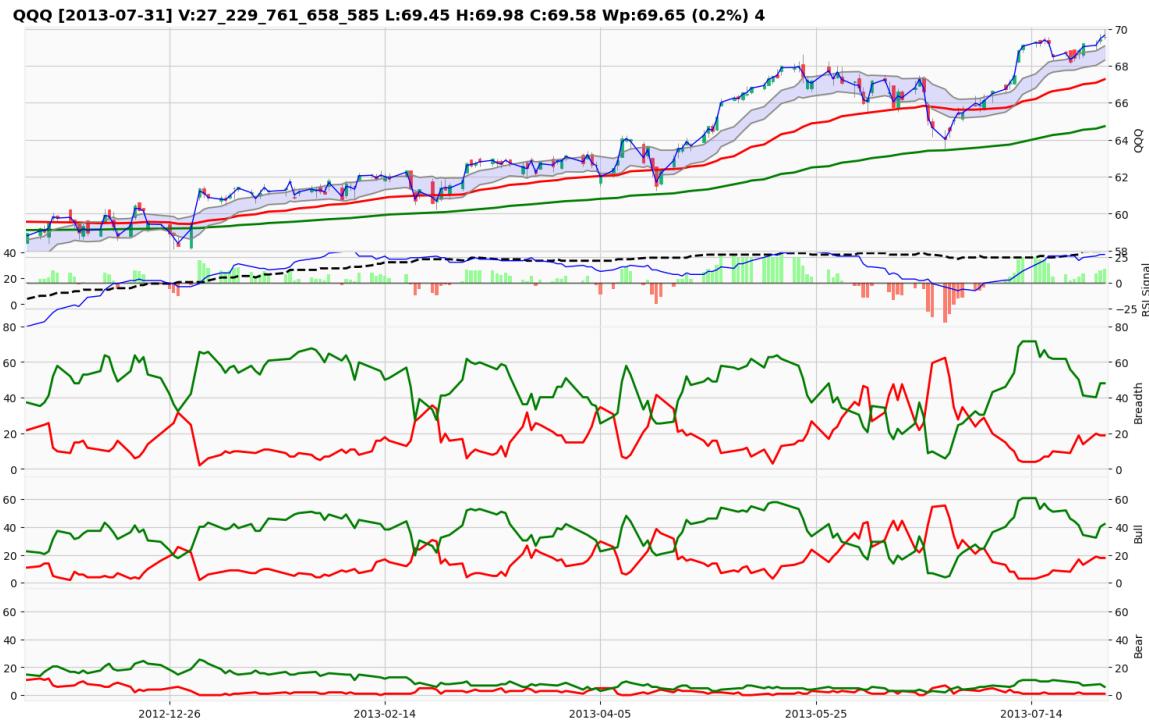


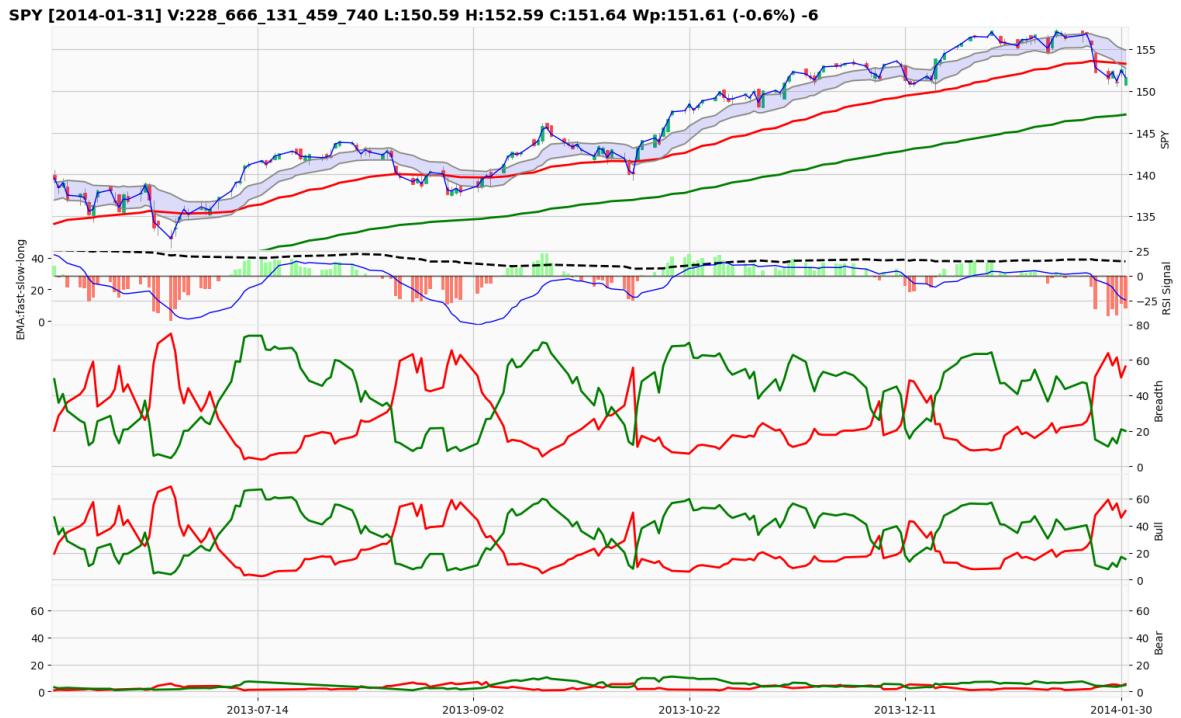
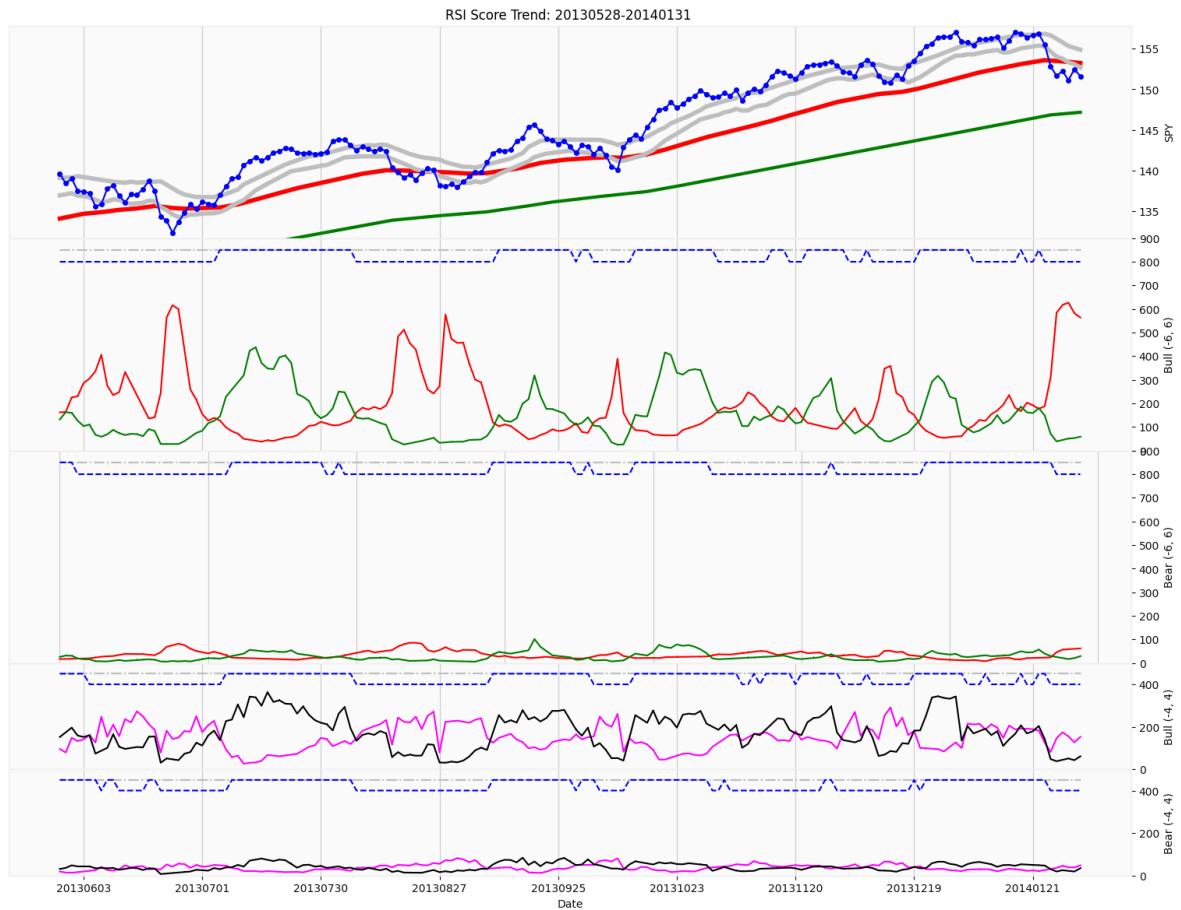


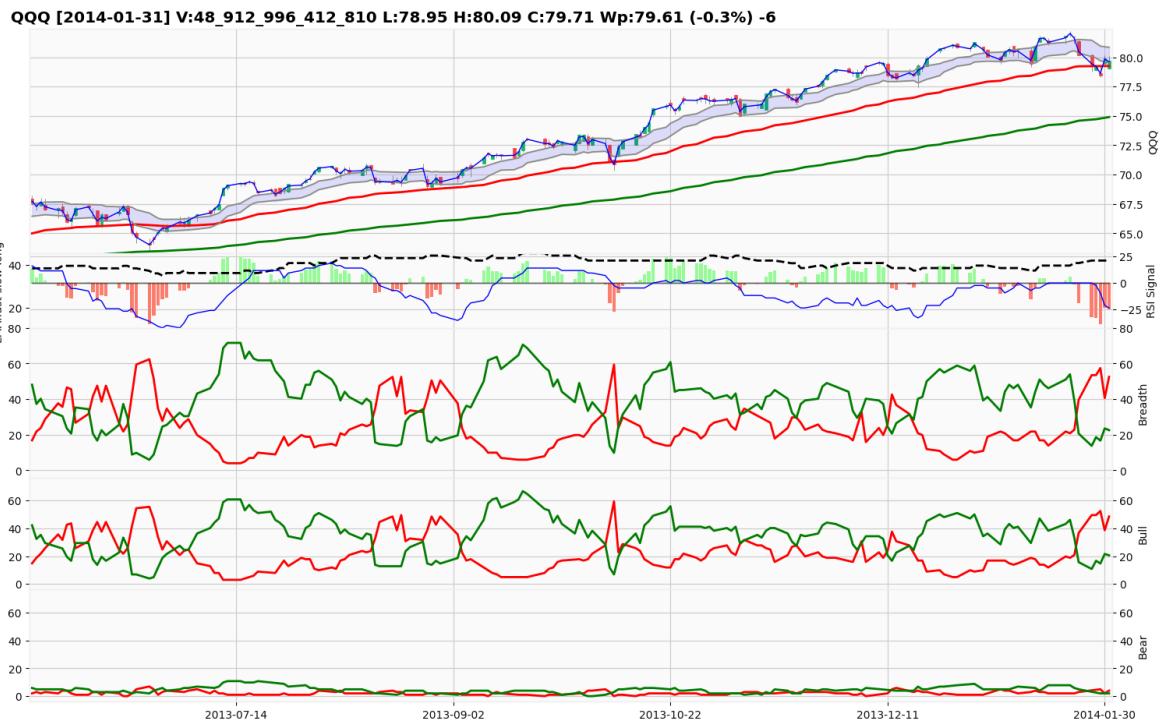


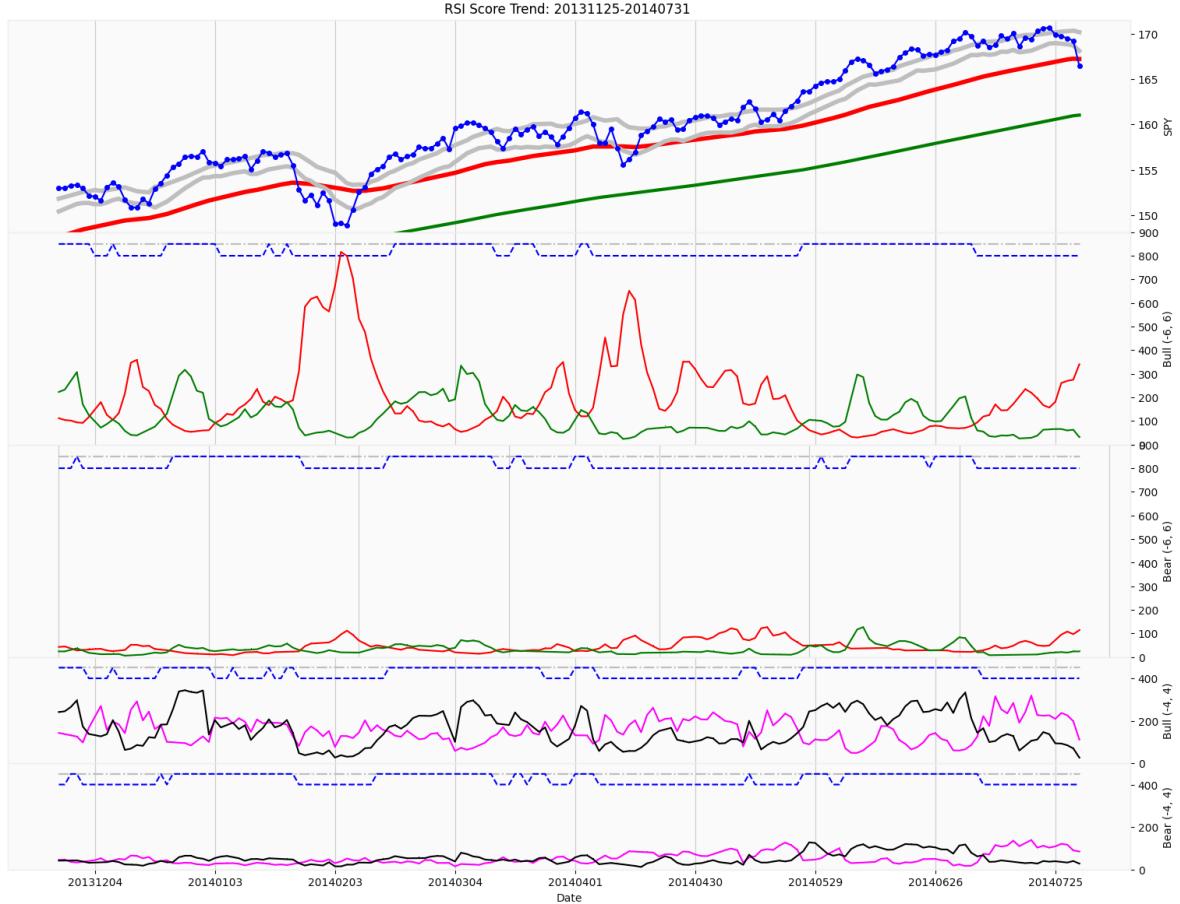




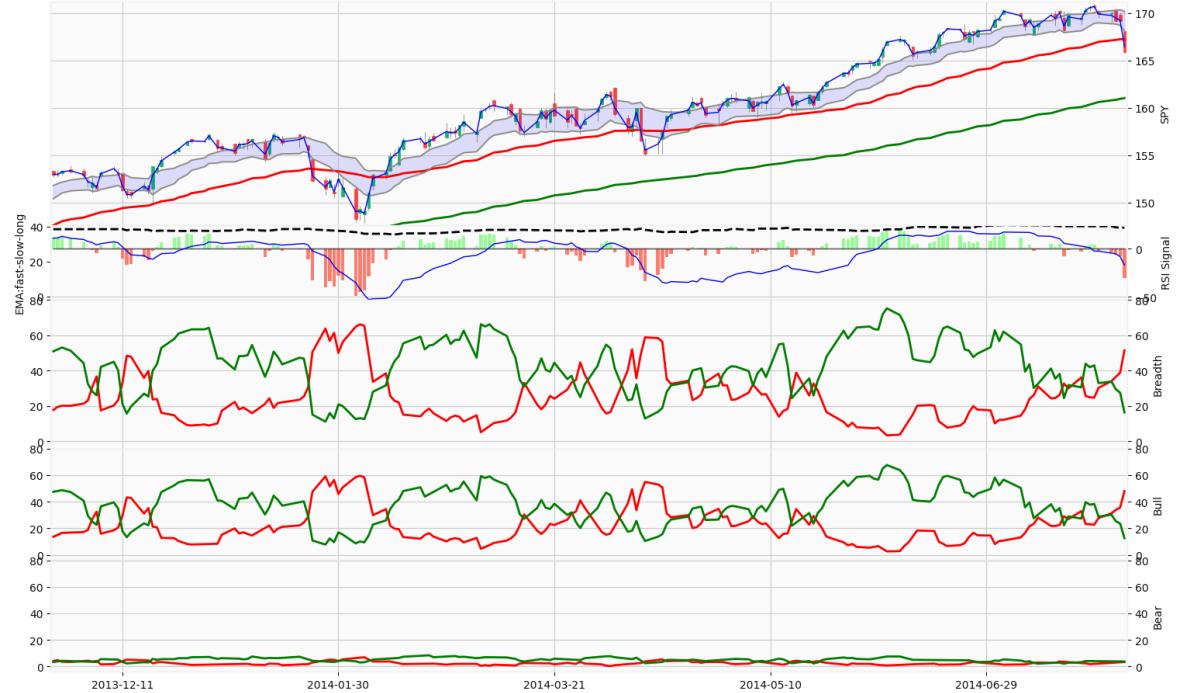




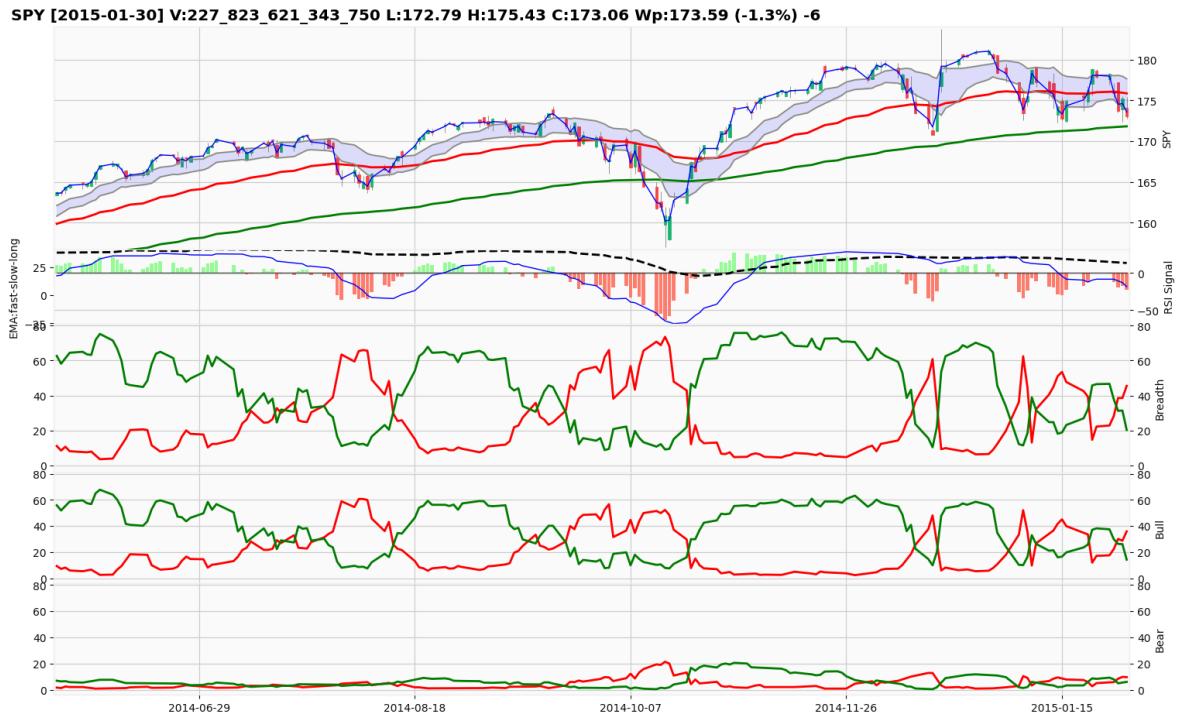
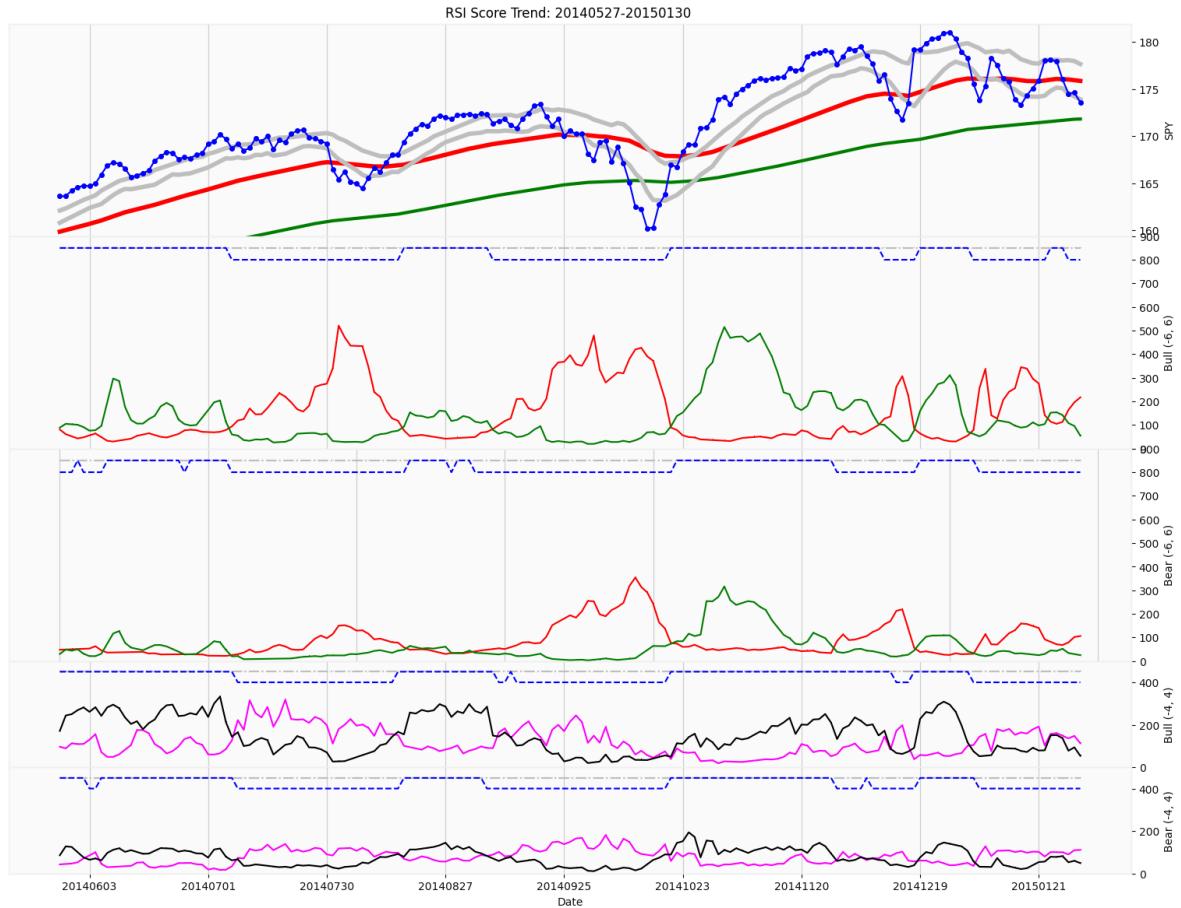


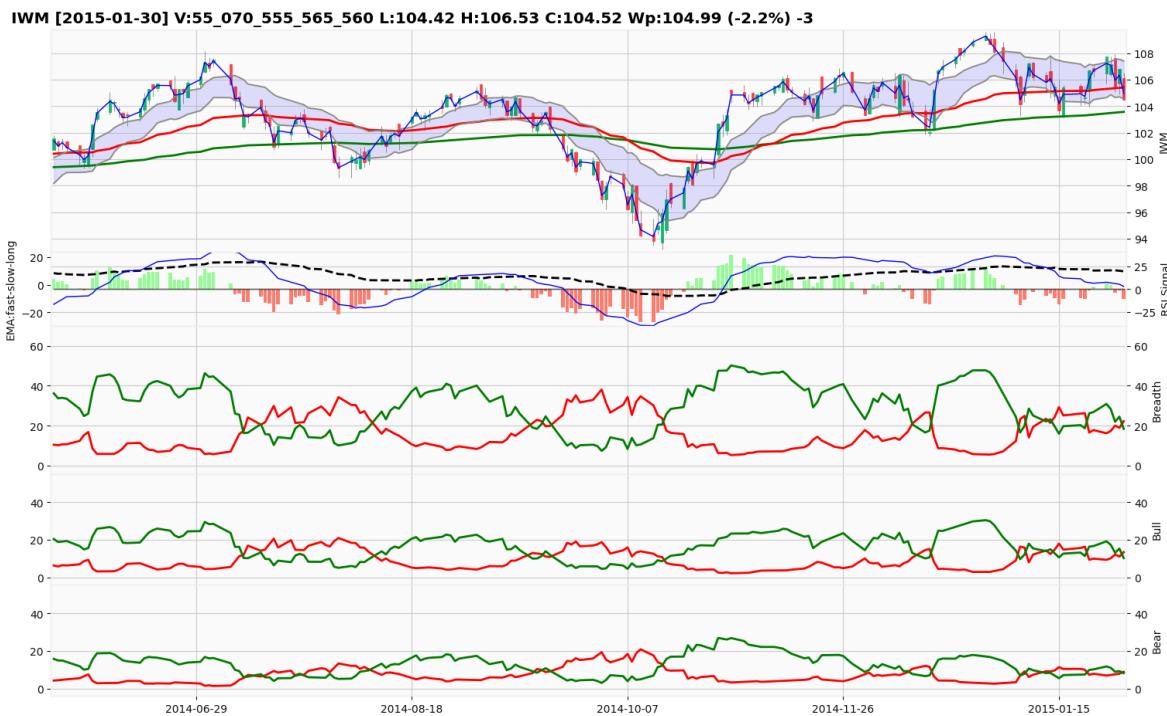
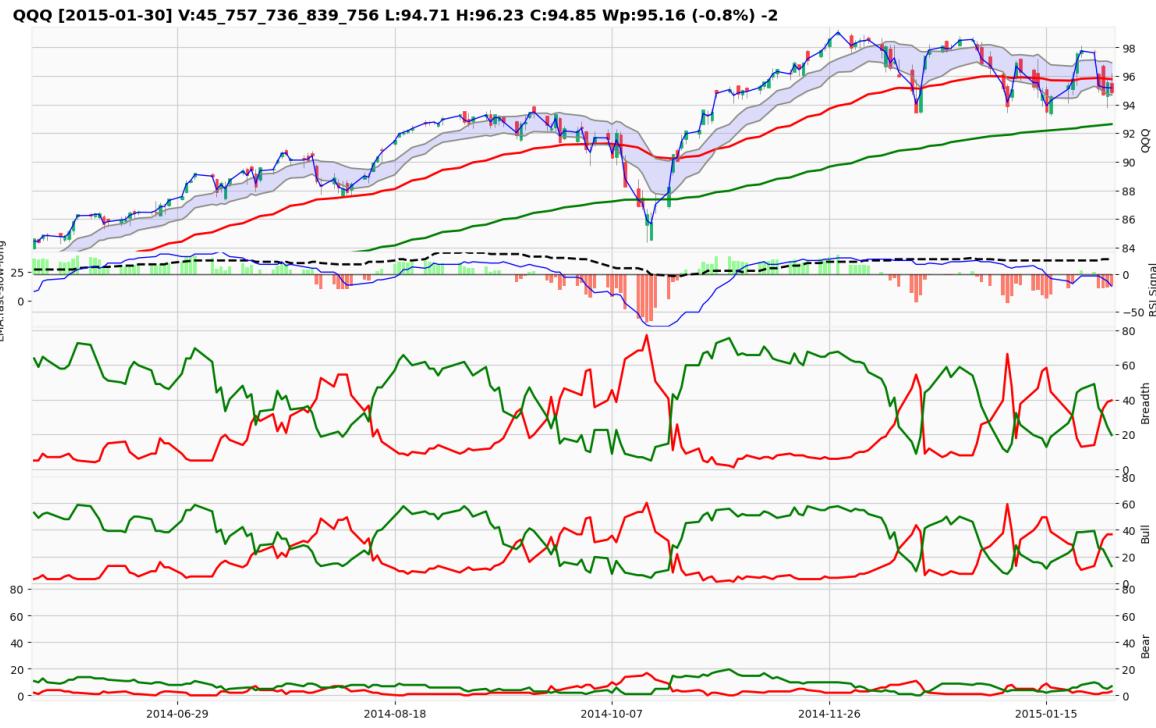


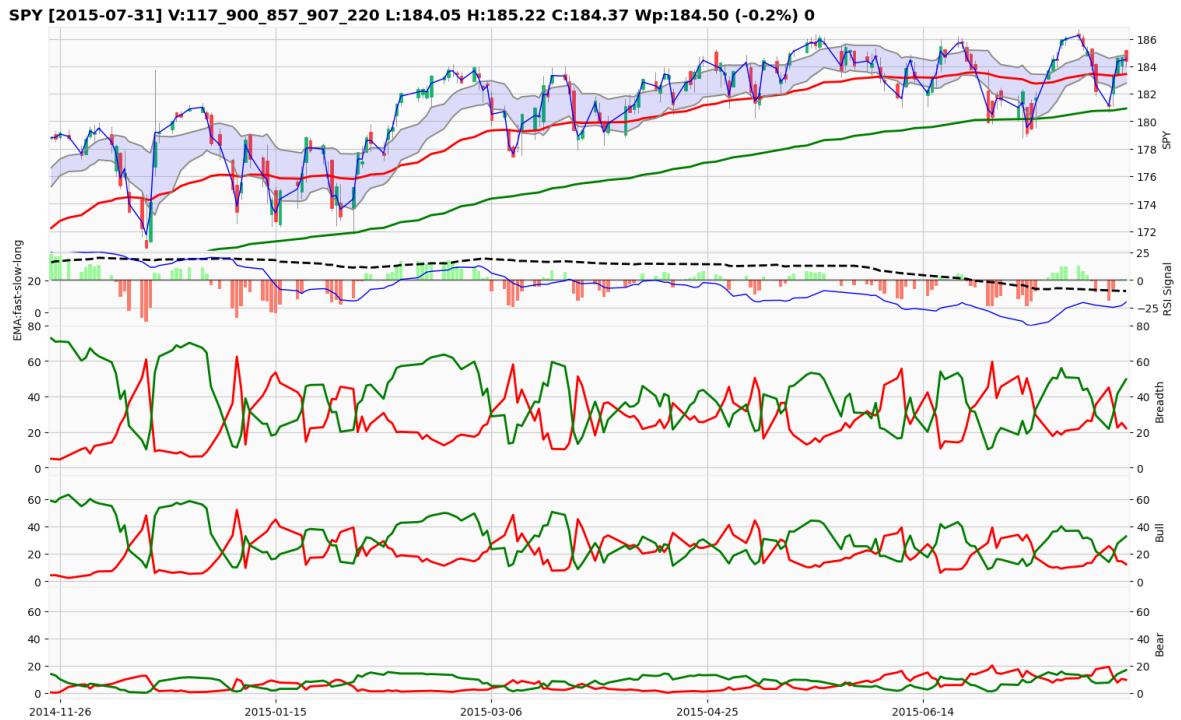
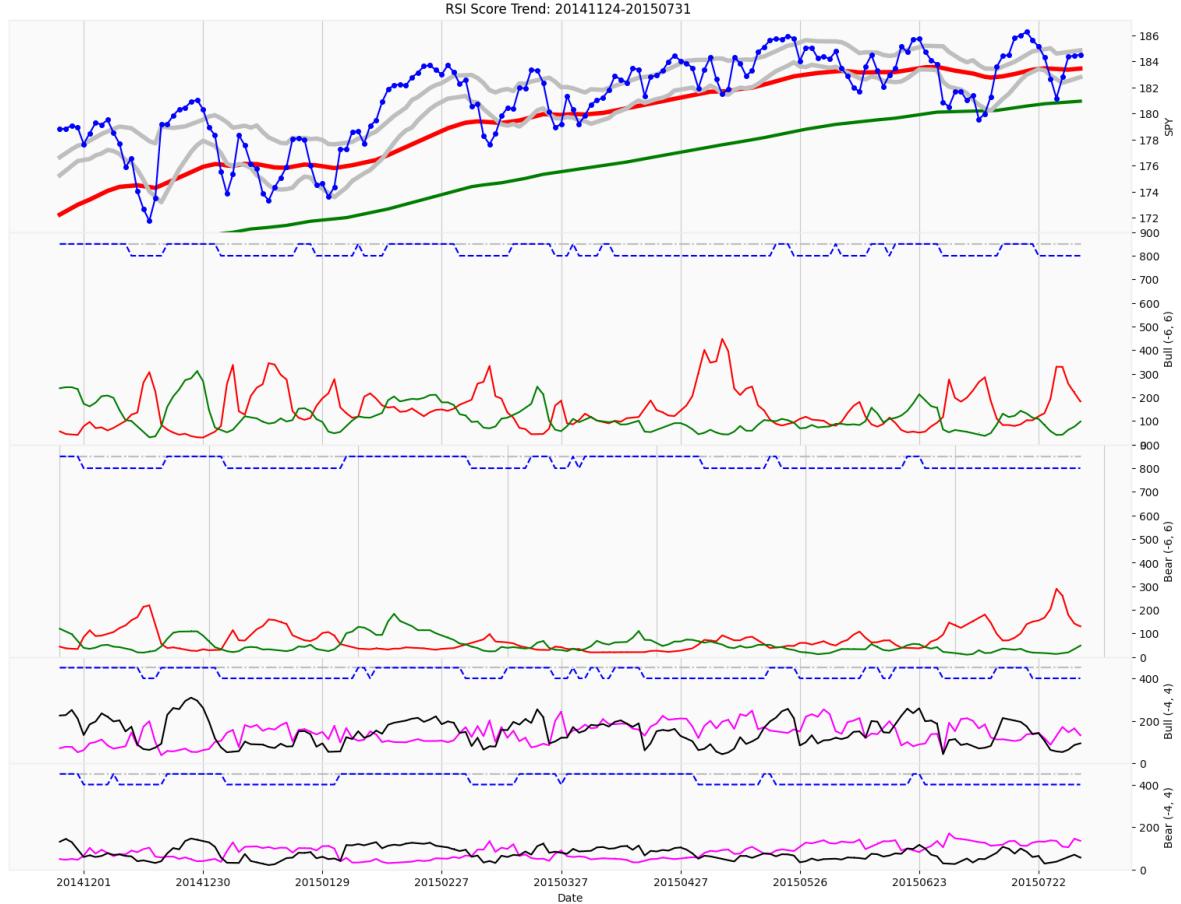
SPY [2014-07-31] V:213\_412\_004\_868\_200 L:165.79 H:168.20 C:165.90 Wp:166.44 (-1.9%) -5

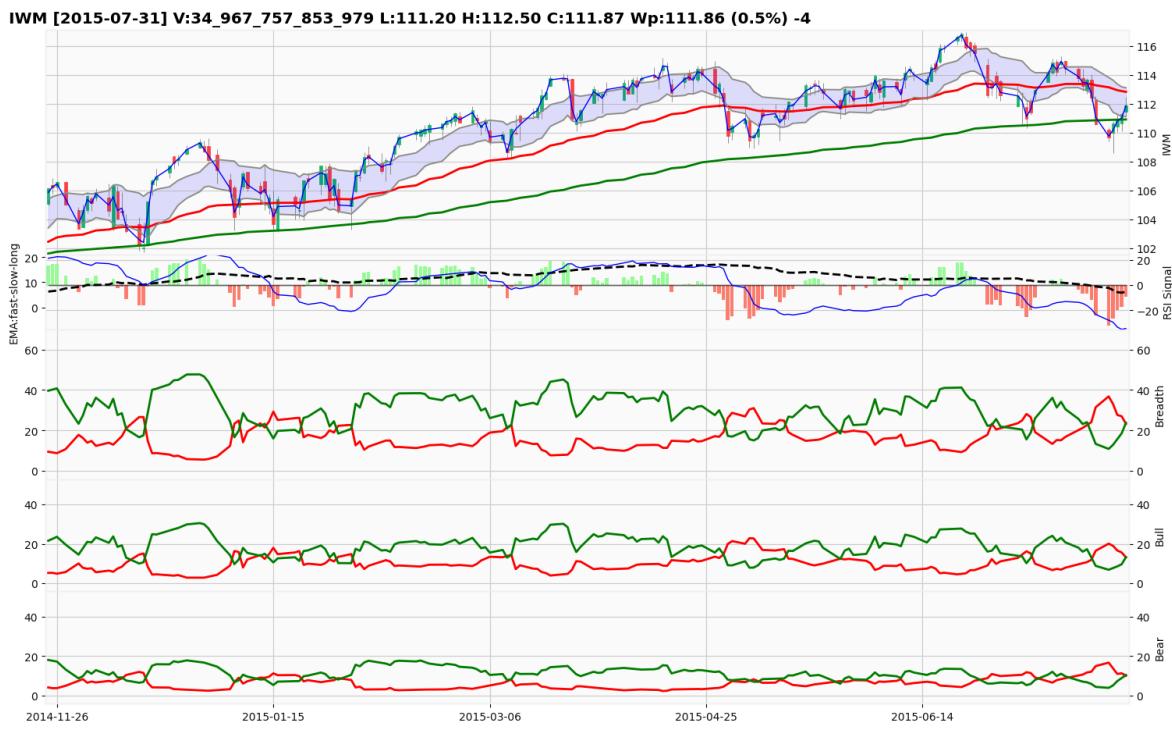


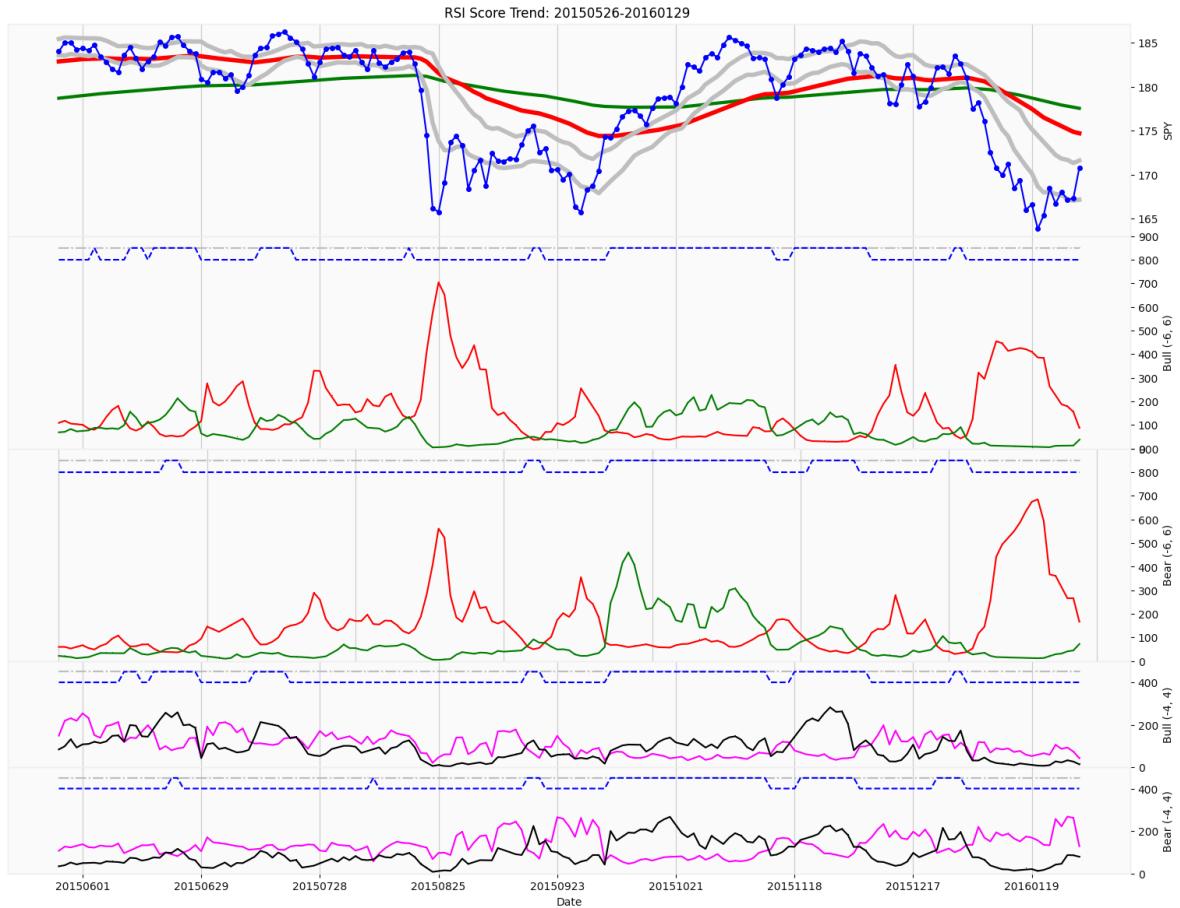






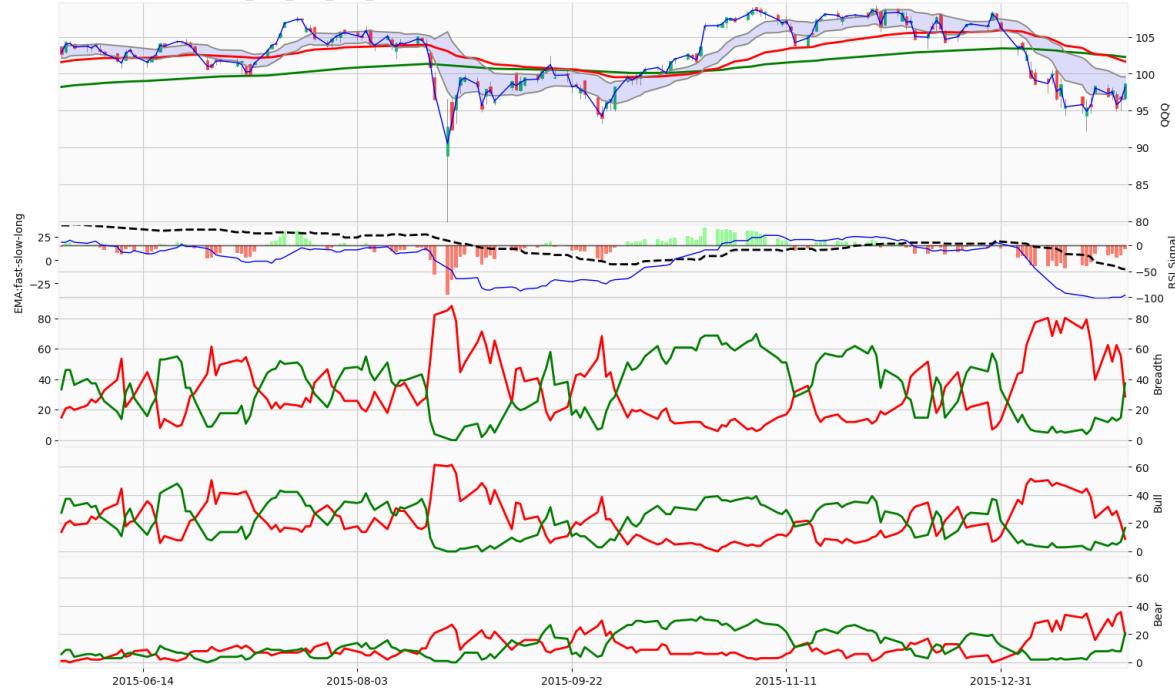
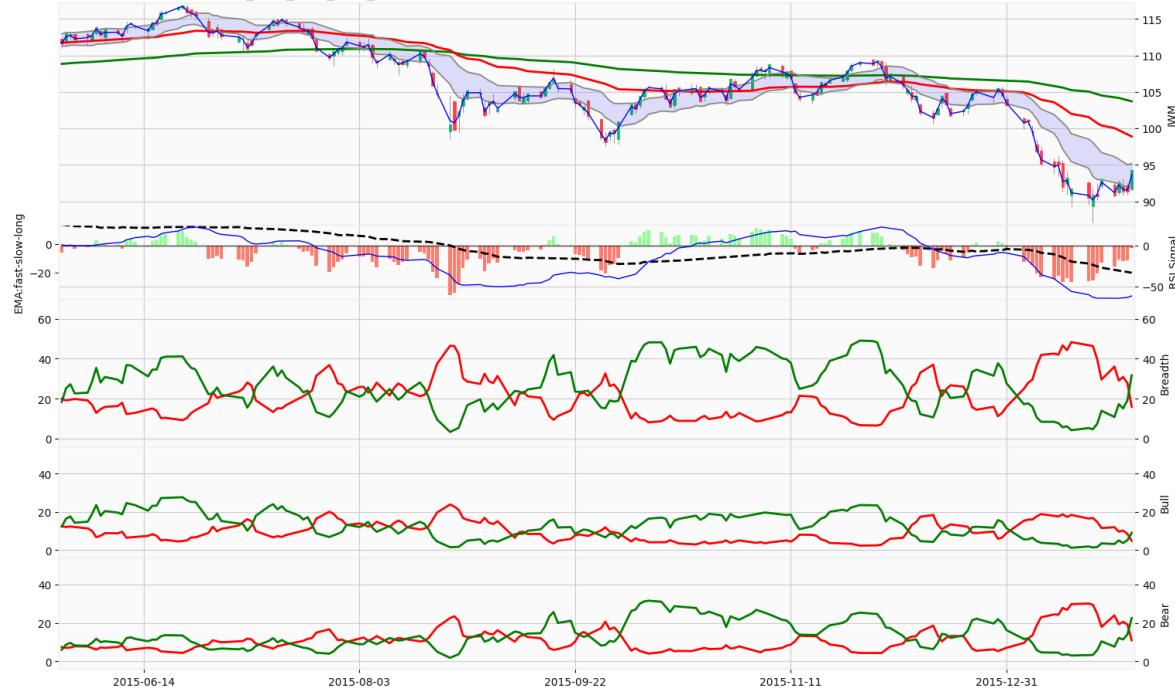


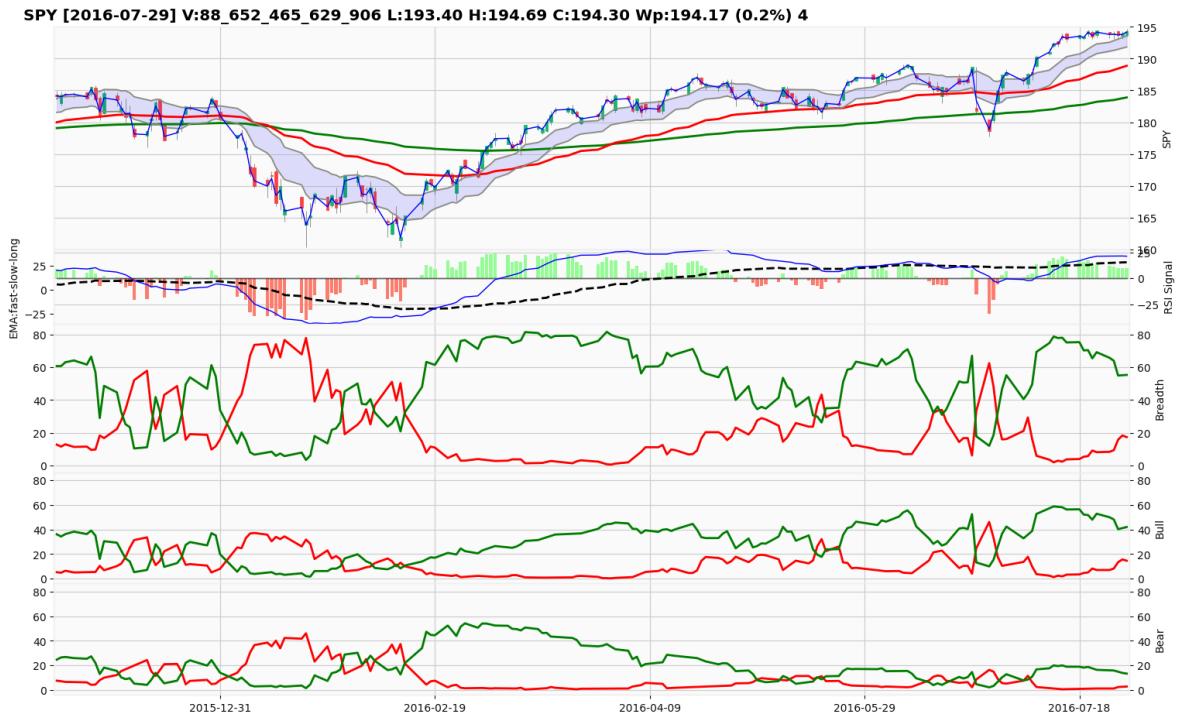
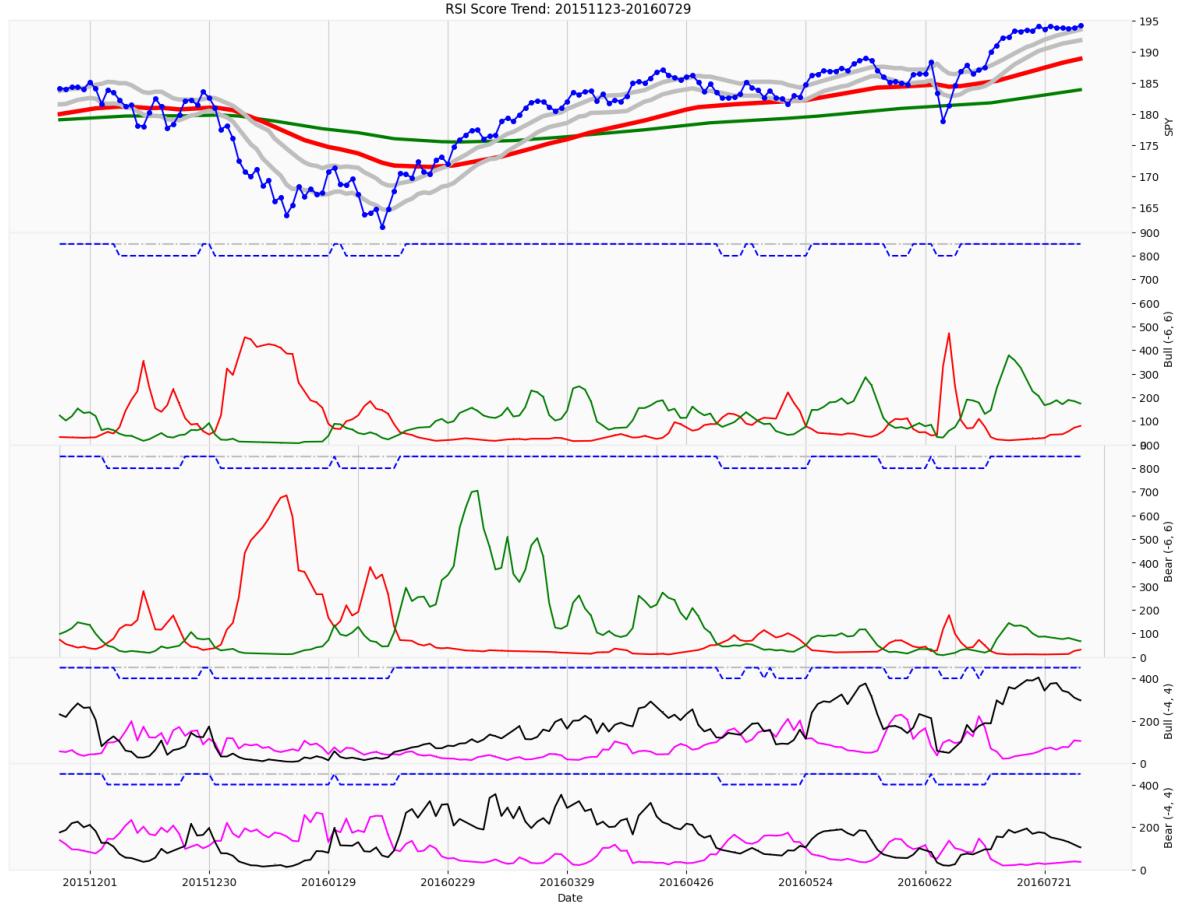


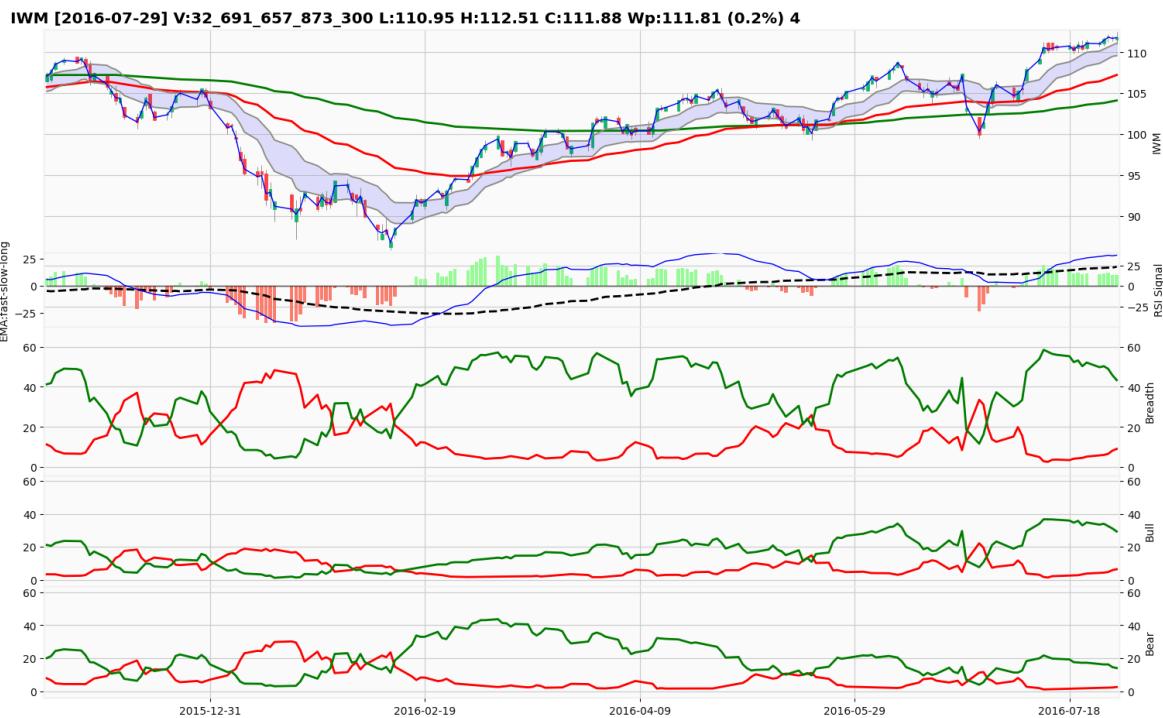
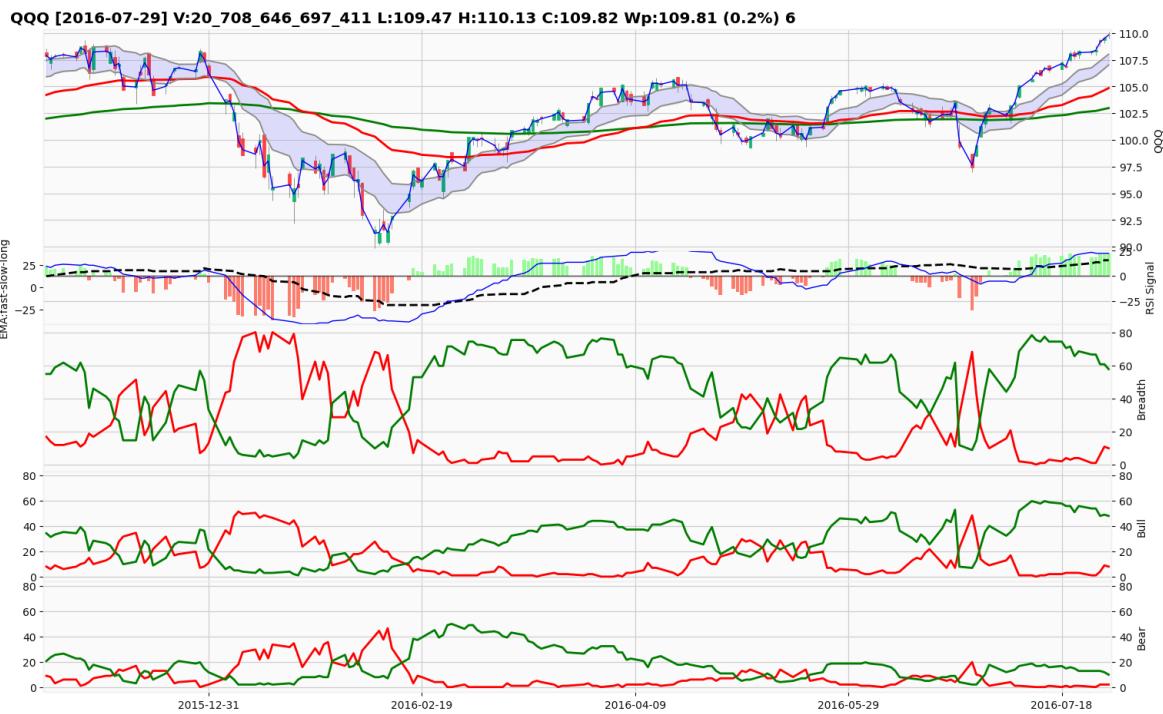


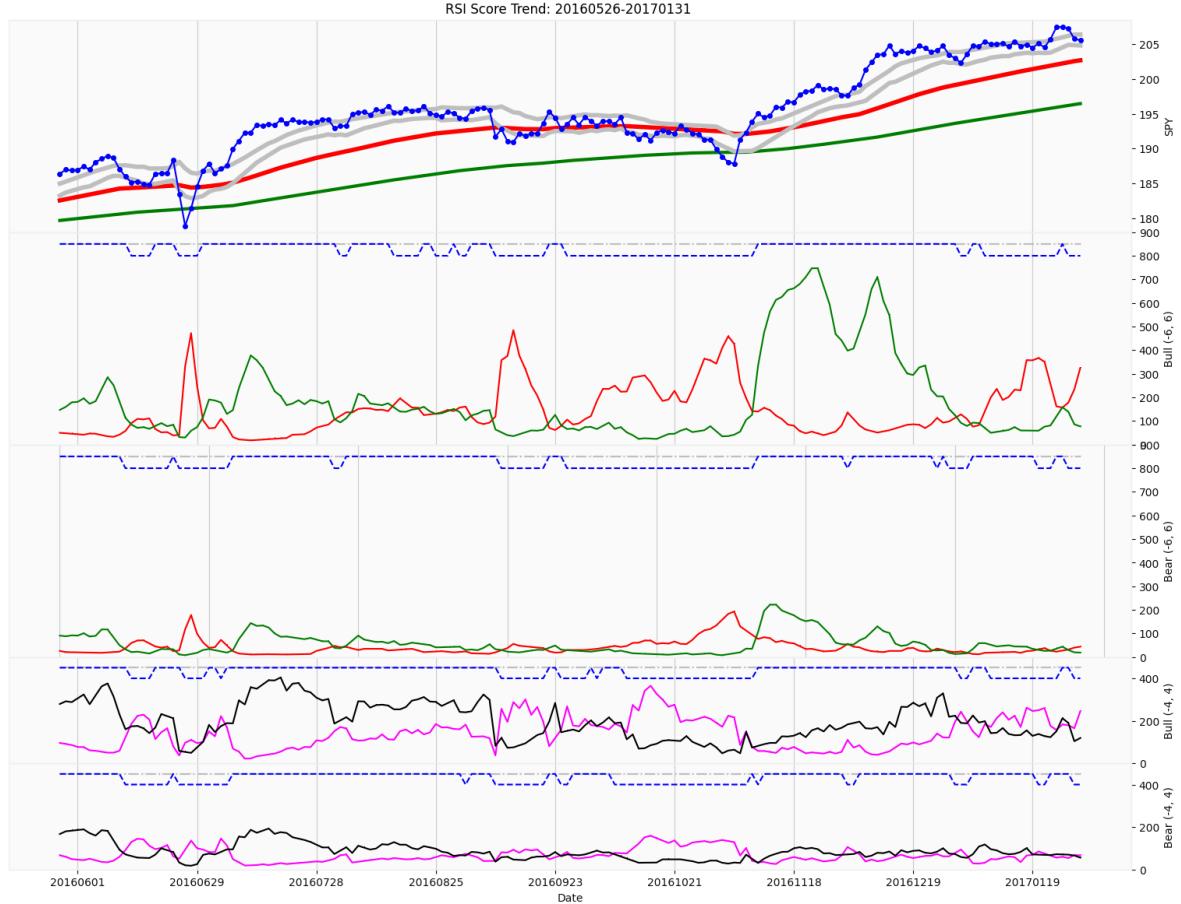
**SPY [2016-01-29] V:237\_439\_627\_669\_800 L:168.19 H:171.74 C:171.58 Wp:170.77 (2.4%) -1**



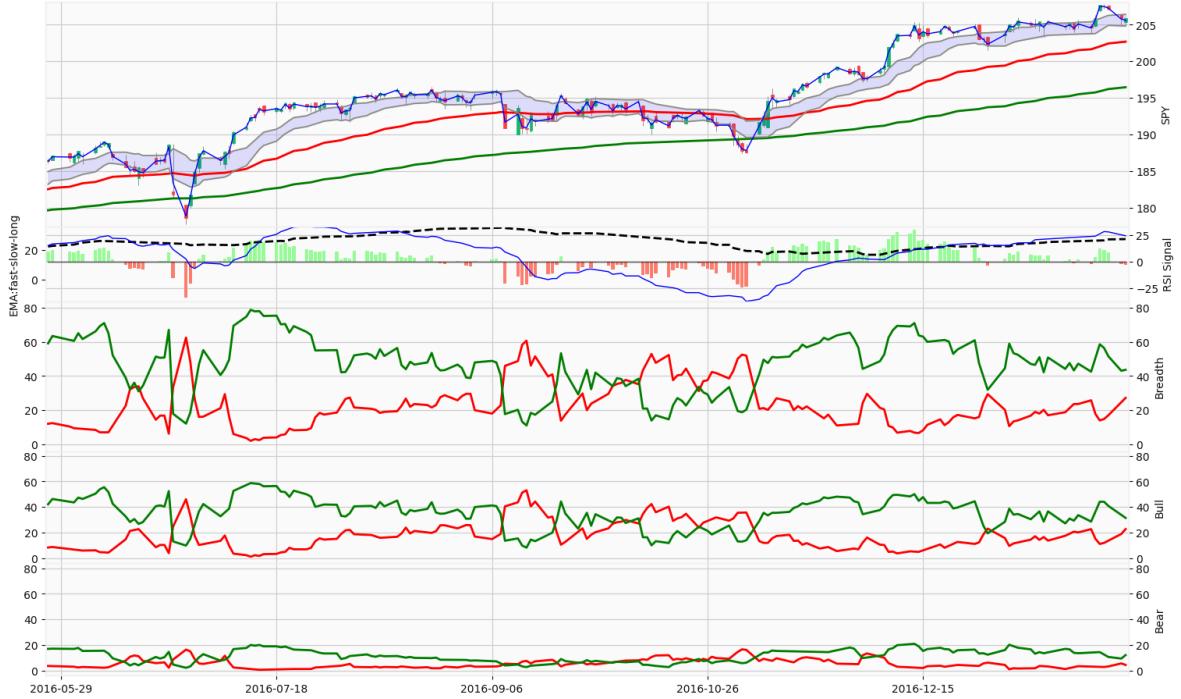
**QQQ [2016-01-29] V:45\_898\_707\_177\_475 L:96.61 H:98.72 C:98.69 Wp:98.18 (2.1%) -2****IWM [2016-01-29] V:46\_087\_612\_973\_743 L:91.60 H:94.36 C:94.34 Wp:93.66 (3.3%) -1**

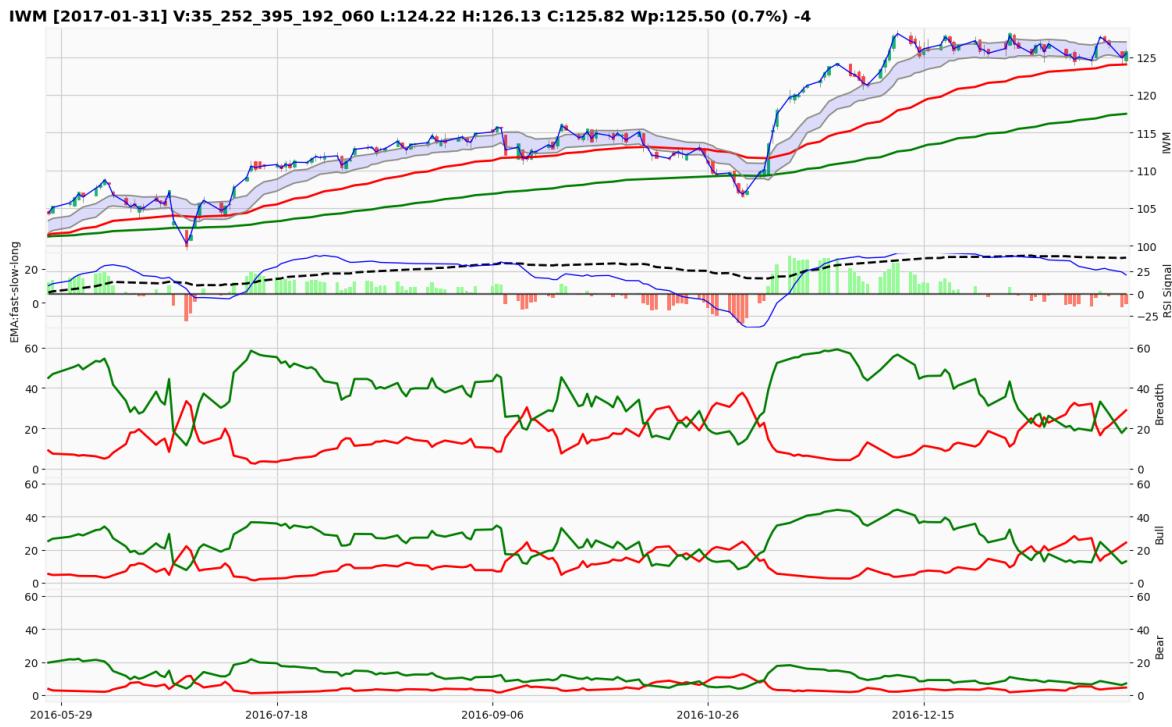
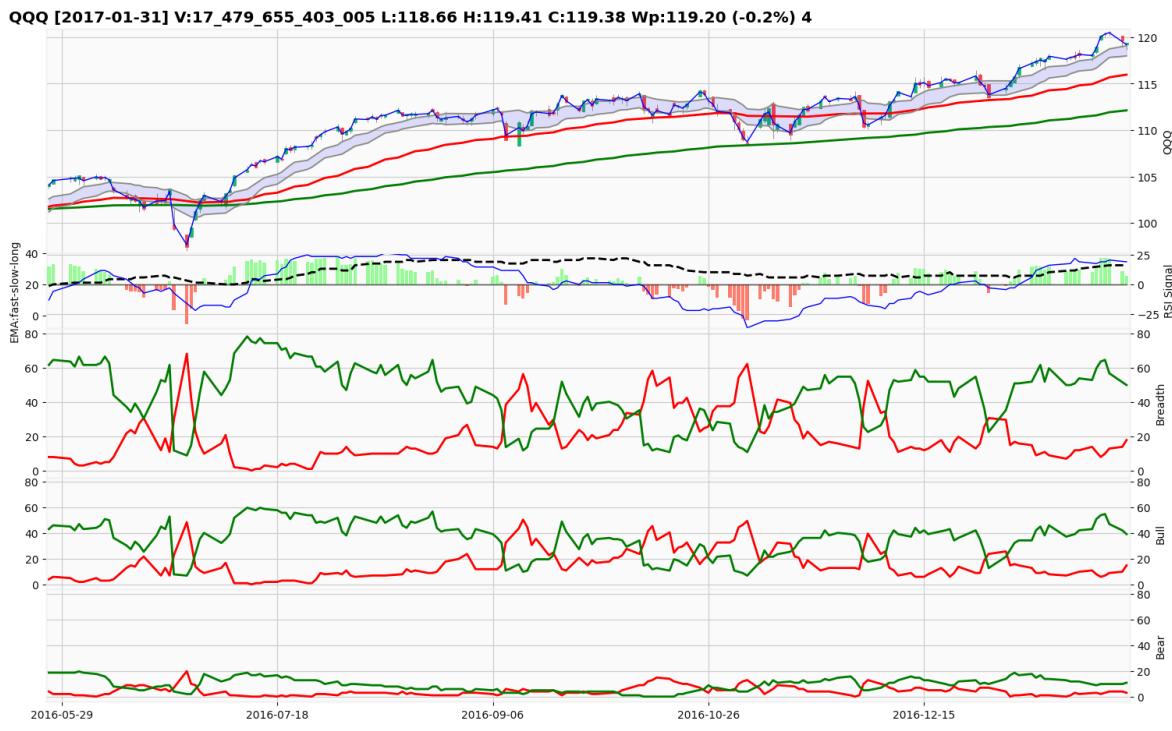


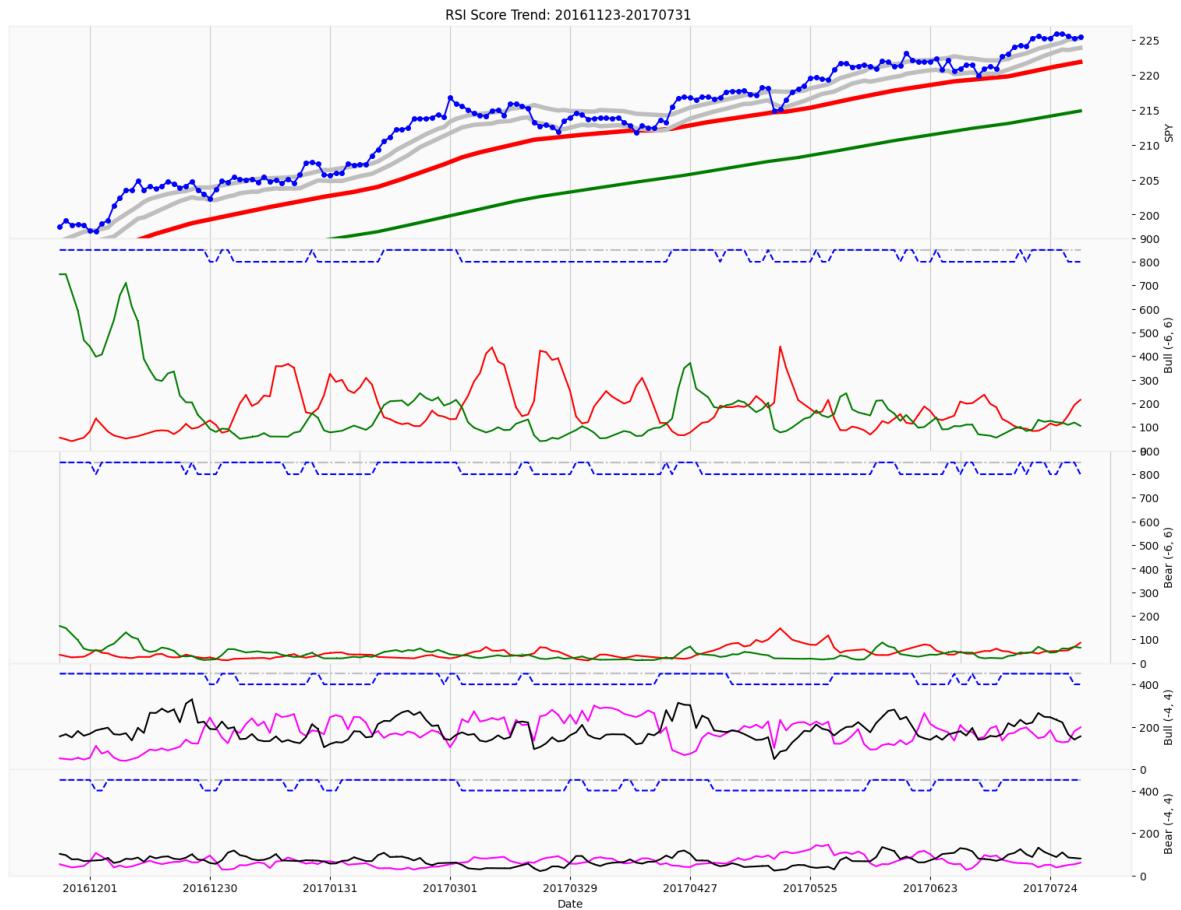




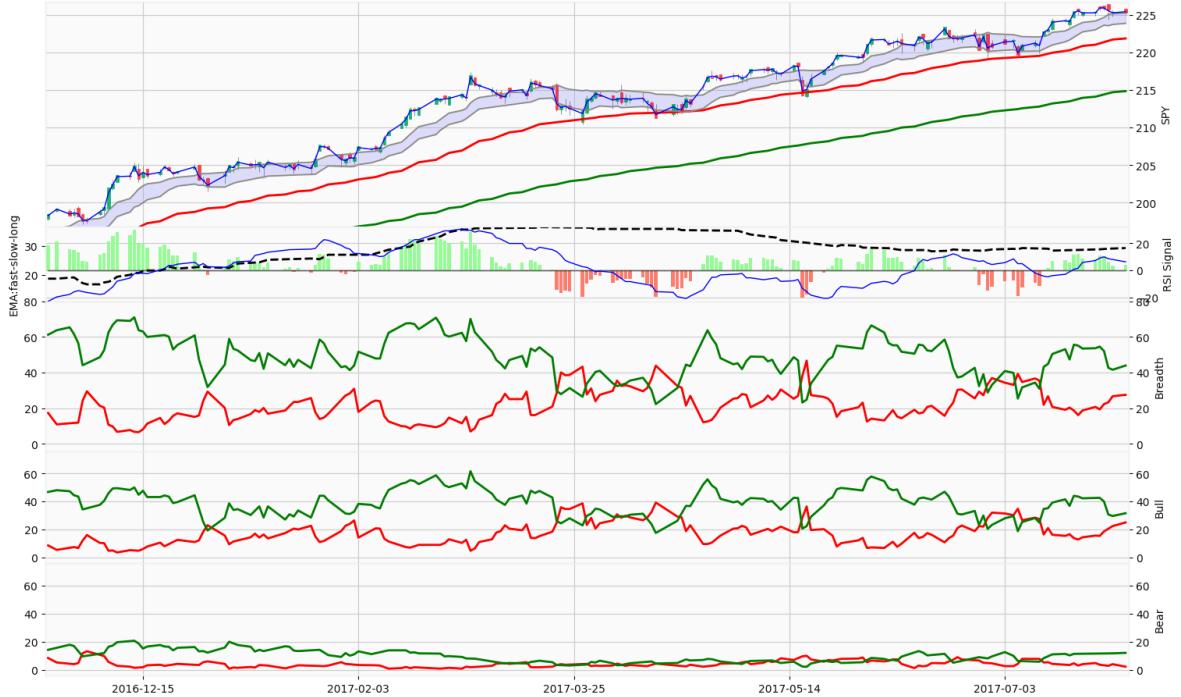
**SPY [2017-01-31] V:83\_323\_305\_440\_129 L:204.78 H:205.91 C:205.84 Wp:205.59 (-0.0%) 0**

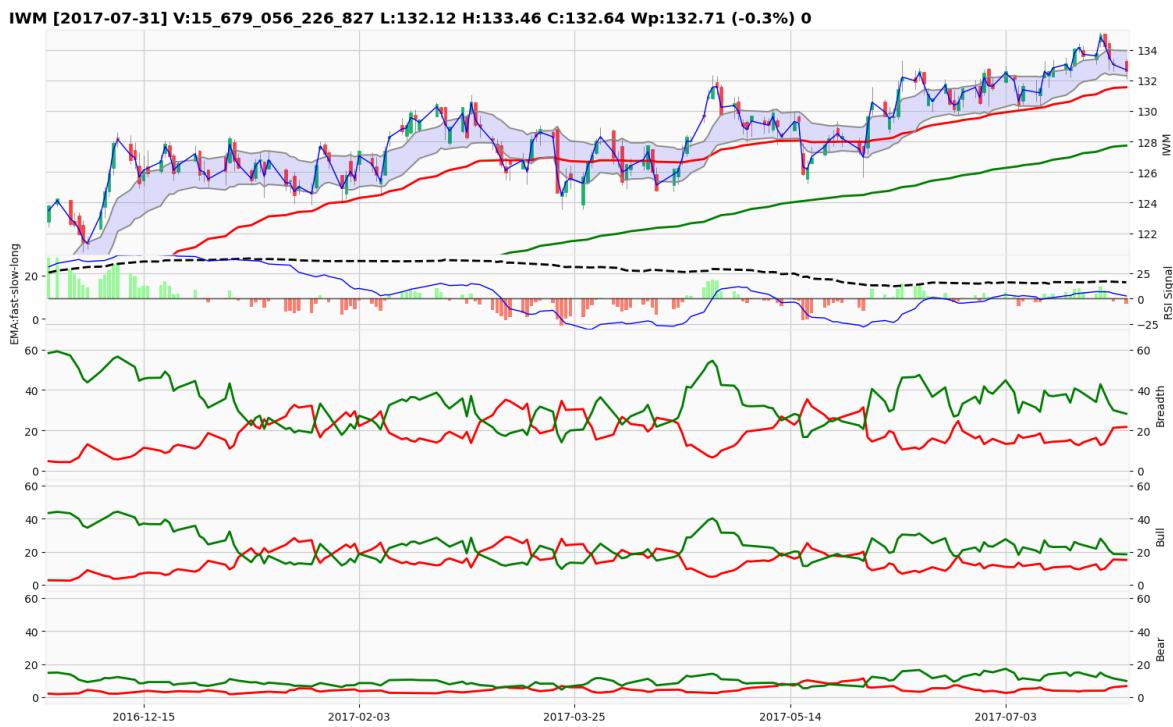
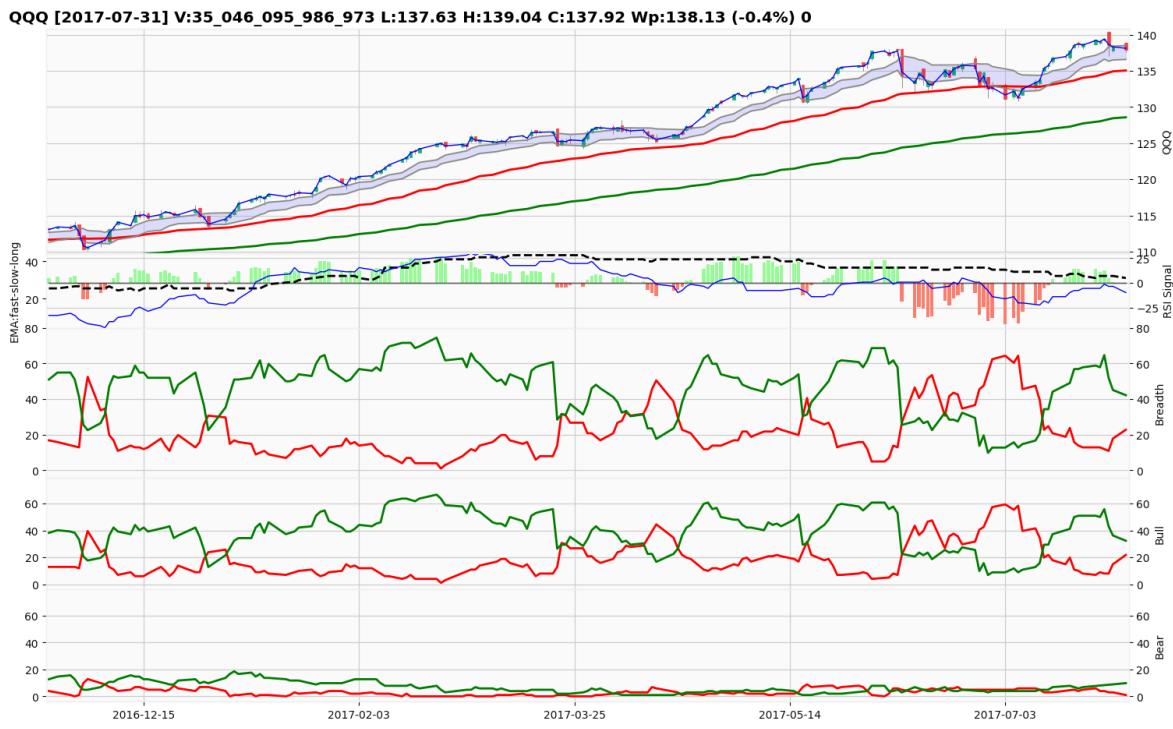


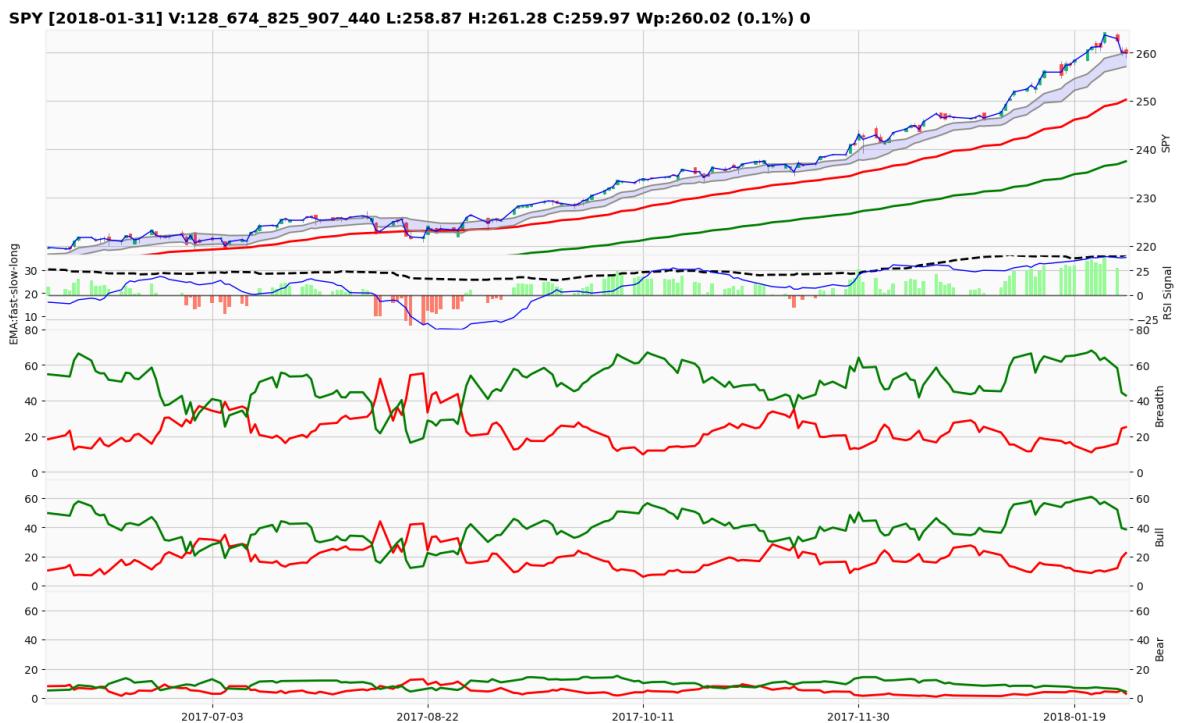
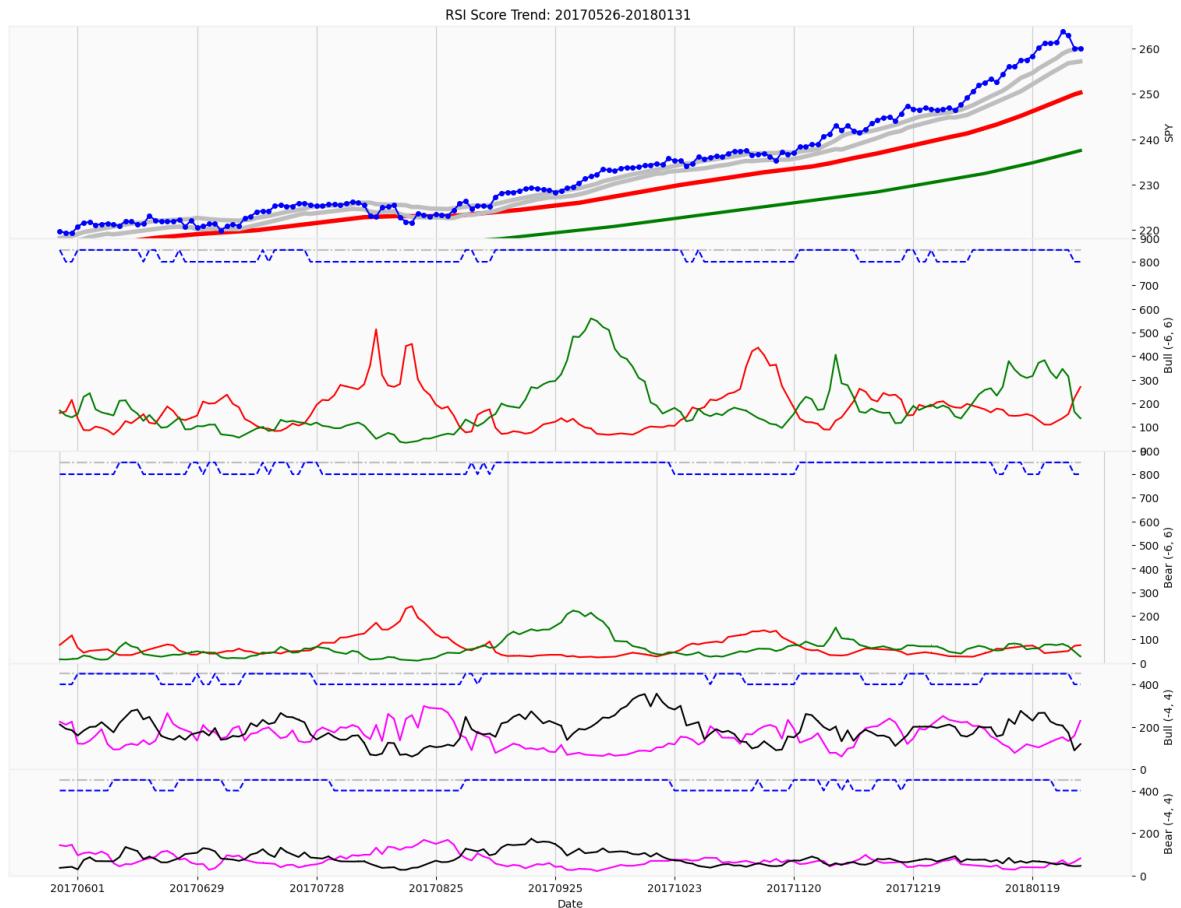


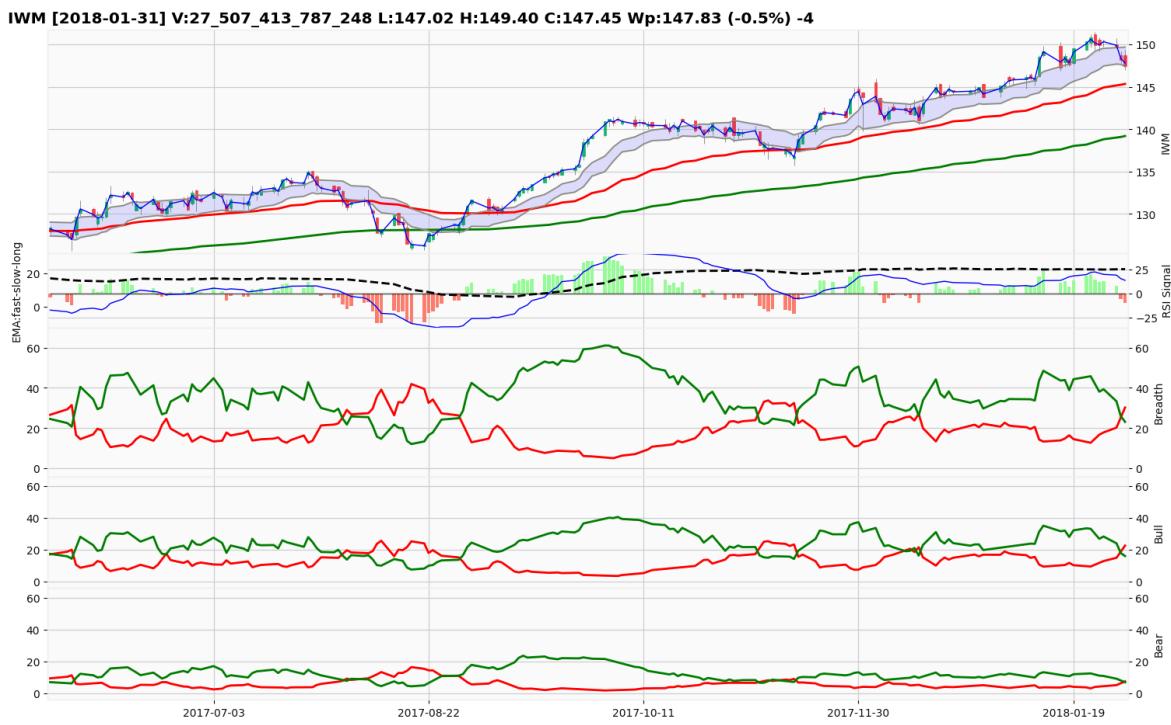
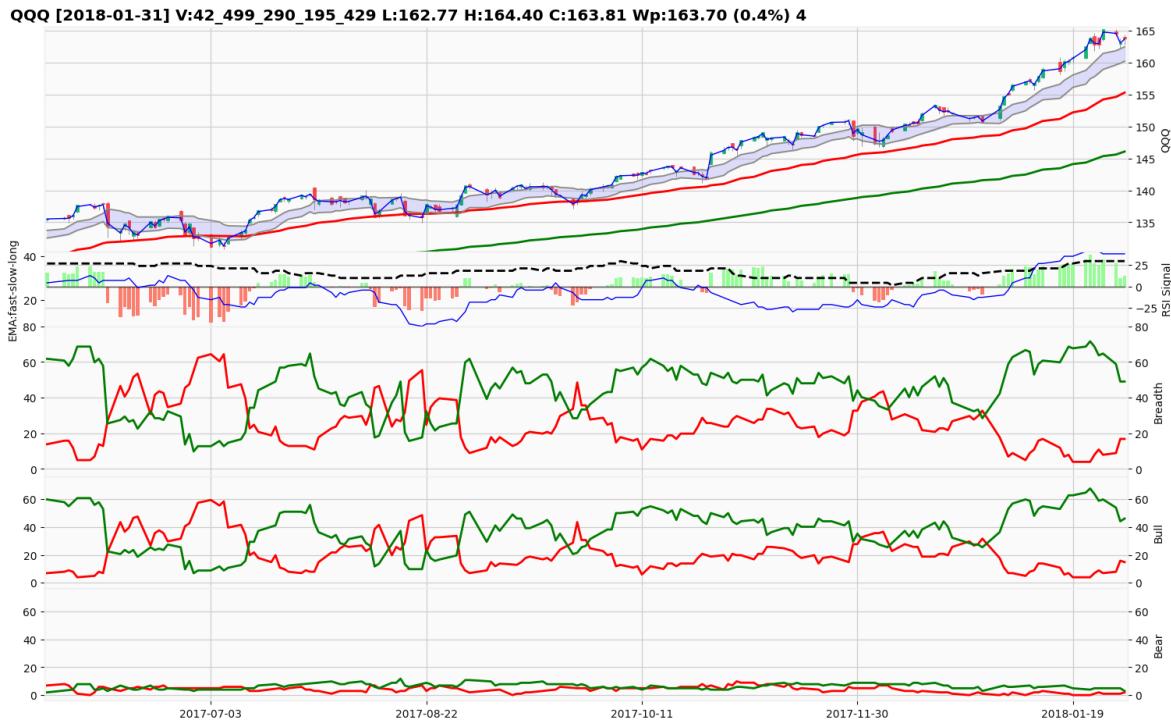


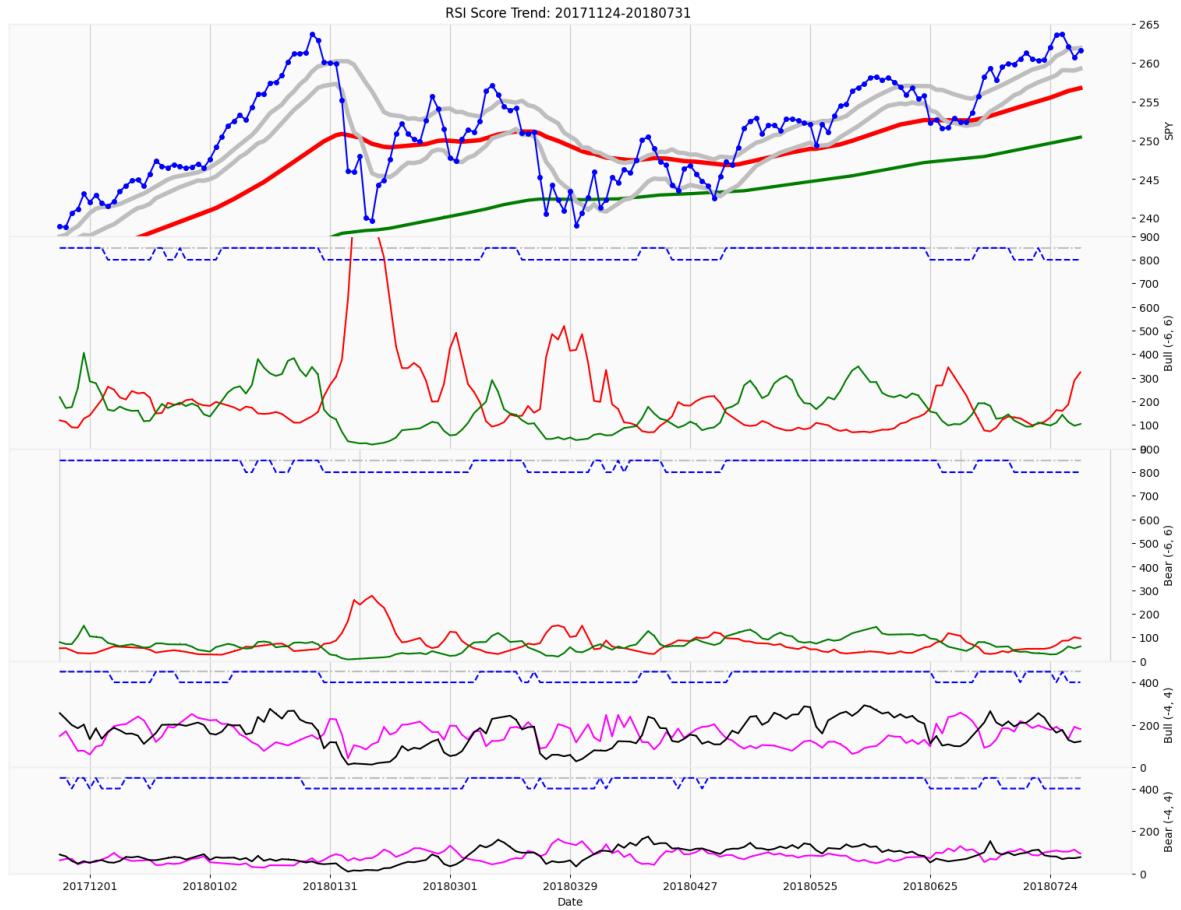
**SPY [2017-07-31] V:71\_698\_132\_347\_706 L:225.09 H:225.96 C:225.32 Wp:225.42 (-0.1%) 0**

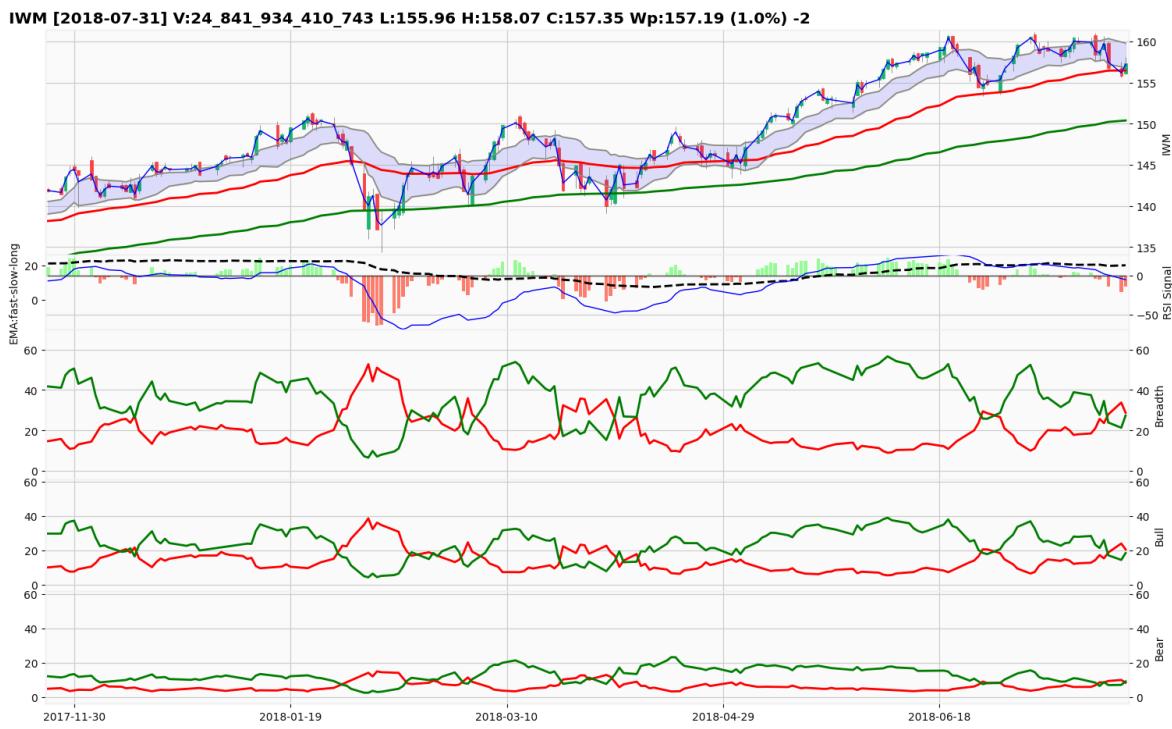
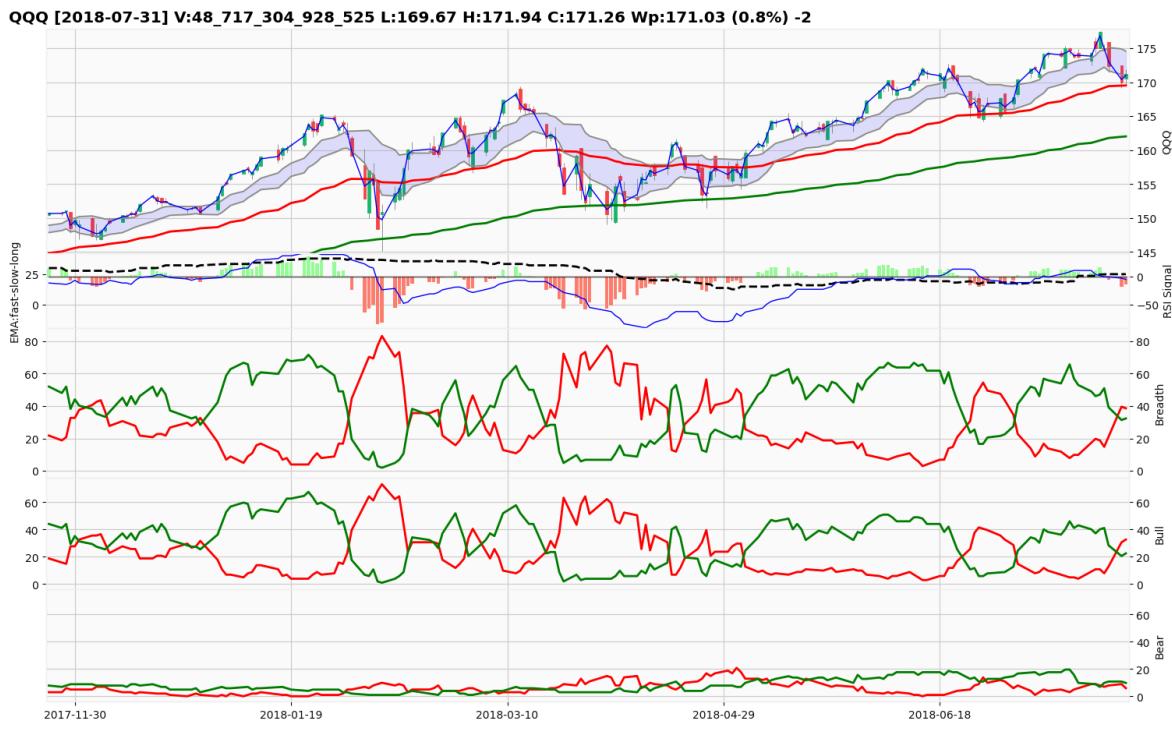


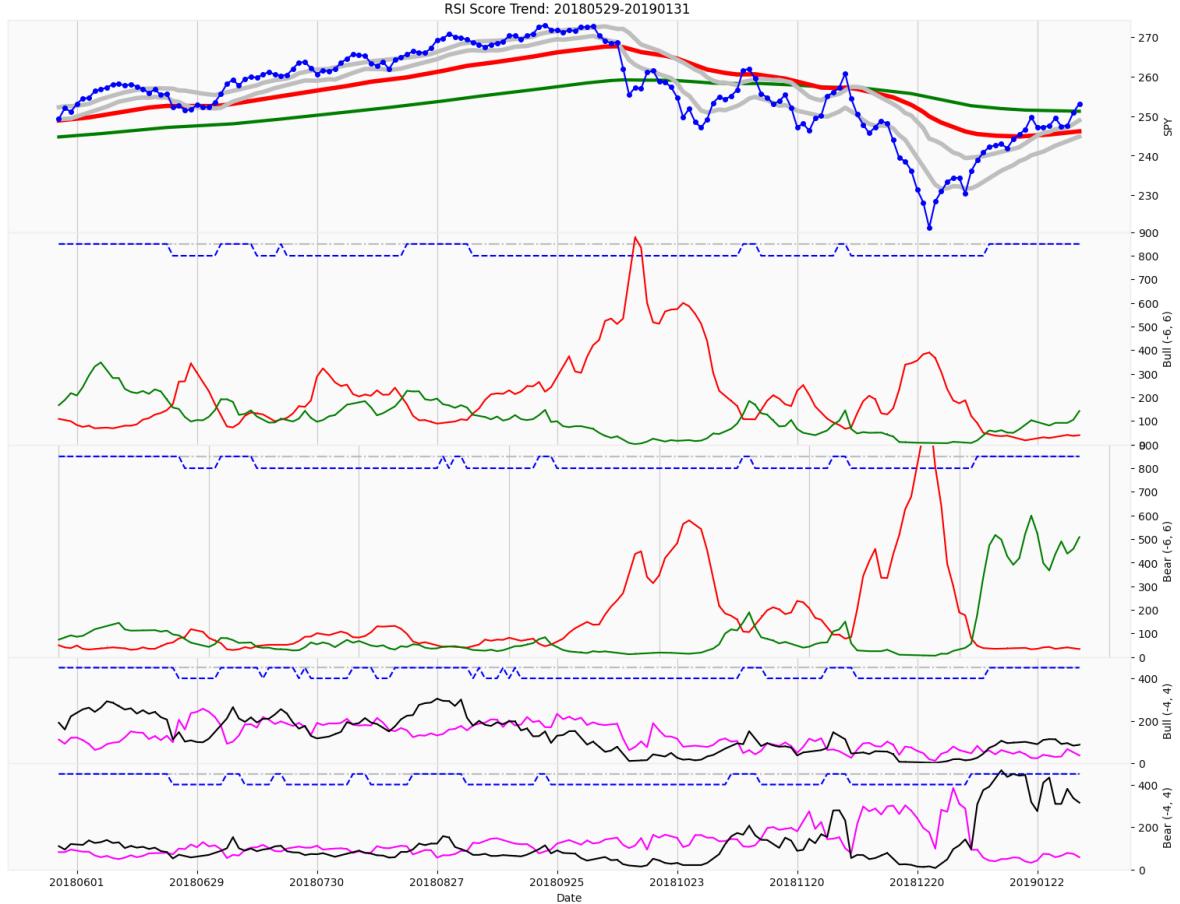


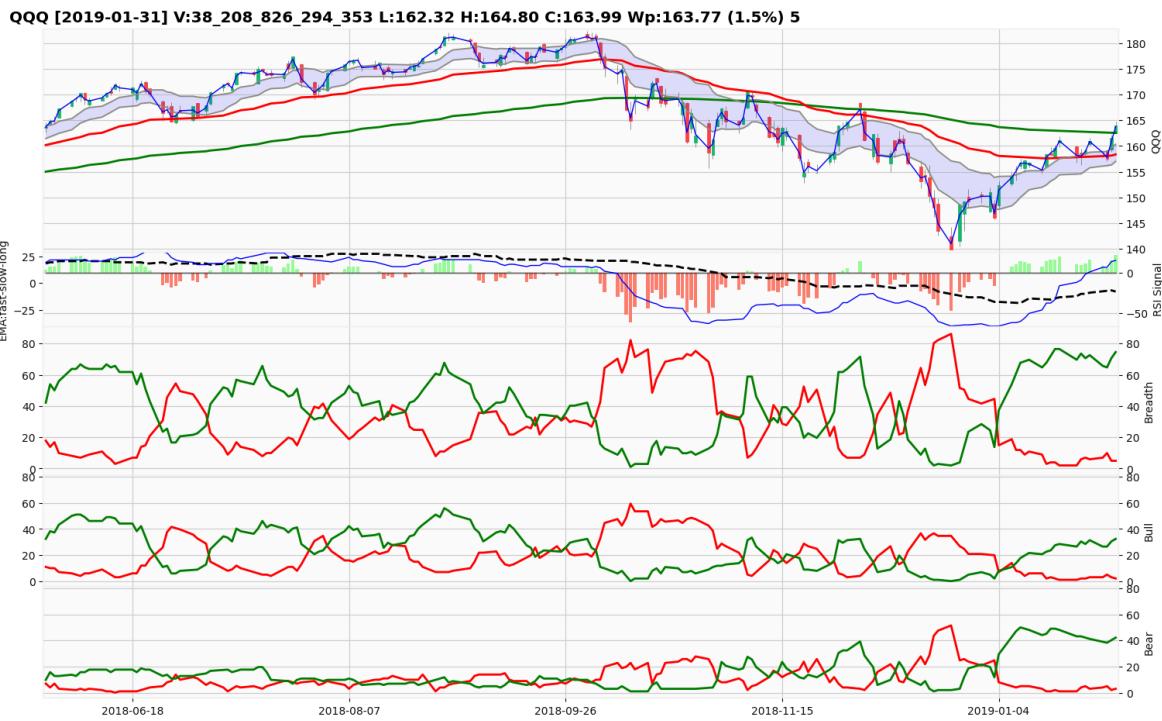


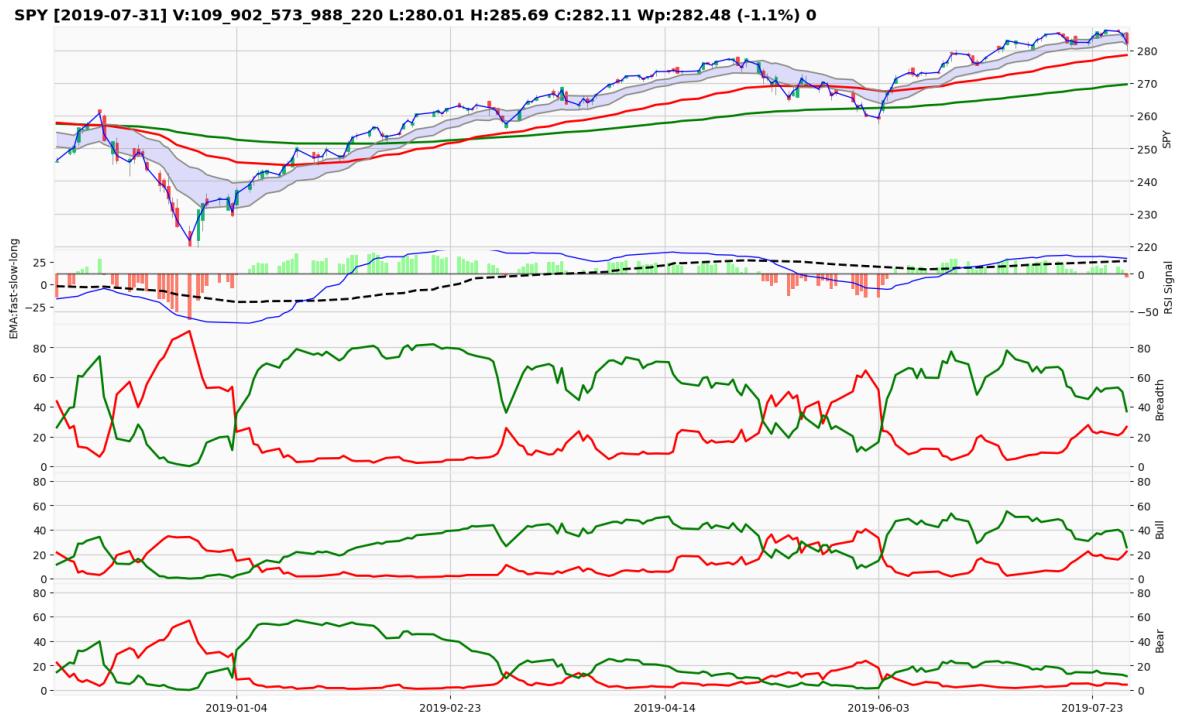
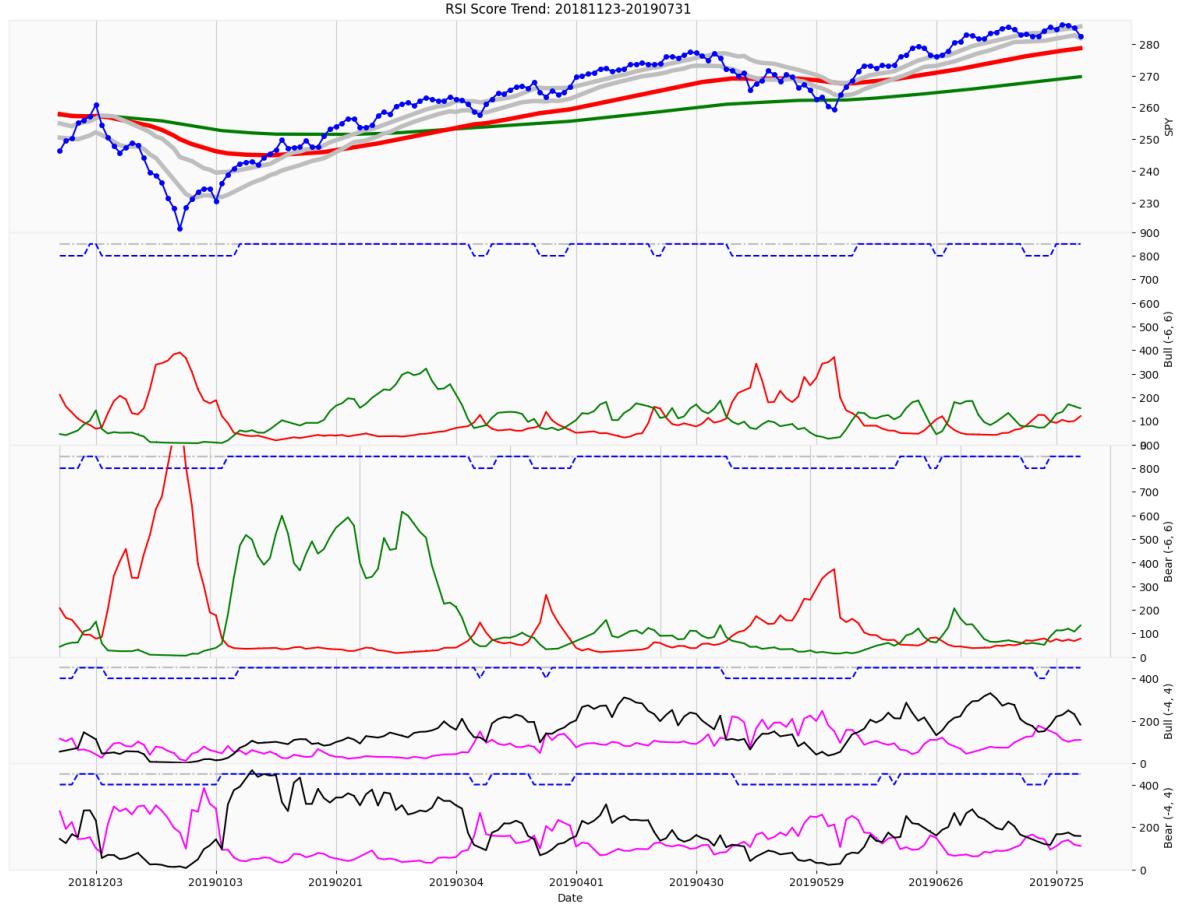


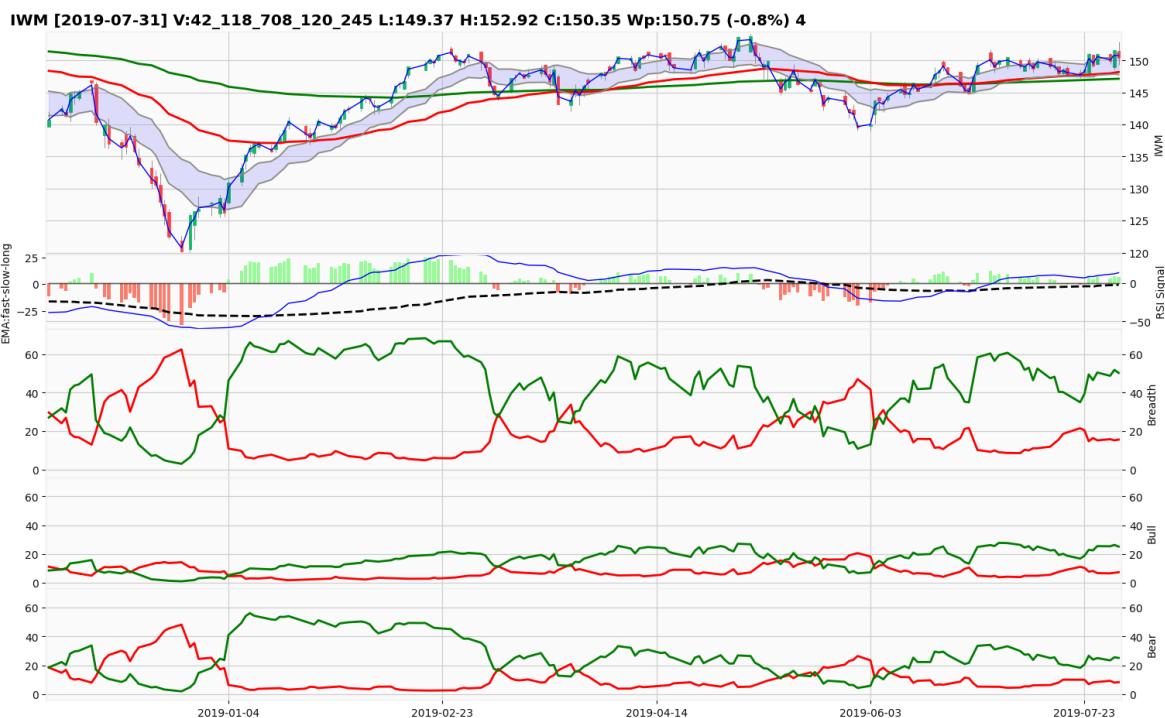
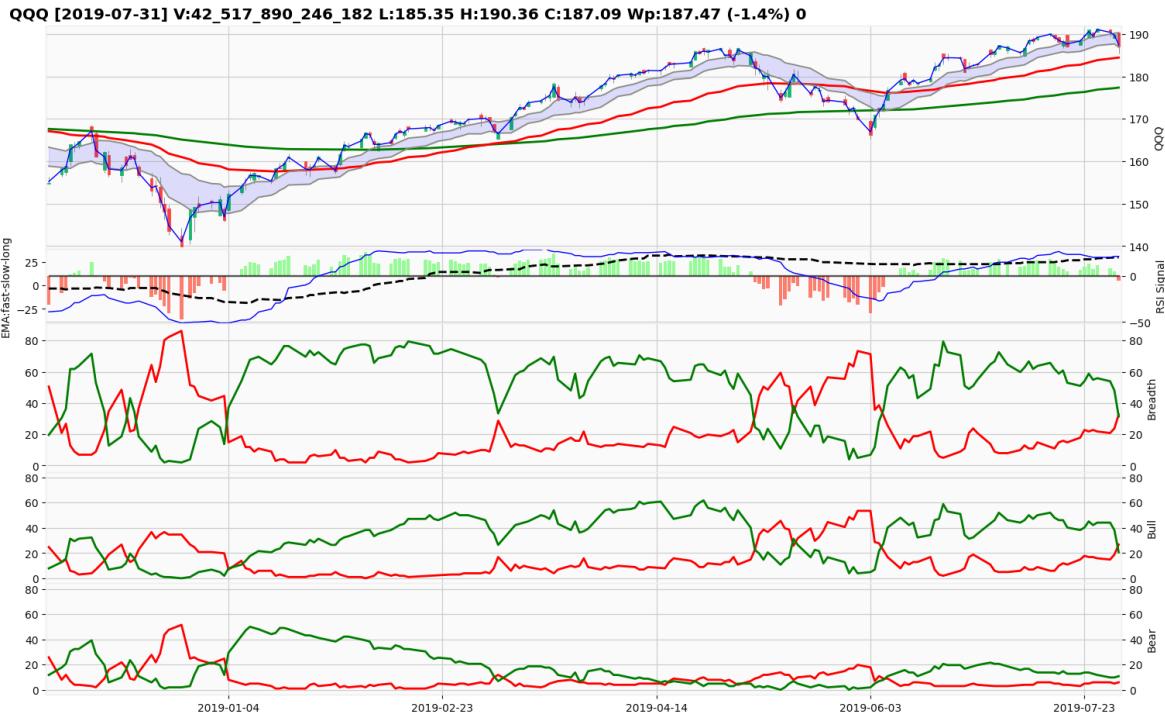


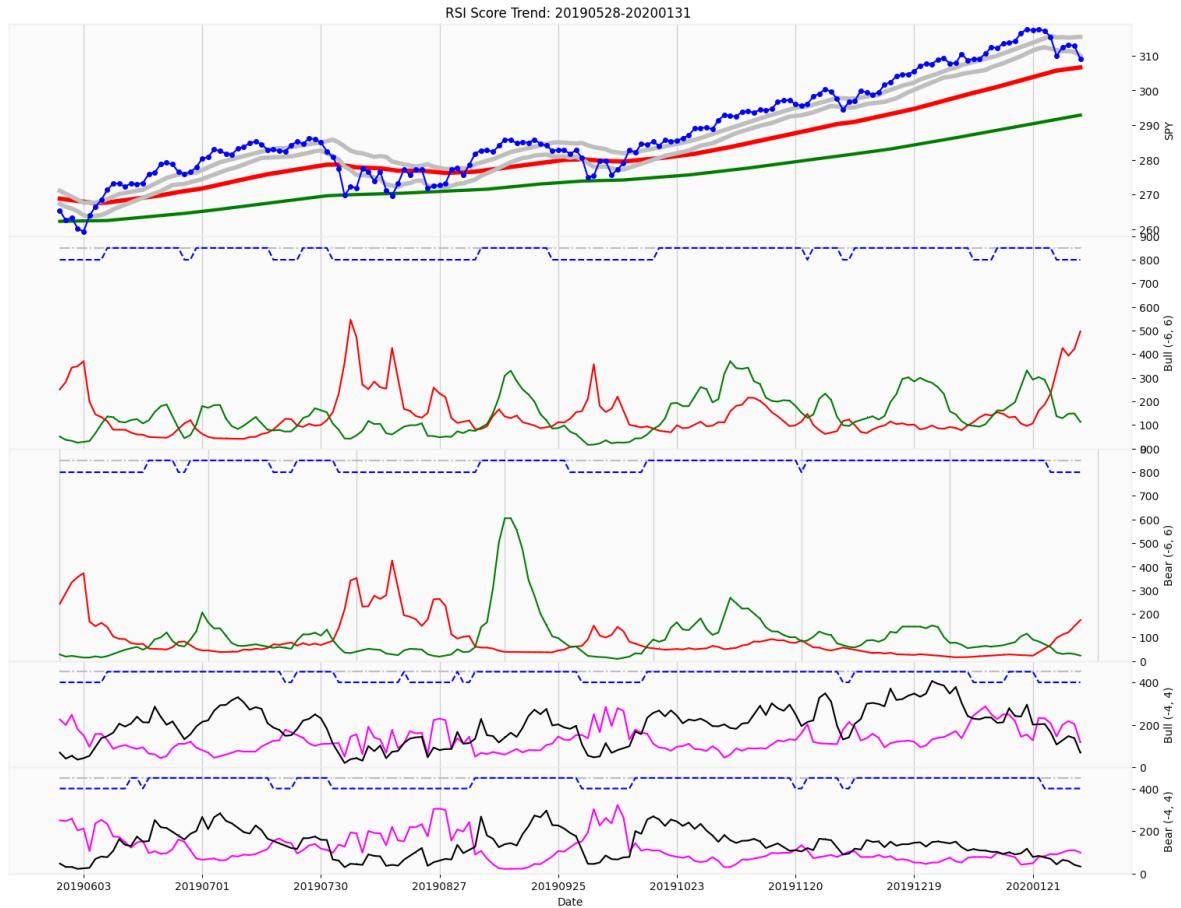




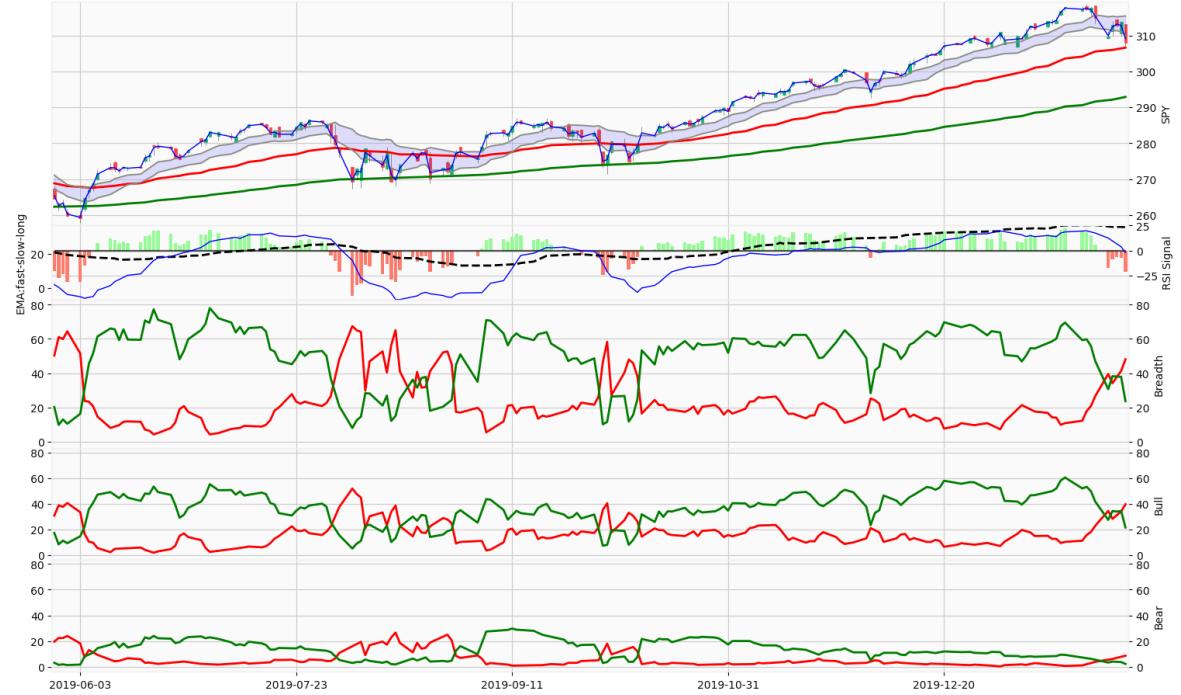


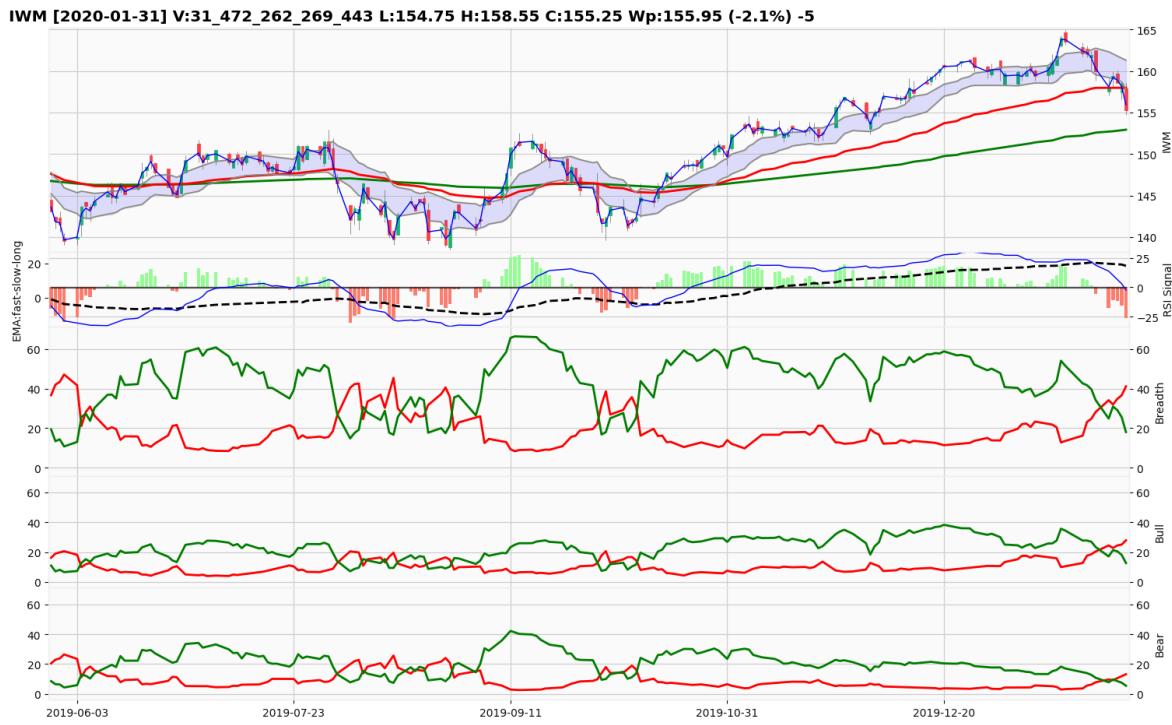
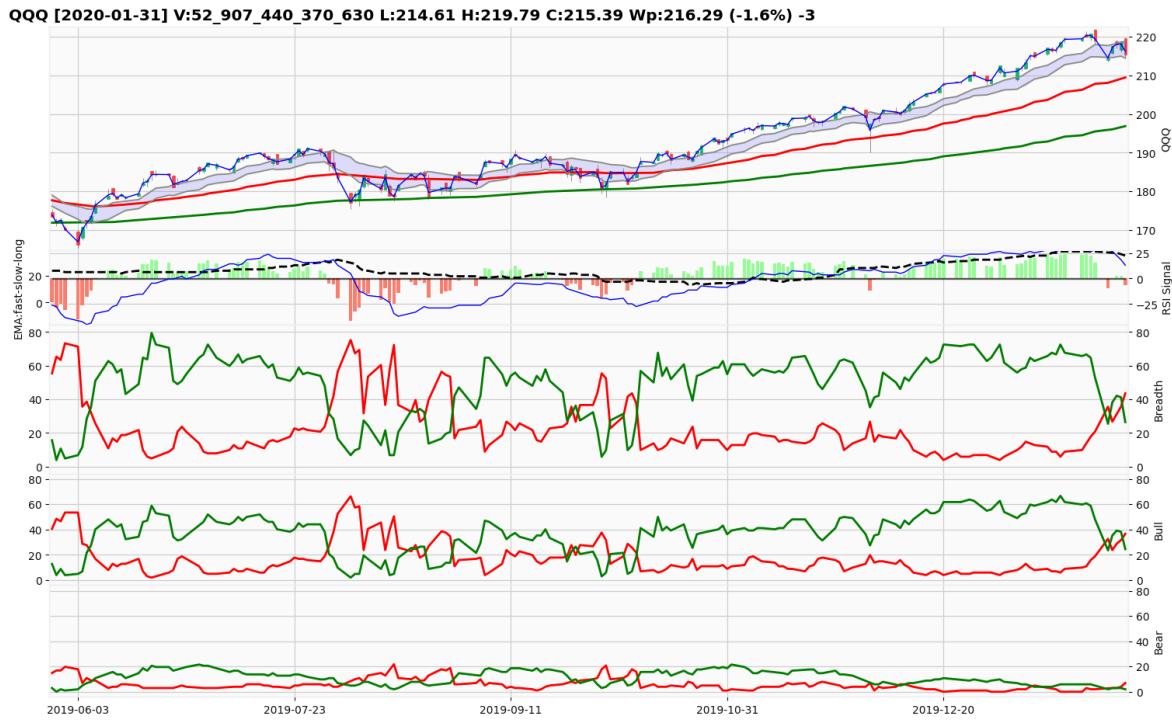


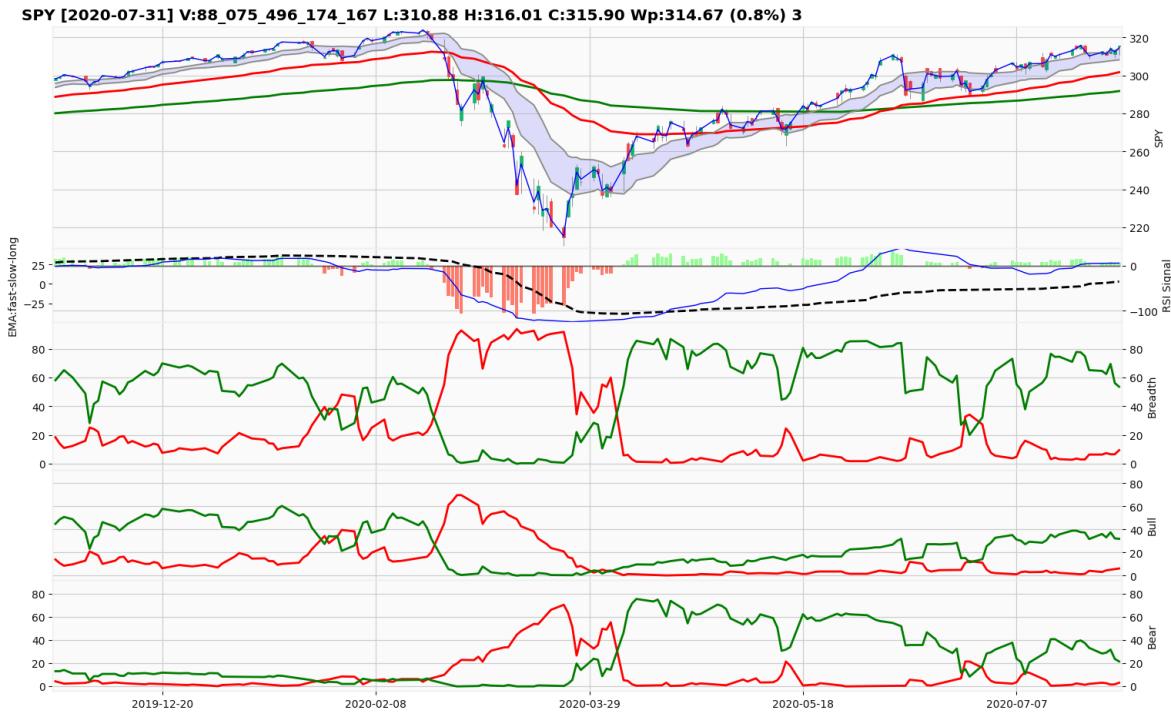
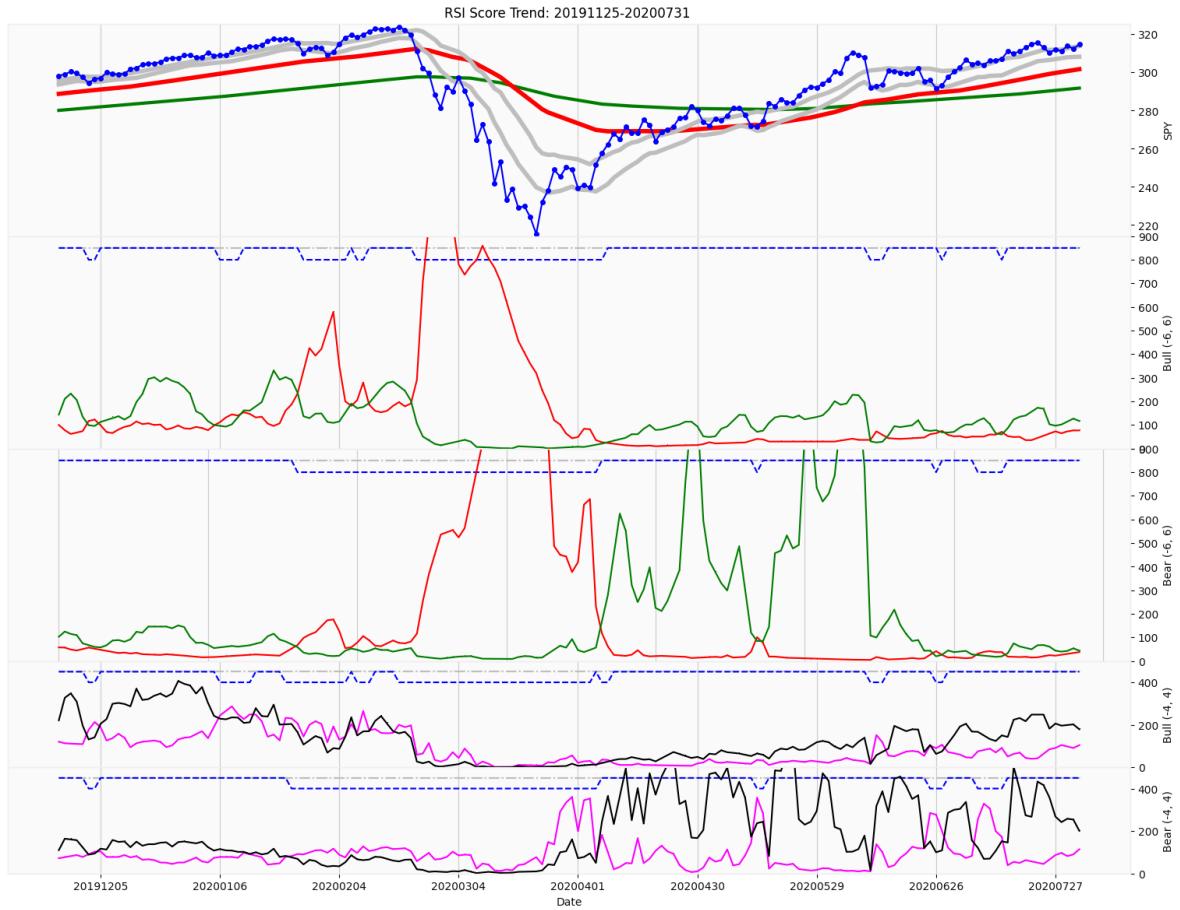


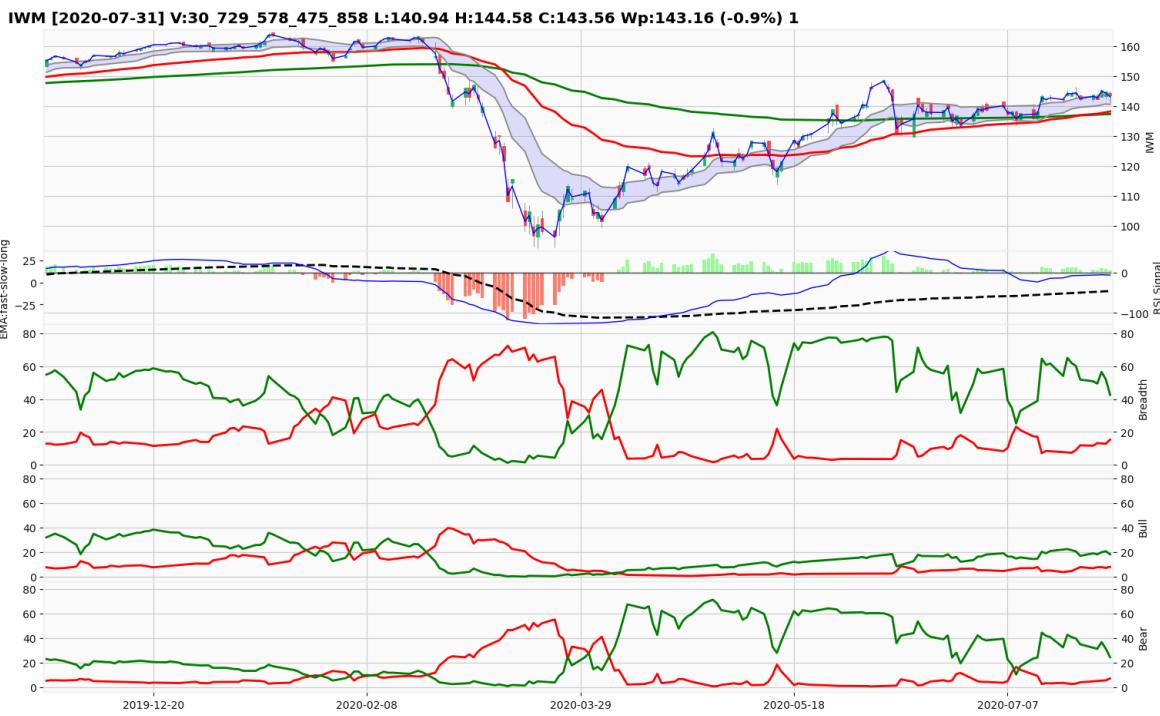
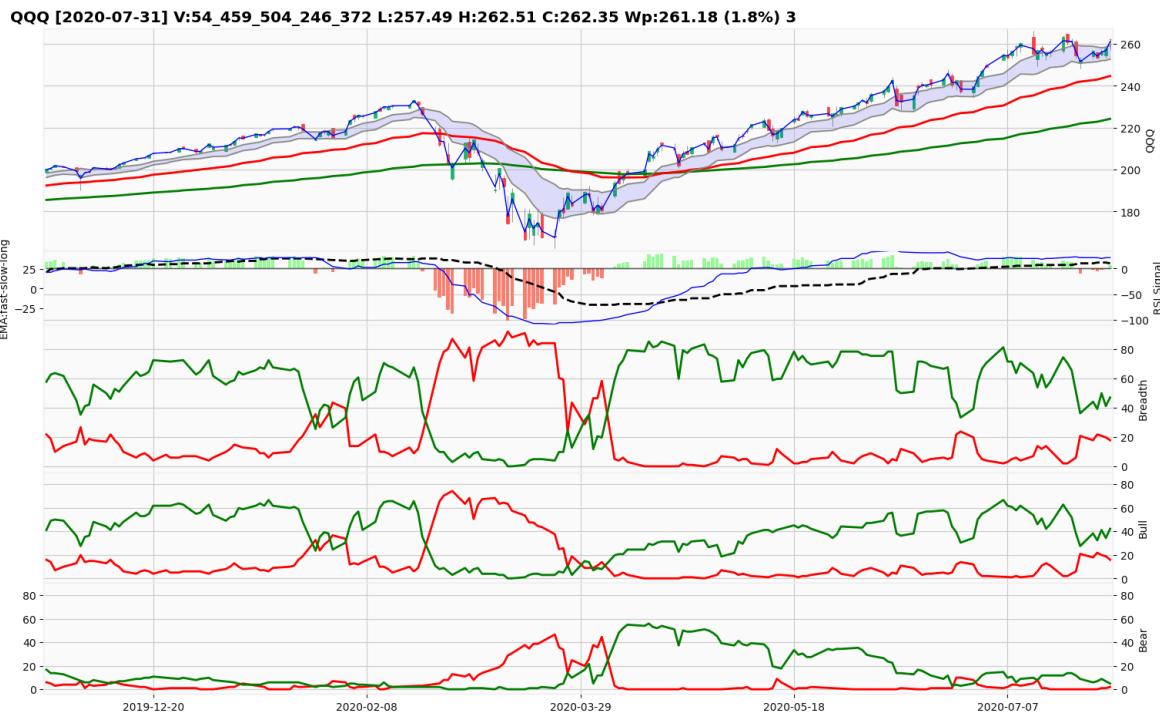


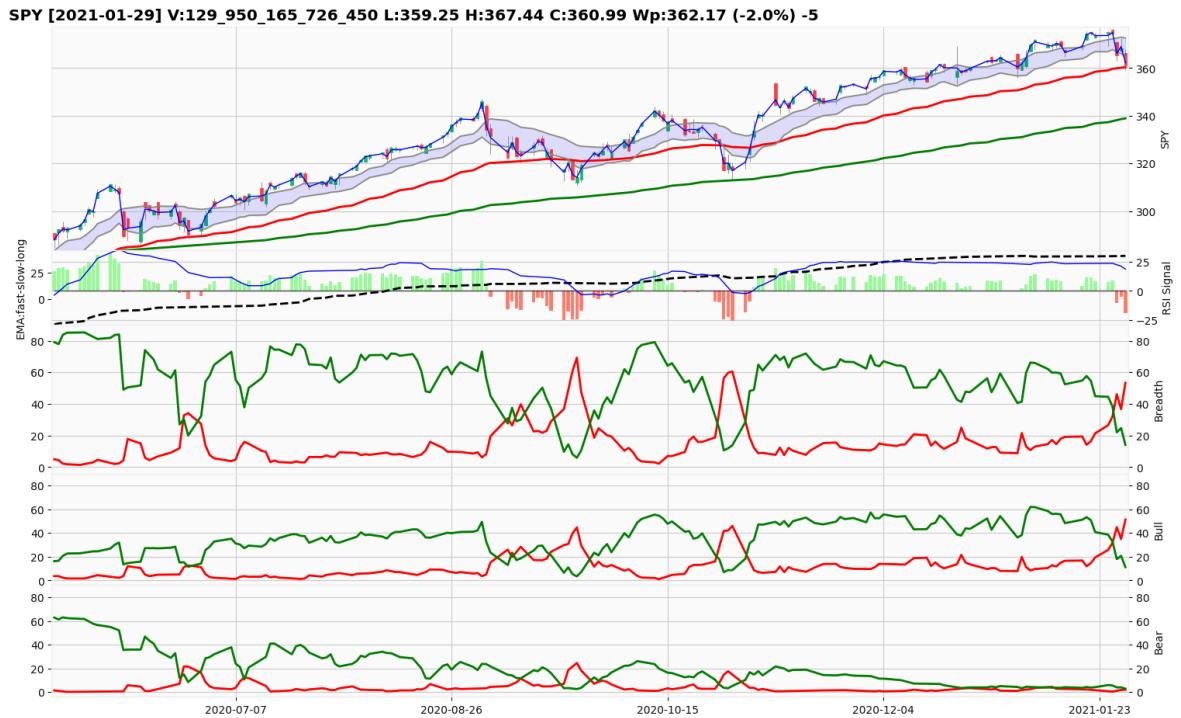
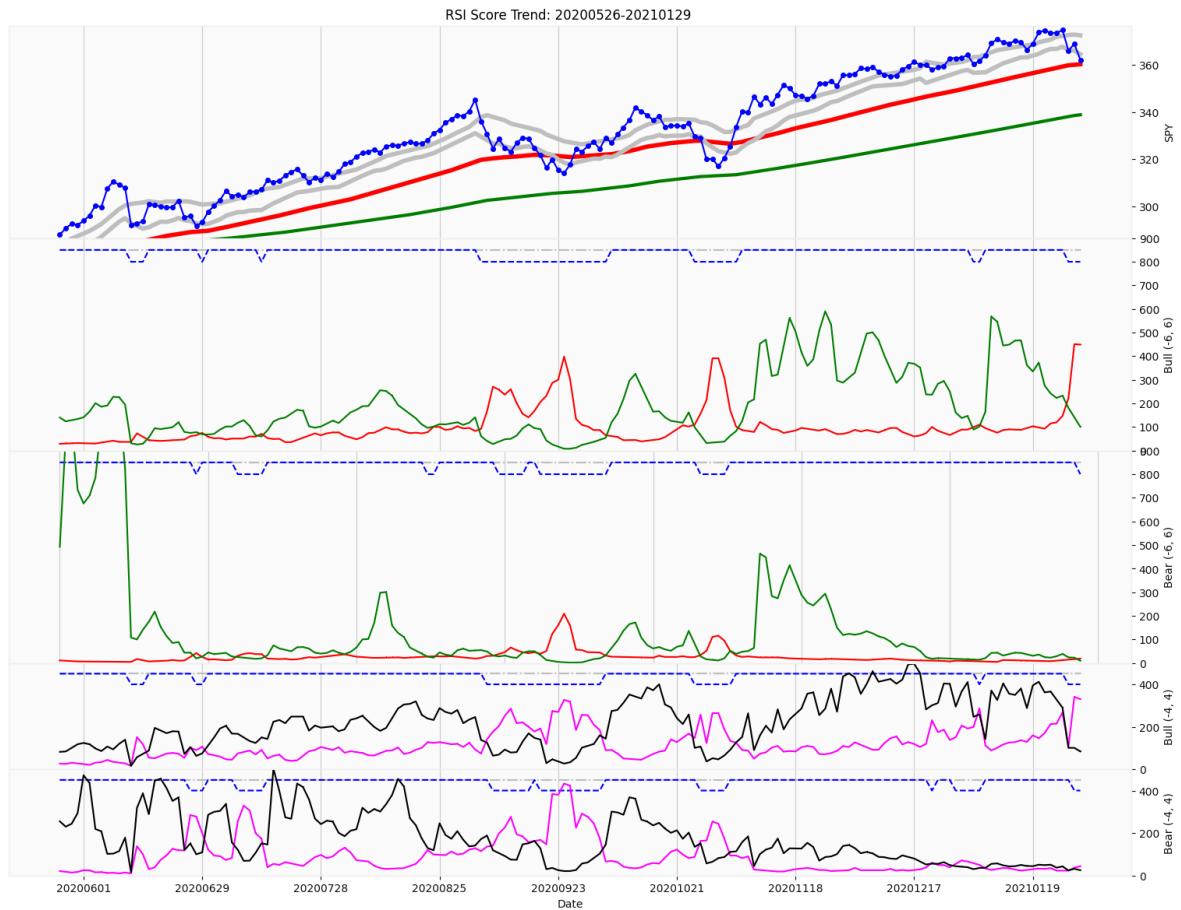
SPY [2020-01-31] V:118\_889\_350\_239\_580 L:307.12 H:313.29 C:308.07 Wp:309.14 (-1.8%) -5

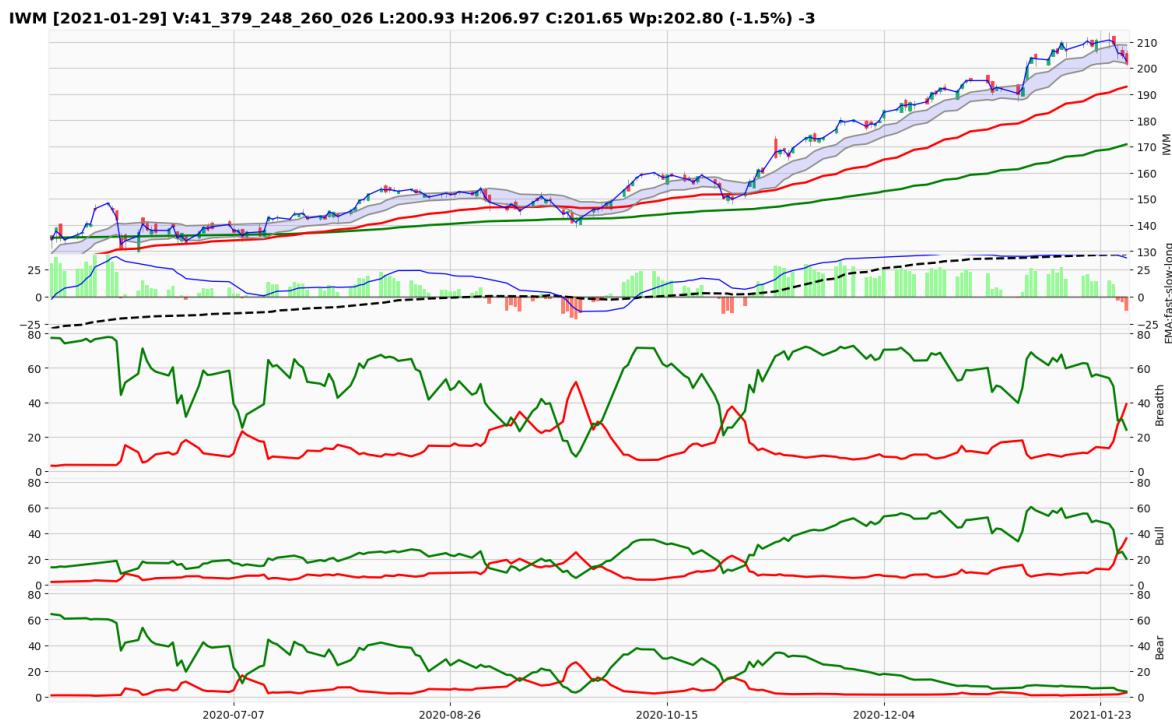
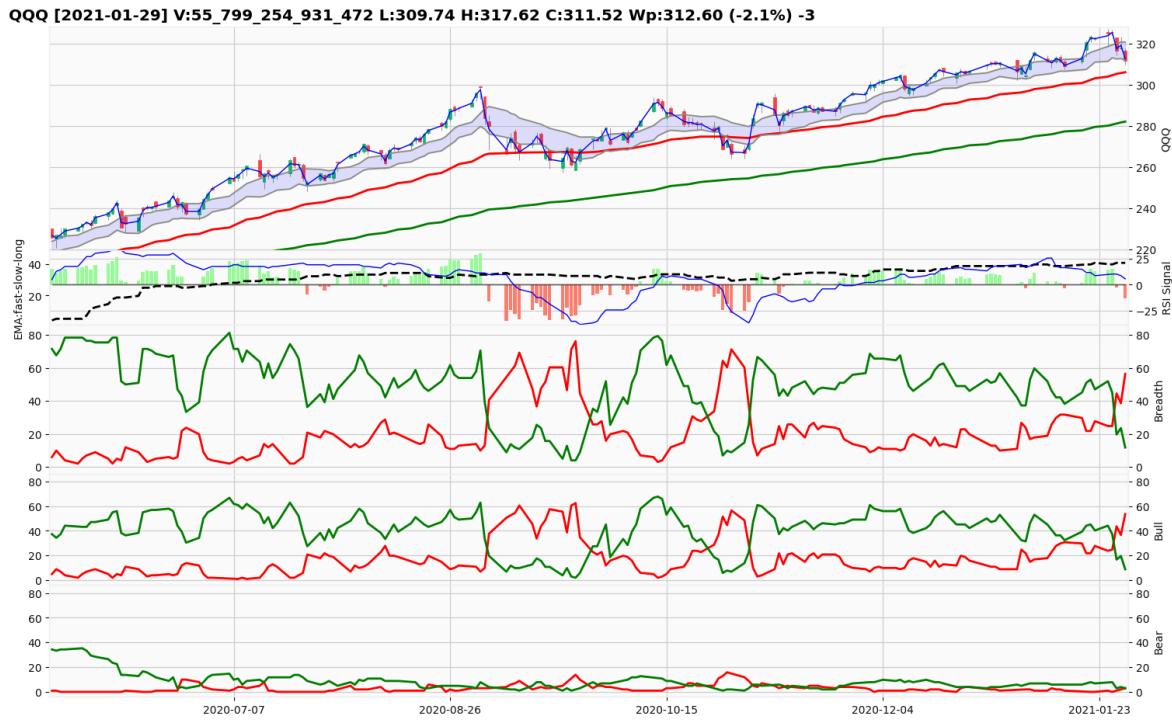


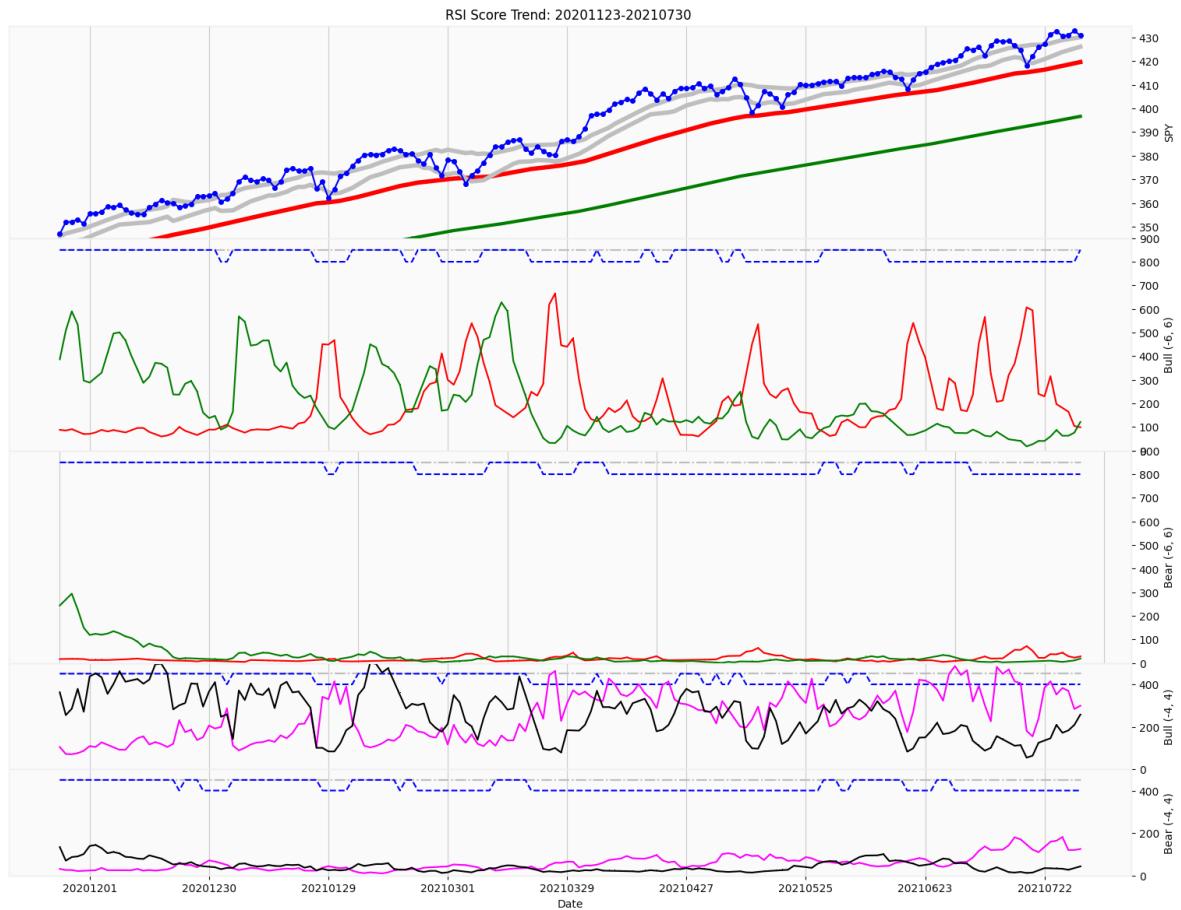




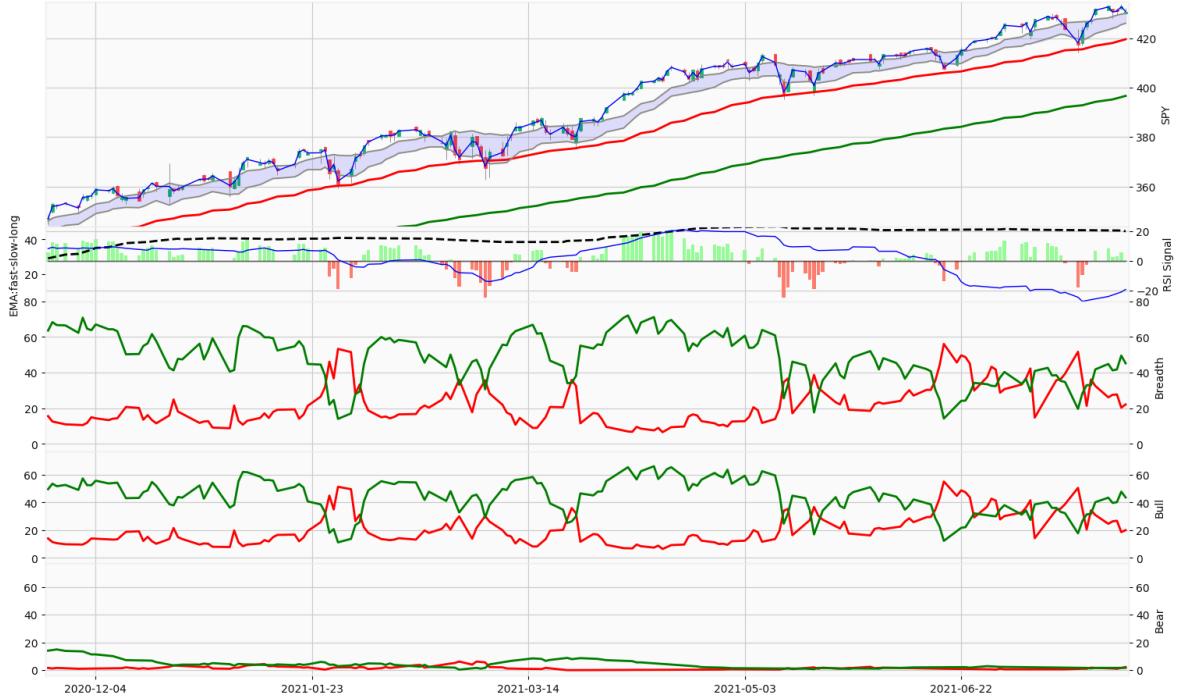




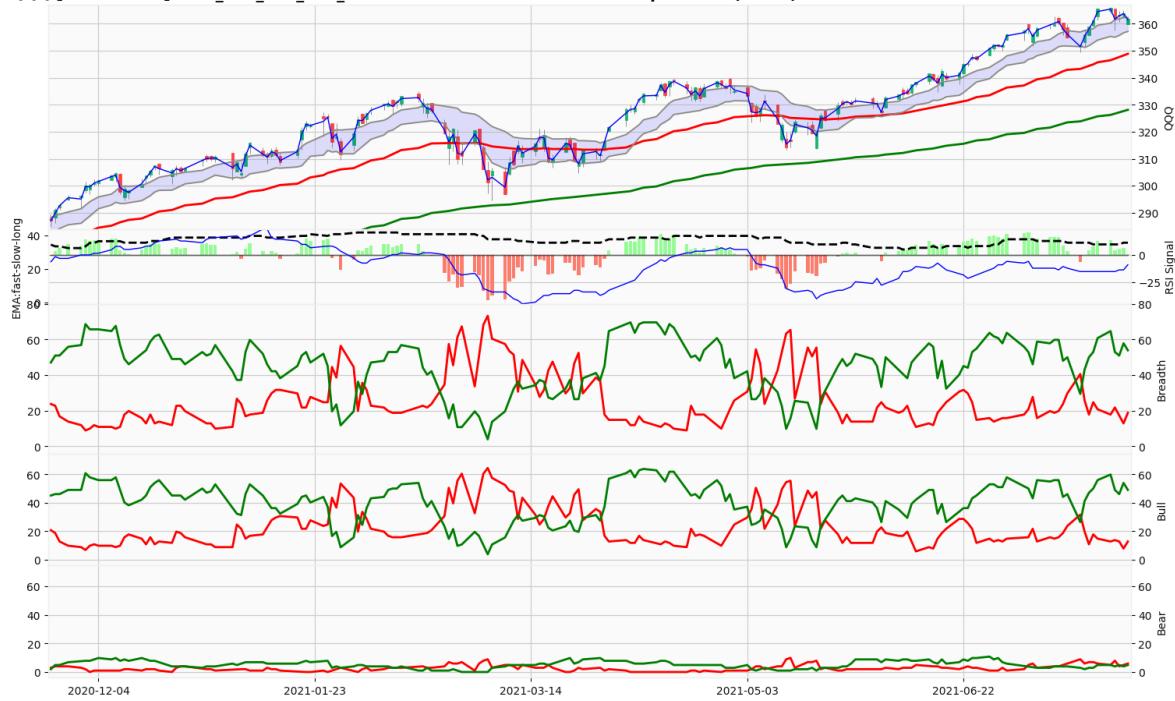




**SPY [2021-07-30] V:70\_222\_124\_337\_583 L:429.85 H:432.09 C:430.58 Wp:430.77 (-0.5%) 1**

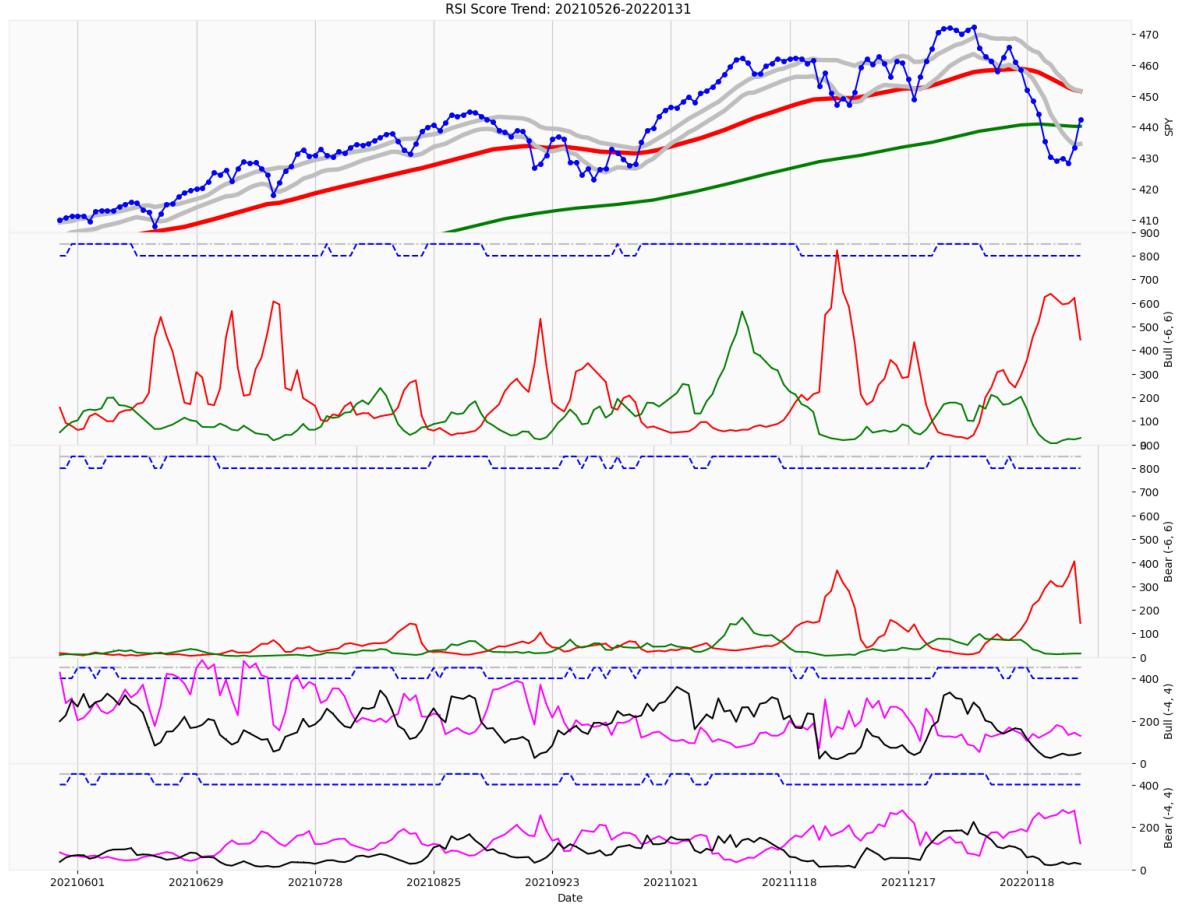


**QQQ [2021-07-30] V:36\_752\_498\_830\_083 L:359.77 H:362.51 C:361.92 Wp:361.53 (-0.5%) 1**

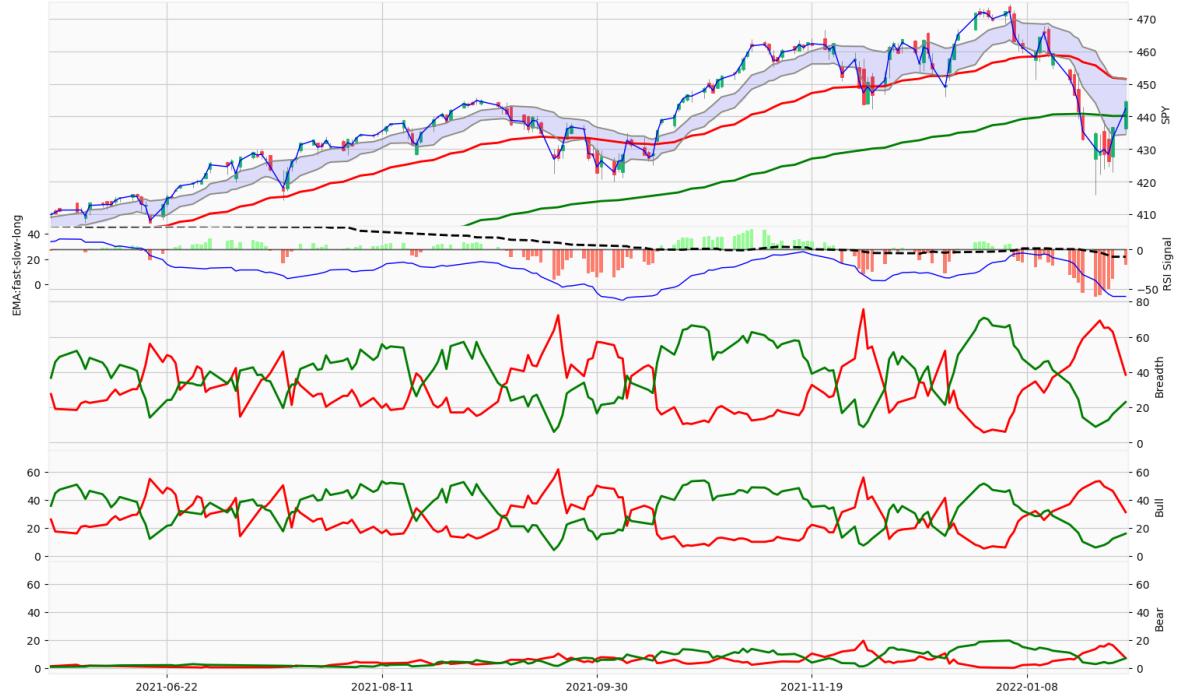


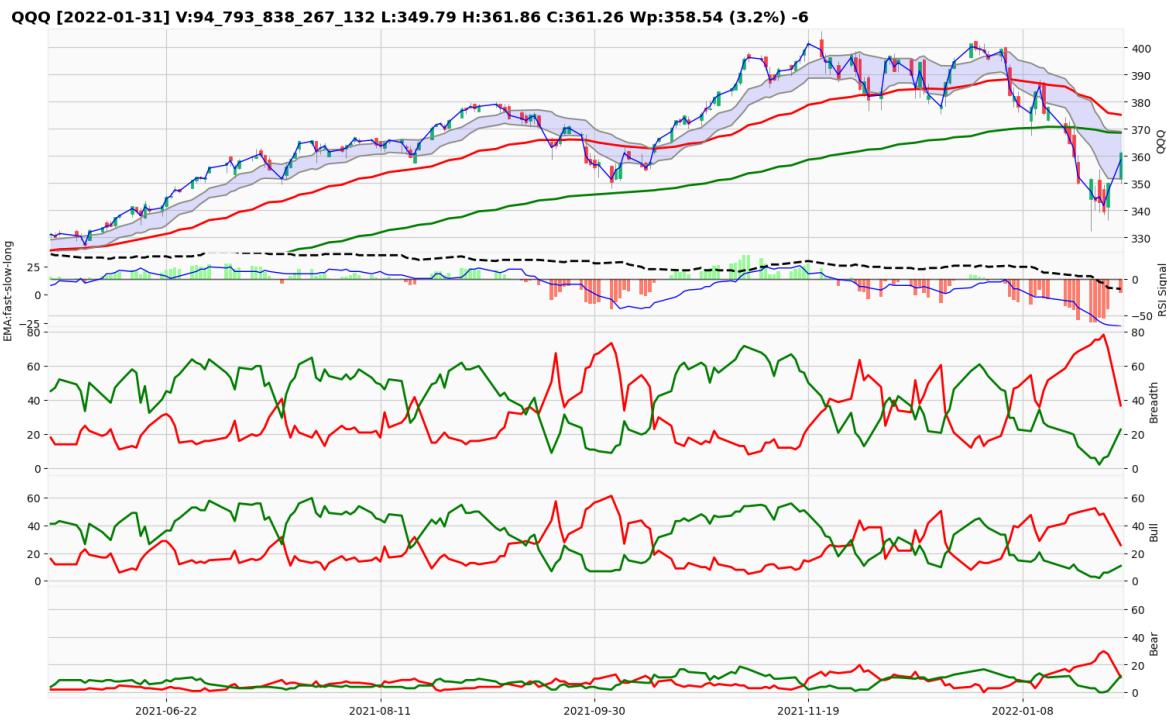
**IWM [2021-07-30] V:28\_926\_706\_361\_200 L:216.82 H:220.53 C:217.58 Wp:218.13 (-0.7%) -3**

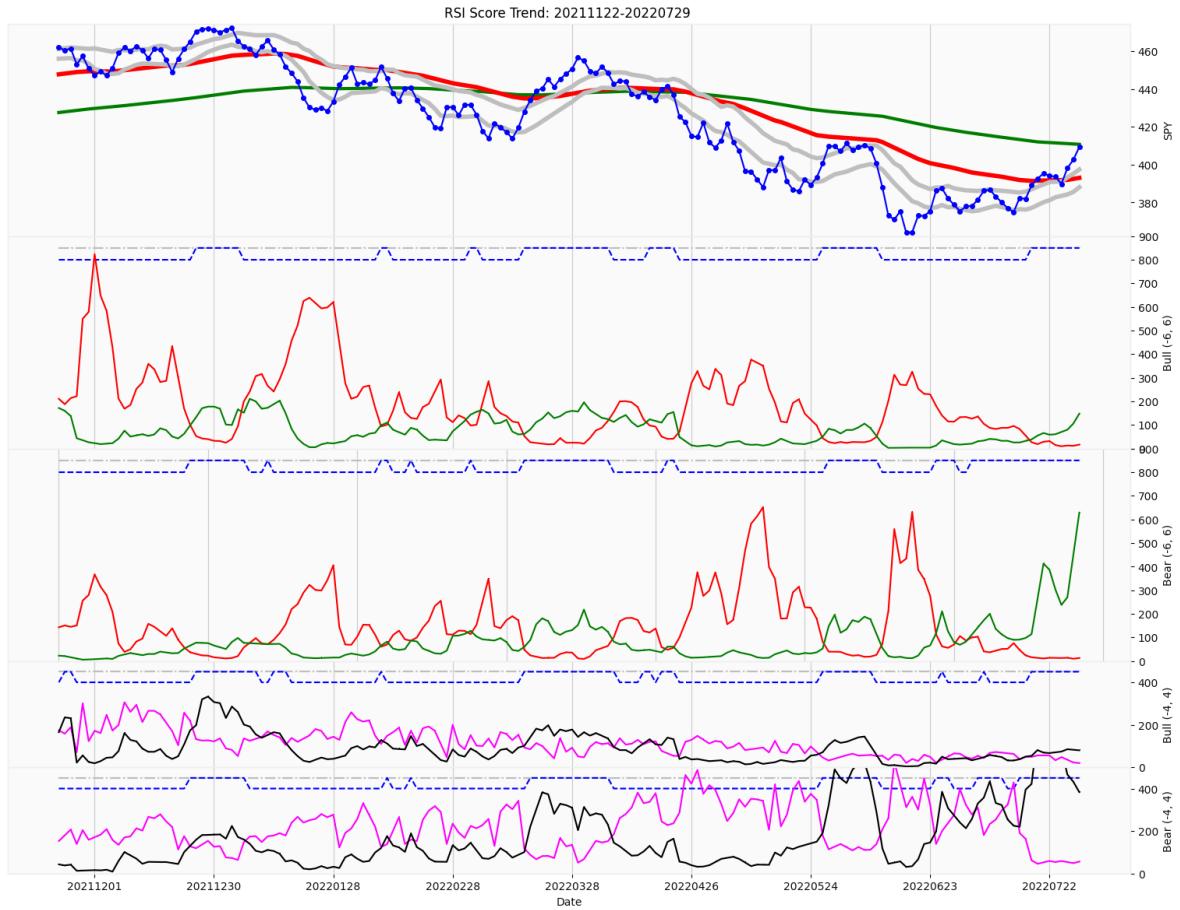




**SPY [2022-01-31] V:154\_018\_467\_946\_970 L:434.76 H:445.11 C:444.75 Wp:442.34 (1.8%) -6**



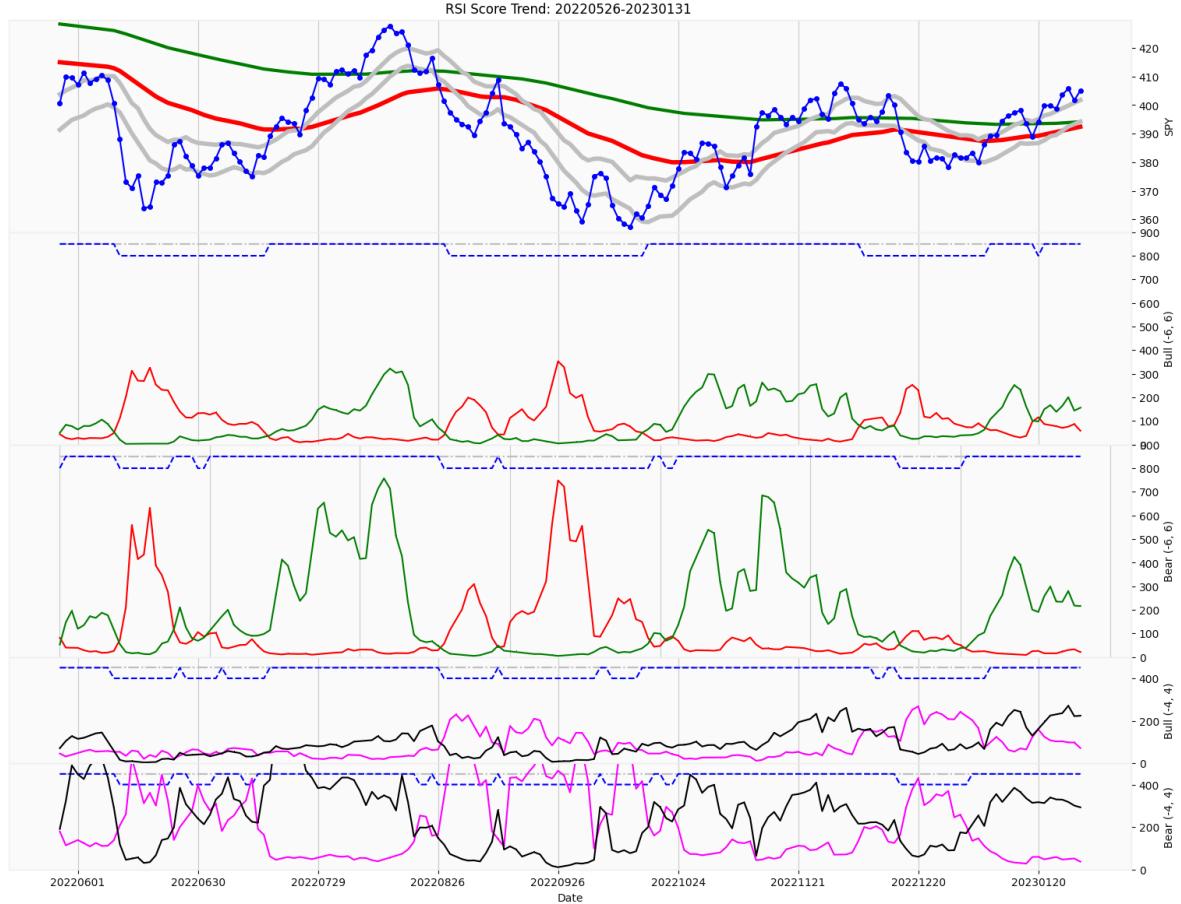




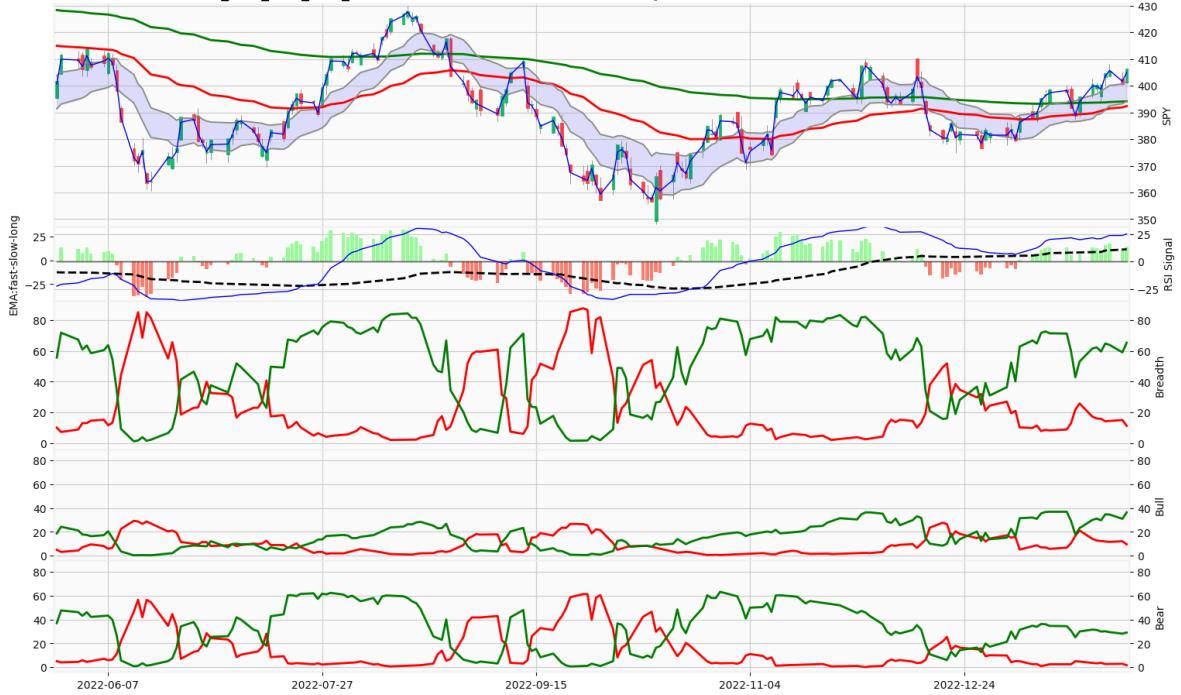
**SPY [2022-07-29] V:87\_361\_969\_834\_701 L:405.10 H:411.34 C:410.30 Wp:409.26 (1.5%) 6**

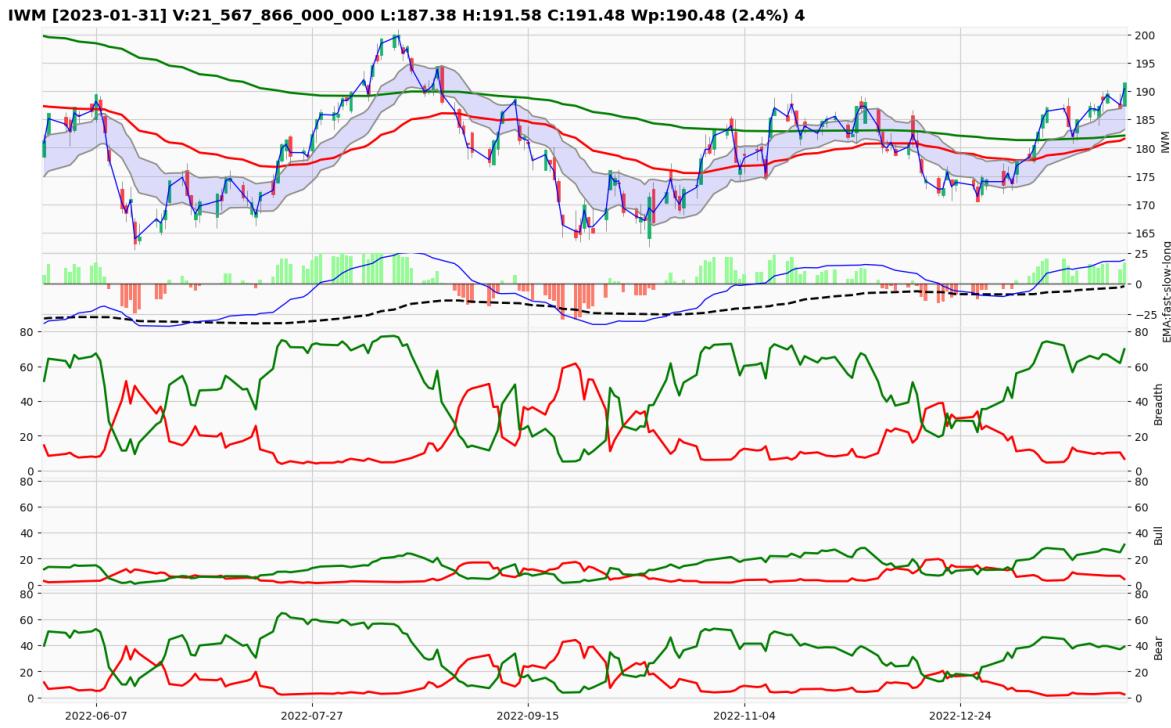
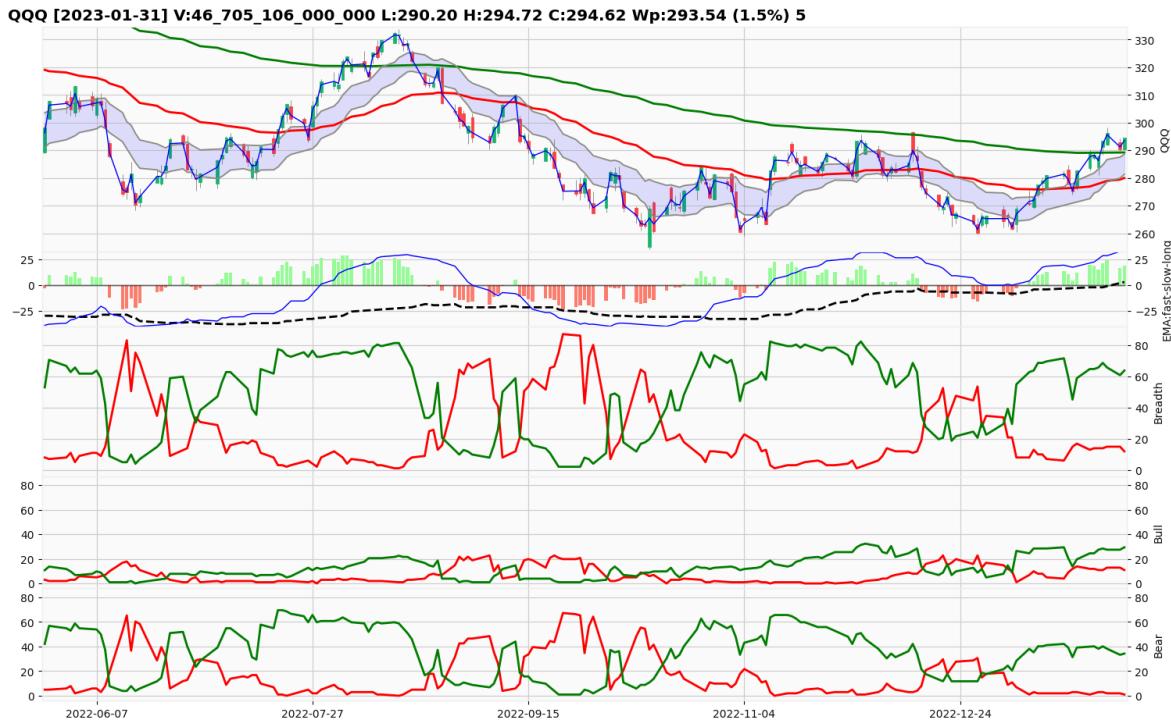


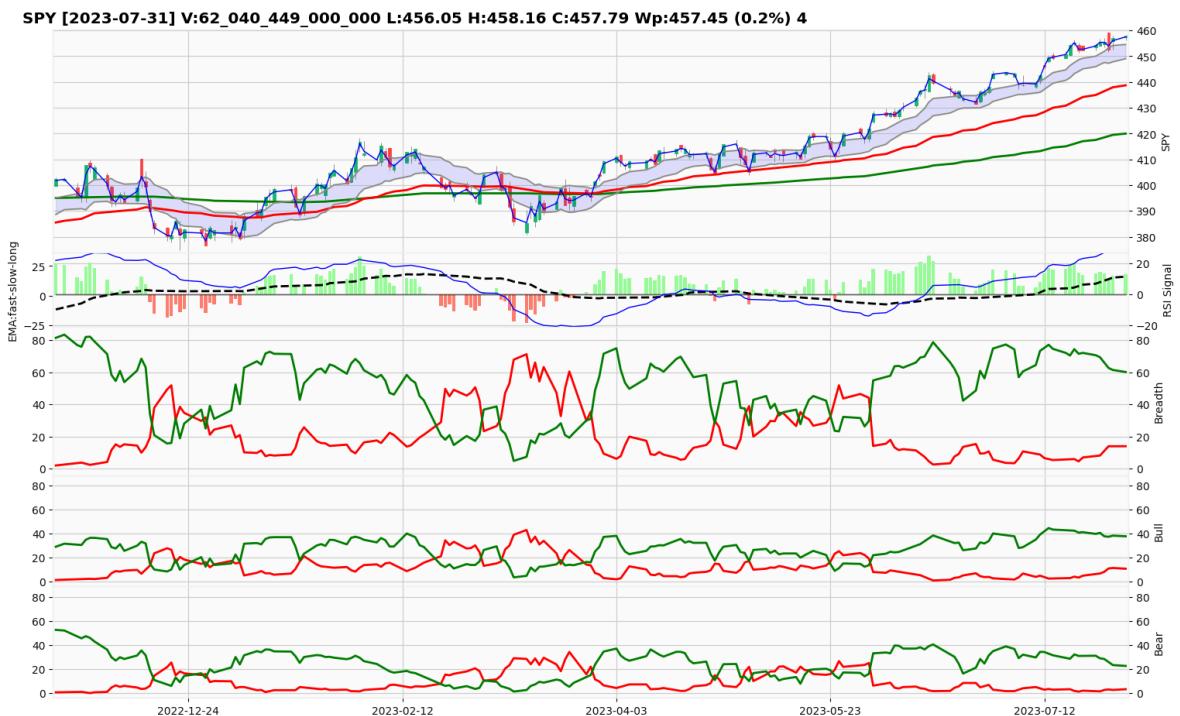
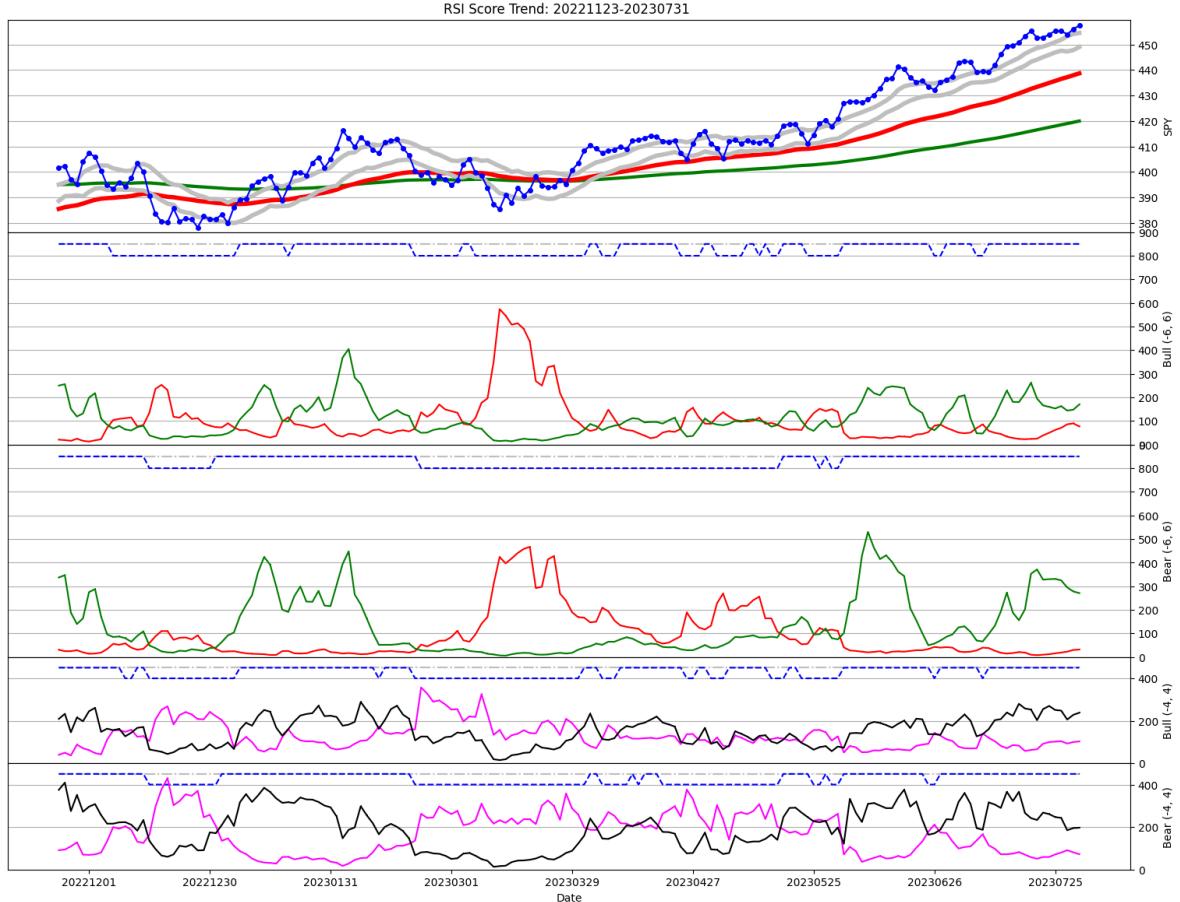


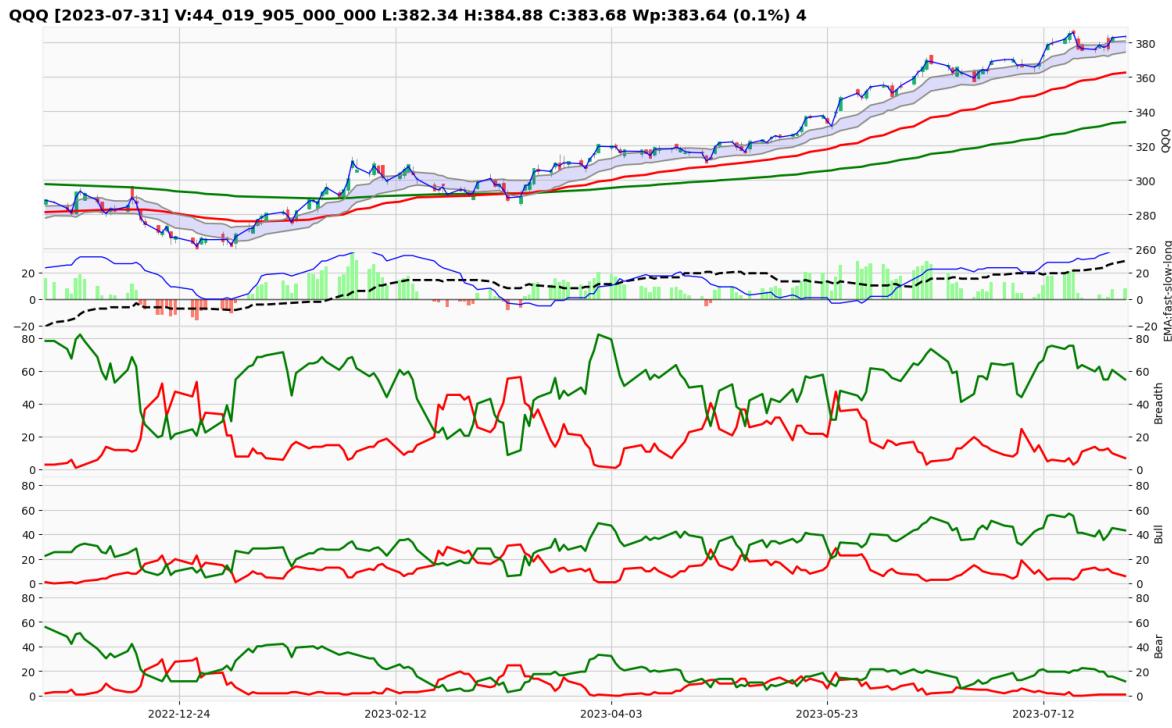


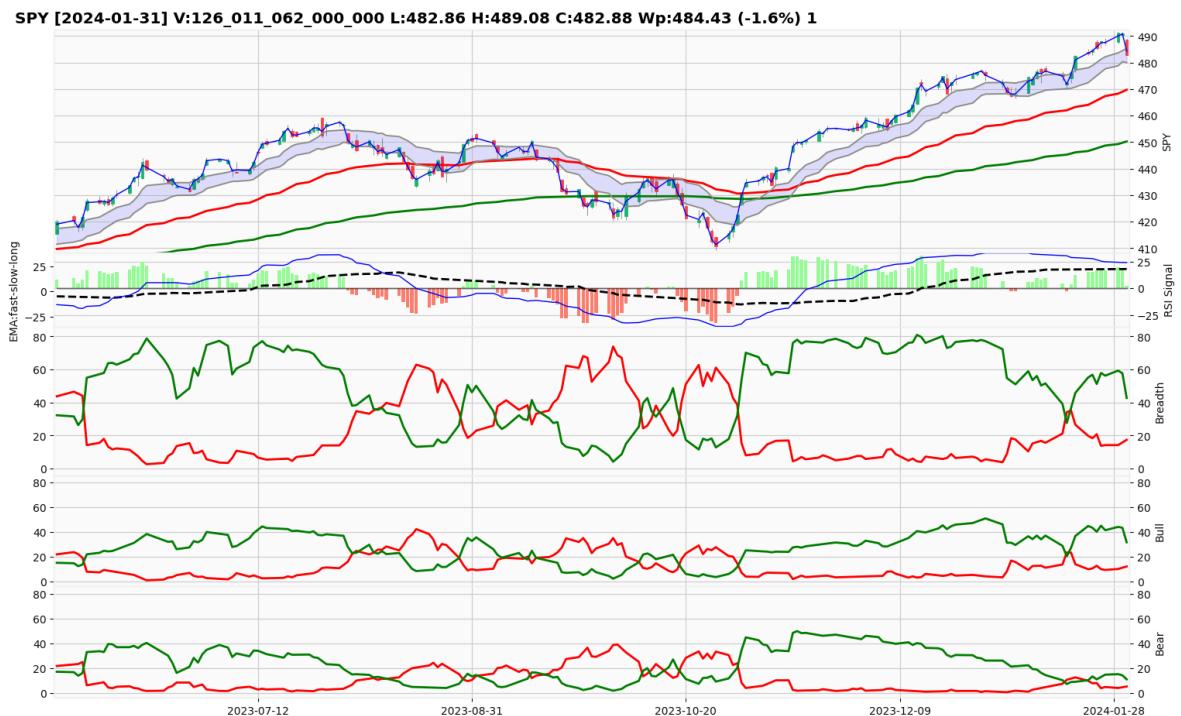
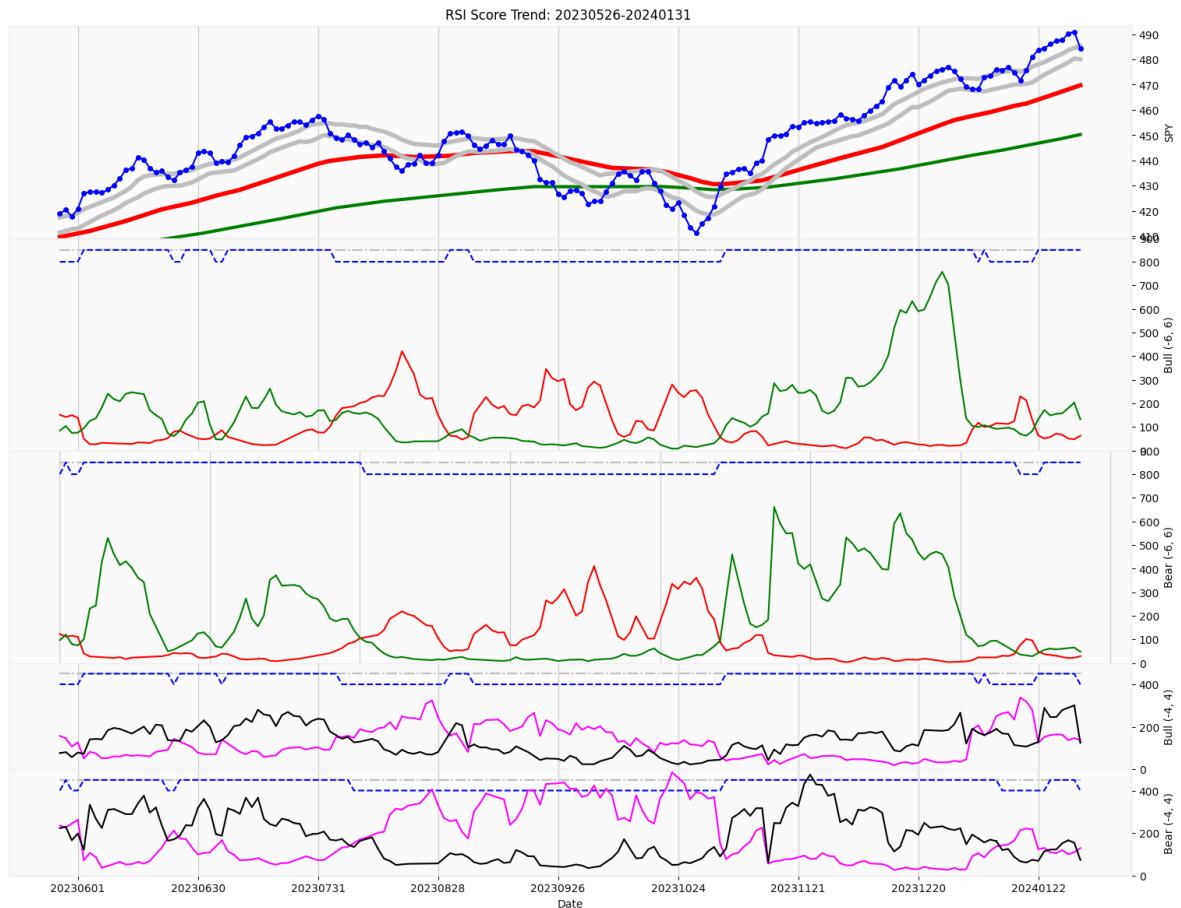
SPY [2023-01-31] V:86\_811\_786\_000\_000 L:400.77 H:406.53 C:406.48 Wp:405.06 (1.5%) 4

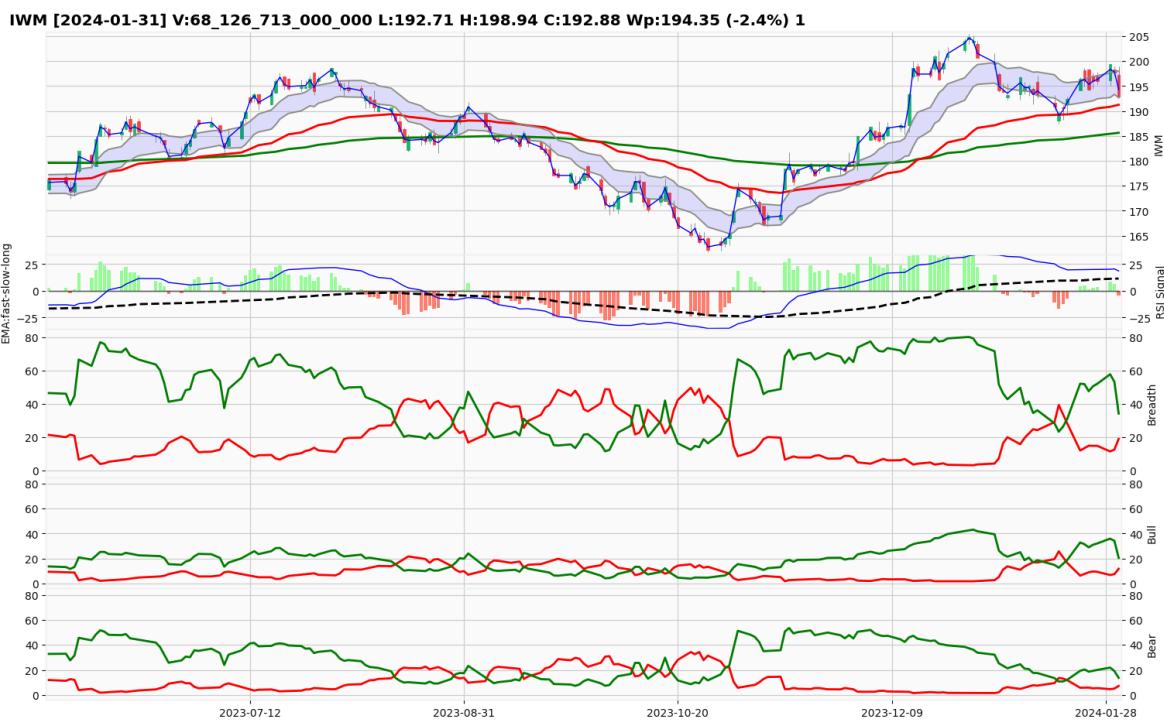
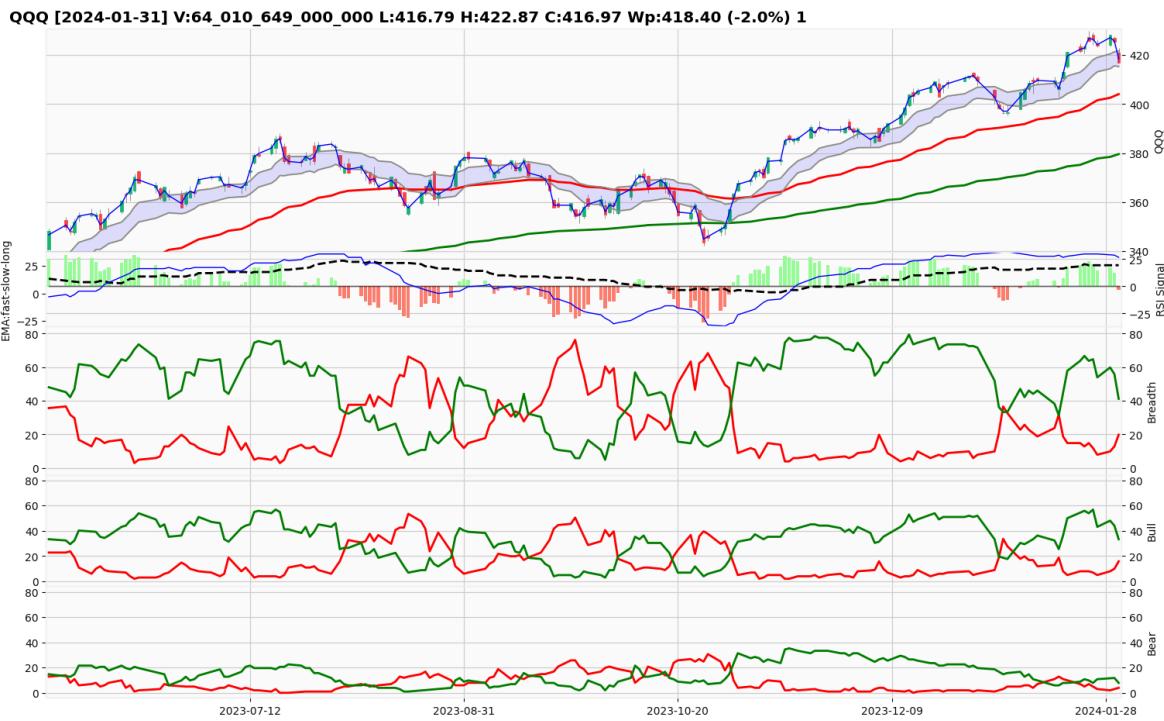


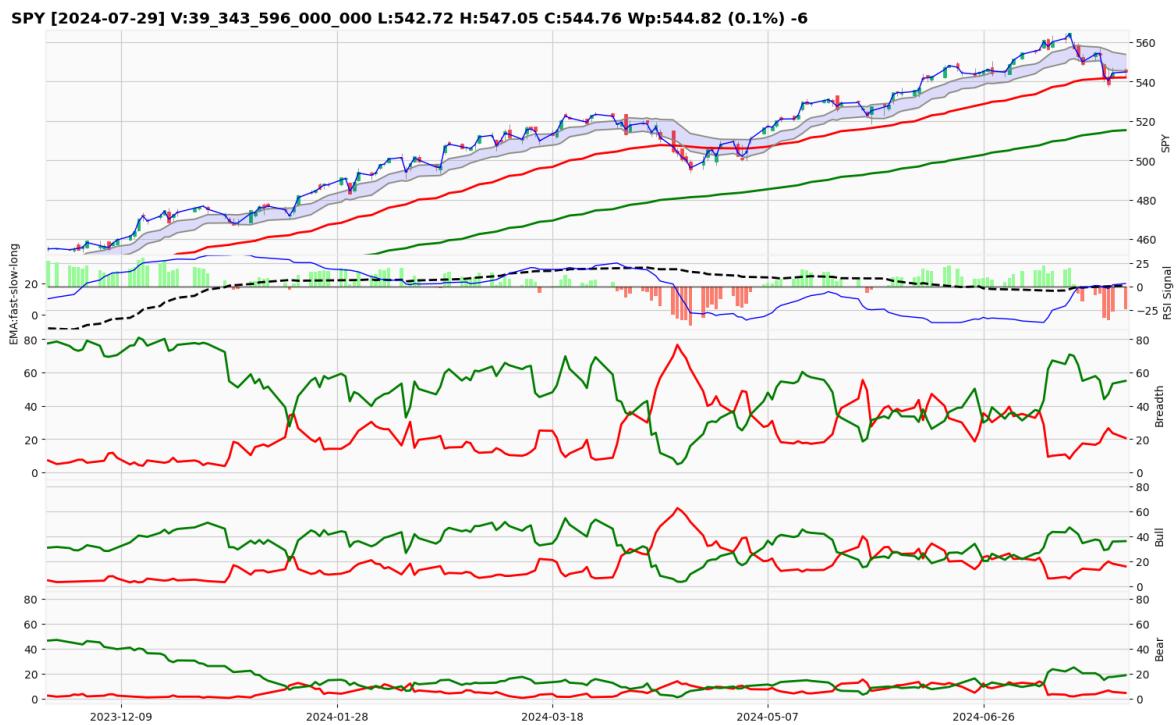
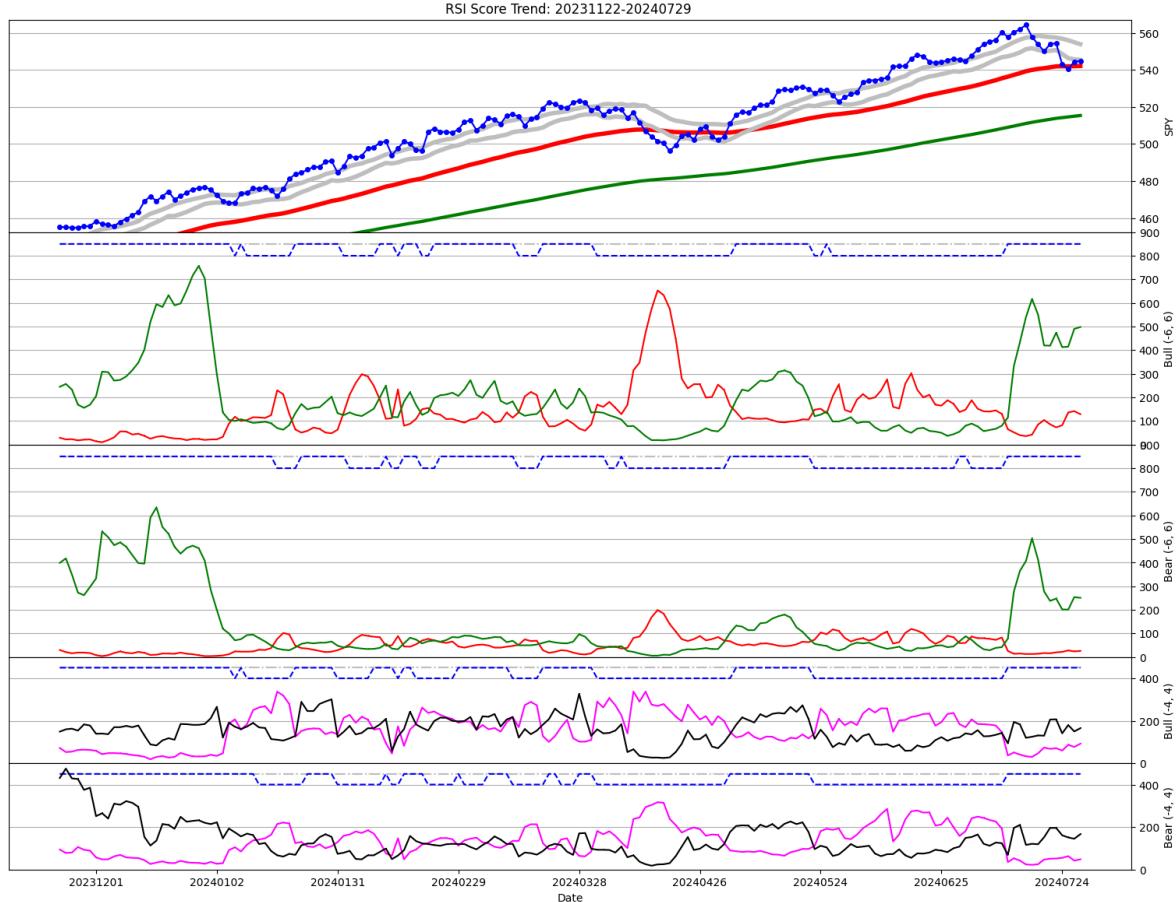


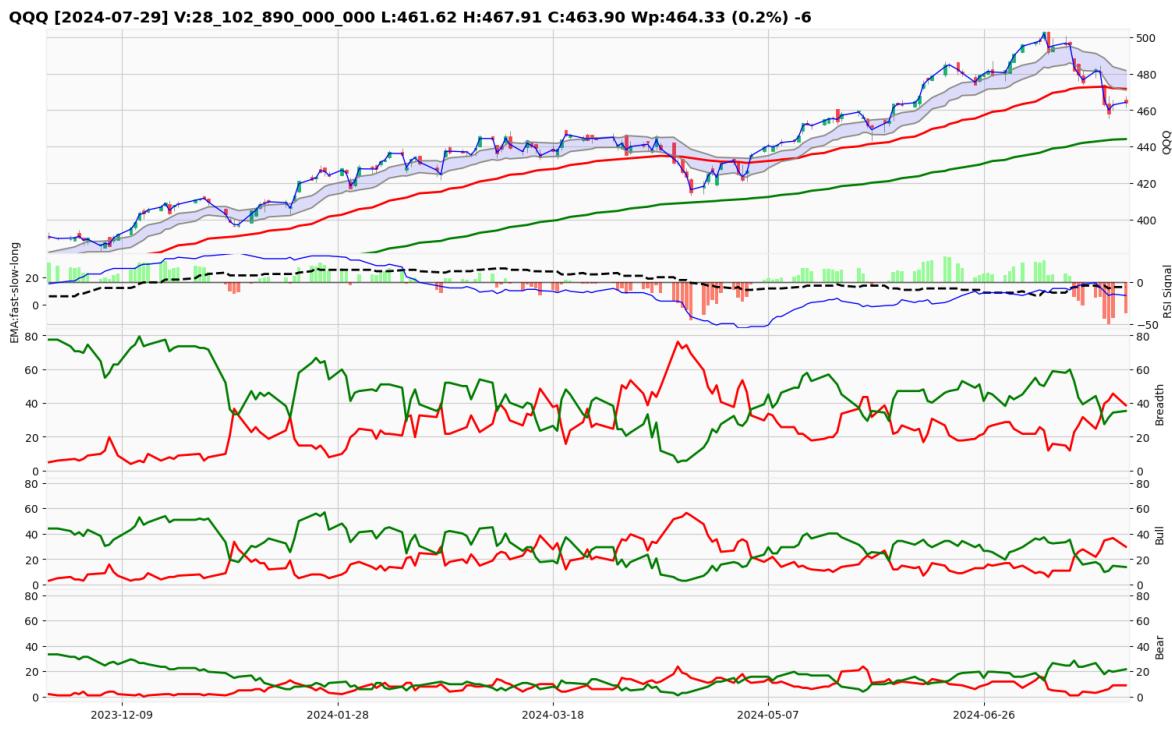












**IWM [2024-07-29] V:33\_780\_827\_000\_000 L:220.77 H:225.42 C:221.73 Wp:222.41 (-1.1%) 6**



[ ]

In [ ]:

In [ ]: