TimeSeries Momentum - India (Intraday Data)

Here I have created a time series momentum strategy for liquid futures contracts on Indian markets.

Methodology:

1. Aggregate Time Series Momentum

   Universe: Futures contracts on the 60 stocks in the Nifty
   Signal: N-day risk-scaled trend
   Leverage Constraint: 0 (sector neutral)
   Cost: 5 bps roundtrip
   Position Sizing: Proportional to indicator values scaled to a risk of 10%.

   The lookback and rebalancing frequencies we try are:

<table>
<thead>
<tr>
<th>Lookback</th>
<th>Rebalancing Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 day</td>
<td>60 min</td>
</tr>
<tr>
<td>5 day</td>
<td>30 min</td>
</tr>
<tr>
<td>3 days</td>
<td>60 min</td>
</tr>
<tr>
<td>1 day</td>
<td>60 min</td>
</tr>
<tr>
<td>1</td>
<td>30 min</td>
</tr>
</tbody>
</table>

2. Relative Sector Momentum

   Universe: All liquid futures contracts
   Signal: N-day risk-scaled trend
   Leverage Constraint: 0 (sector neutral)
   Cost: 5 bps roundtrip
   Position Sizing: proportional to indicator values scaled to a risk of 10%

   Portfolio Combination: Equal Weighted

   The lookback and rebalancing frequencies we try are:

<table>
<thead>
<tr>
<th>Lookback</th>
<th>Rebalancing Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 day</td>
<td>60 min</td>
</tr>
<tr>
<td>5 day</td>
<td>30 min</td>
</tr>
<tr>
<td>3 days</td>
<td>60 min</td>
</tr>
<tr>
<td>1 day</td>
<td>60 min</td>
</tr>
<tr>
<td>1</td>
<td>30 min</td>
</tr>
</tbody>
</table>

Authors:

- Sonam Srinivasa
- Gaumrith Chakravarty
- Manali Singh

Quant Project Management Link:

[Link]

---

Get Sector Information:

```python
if x in df['Sector'].tolist():
    sectors[x] = df.loc[df['Sector'] == x, 'Sector'].tolist()
```

Get data functions:

```python
def get_data(df):
    return df
```
def get_log_returns(symbol):
    return np.log(price['Close'] / price['Open'])

def get_relative_momentum_results(lookback, rebalancing_freq, resampling_freq):
    all_weights = ()
    all_returns = ()
    for sector in sectors:
        if sector == 'Nifty': continue
        print(sector, end=': ')
        symbols = sectors[sector]
        weights, final_return = get_final_results(lookback, rebalancing_freq, resampling_freq)
        all_weights.append(weights)
        all_returns.append(final_return)
    return

def get_aggregate_momentum_results(lookback, rebalancing_freq, resampling_freq):
    all_weights = ()
    all_returns = ()
    for sector in sectors:
        if sector == 'Nifty': continue
        print(sector, end=': ')
        symbols = sectors[sector]
        weights, final_return = get_final_results(lookback, rebalancing_freq, resampling_freq)
        all_weights.append(weights)
        all_returns.append(final_return)
    return

def getAnnualizedTurnover(allocation_df, freq=1):
    return allocation_df['allocation'].sum() / len(allocation_df)
Lookback = 5 days, rebalancing frequency = 30 mins

Lookback = 3 days, rebalancing frequency = 60 mins

Lookback = 3 days, rebalancing frequency = 30 mins

Lookback = 1 day, rebalancing frequency = 60 mins

Lookback = 1 days, rebalancing frequency = 30 mins

```
stats = get_formatted_summary_stats_from_df_cost(net_ret_consolidated.loc[['Annualized_returns', 'Annualized_stdev', 'worst_drawdown', 'Annual_sharpe_ratio_daily_returns', 'sortino_ratio', 'information_ratio'], :],
stats.loc[['Turnover'], :])

fig = plt.figure(figsize=(12, 6))
ax = fig.add_subplot(2, 1)
data = stats[['Lookback 5d, Rebal Freq 50m', 'Lookback 5d, Rebal Freq 50m', 'Lookback 3d, Rebal Freq 30m', 'Lookback 5d, Rebal Freq 50m']]
data.index = [1, 2, 3, 4]
data.plot(kind='bar')
plt.title('Annualized Returns')
plt.tight_layout()
plt.grid()
plt.legend()
ax.set_ylabel('Lookback (0)')
```

Relative Sector Timeseries Momentum

Lookback = 5 days, rebalancing frequency = 60 mins

global no_cost_ret = pd.DataFrame()

cost_ret = pd.DataFrame()

turnover = ()

gf = ()

no_cost(df) = pd.DataFrame()

def analysis(lookback, rebalancing_freq):

global no_cost_ret, cost_ret

if lookback, rebalancing_freq not in data_store.keys():
    all_returns_df, all_weights, eq_returns, stats = get_relative_sector_momentum_results(lookback, rebalancing_freq, resampling_freq='3T')
    data_store[lookback, rebalancing_freq] = [all_returns_df, all_weights, eq_returns, stats]
else:
    all_returns_df, all_weights, eq_returns, stats = data_store[lookback, rebalancing_freq]
    eq_returns['Aggregate'] = scaling_factor * get_normalized_stddev(eq_returns['Aggregate'])
    eq_returns['Aggregate'] = scaling_factor * weights_df * pd.DataFrame(all_weights.values(), axis1=weights_df)
    cost_ret = get_cost_adjusted_returns(eq_returns['Aggregate'], cost_trig, axis1)
    rets = pd.concat([eq_returns['Aggregate'], cost_trig], axis=1)

returns = (returns - 2) * 100

lookback=375
rebalancing_freq=60
analysis(lookback, rebalancing_freq)

Auto Ancillaries Energy MKG Finance Infra IT Metal Pharma PSB

Lookback = 5 days, rebalancing frequency = 30 min

lookback=375
rebalancing_freq=30
analysis(lookback, rebalancing_freq)

Auto Ancillaries Energy MKG Finance Infra IT Metal Pharma PSB

Lookback = 3 days, rebalancing frequency = 60 min

lookback=375
rebalancing_freq=60
analysis(lookback, rebalancing_freq)

Auto Ancillaries Energy MKG Finance Infra IT Metal Pharma PSB

Lookback = 3 days, rebalancing frequency = 30 min

lookback=375
rebalancing_freq=30
analysis(lookback, rebalancing_freq)

Auto Ancillaries Energy MKG Finance Infra IT Metal Pharma PSB

Lookback = 1 day, rebalancing frequency = 60 min

lookback=375
rebalancing_freq=60
analysis(lookback, rebalancing_freq)

Auto Ancillaries Energy MKG Finance Infra IT Metal Pharma PSB

Lookback = 1 day, rebalancing frequency = 30 min

lookback=375
rebalancing_freq=30
analysis(lookback, rebalancing_freq)

Auto Ancillaries Energy MKG Finance Infra IT Metal Pharma PSB
# Financial Performance Analysis

## Annualized Returns

<table>
<thead>
<tr>
<th>Stat</th>
<th>Lookback 3d, Rebal Freq 60m</th>
<th>Lookback 5d, Rebal Freq 60m</th>
<th>Lookback 3d, Rebal Freq 30m</th>
<th>Lookback 5d, Rebal Freq 30m</th>
<th>Lookback 1d, Rebal Freq 60m</th>
<th>Lookback 1d, Rebal Freq 30m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annualized Return</td>
<td>34.35</td>
<td>68.55</td>
<td>53.57</td>
<td>98.00</td>
<td>111.31</td>
<td>224.09</td>
</tr>
<tr>
<td>Annualized Vol</td>
<td>13.89</td>
<td>13.84</td>
<td>13.88</td>
<td>13.80</td>
<td>13.88</td>
<td>13.71</td>
</tr>
</tbody>
</table>

## Annual Sharpe Ratio Daily Returns

<table>
<thead>
<tr>
<th>Stat</th>
<th>Lookback 3d, Rebal Freq 60m</th>
<th>Lookback 5d, Rebal Freq 60m</th>
<th>Lookback 3d, Rebal Freq 30m</th>
<th>Lookback 5d, Rebal Freq 30m</th>
<th>Lookback 1d, Rebal Freq 60m</th>
<th>Lookback 1d, Rebal Freq 30m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annual Sharpe Ratio</td>
<td>2.07</td>
<td>4.82</td>
<td>3.74</td>
<td>7.06</td>
<td>8.65</td>
<td>16.14</td>
</tr>
<tr>
<td>Sortino Ratio</td>
<td>3.83</td>
<td>9.16</td>
<td>6.85</td>
<td>16.24</td>
<td>18.74</td>
<td>58.22</td>
</tr>
<tr>
<td>Information Ratio</td>
<td>2.21</td>
<td>4.03</td>
<td>3.61</td>
<td>6.77</td>
<td>7.91</td>
<td>15.36</td>
</tr>
</tbody>
</table>

## Today

- Annualized Returns
- Annualized Vol
- Annual Sharpe Ratio Daily Returns
- Sortino Ratio
- Information Ratio
- Turnover

---

```
plt.figure(figsize=(12, 6))
fig = plt.figure(figsize=(12, 6))
data = stats.loc[['Annualized Returns', 'Worst Drawdown', 'Lookback 3d, Rebal Freq 60m', 'Lookback 5d, Rebal Freq 60m', 'Lookback 3d, Rebal Freq 30m', 'Lookback 5d, Rebal Freq 30m']]
data = stats.loc[['Annualized Returns', 'Worst Drawdown', 'Lookback 3d, Rebal Freq 60m', 'Lookback 5d, Rebal Freq 60m', 'Lookback 3d, Rebal Freq 30m', 'Lookback 5d, Rebal Freq 30m']]
```

---

```
plt.figure(figsize=(12, 6))
fig = plt.figure(figsize=(12, 6))
data = stats.loc[['Annualized Returns', 'Worst Drawdown', 'Lookback 3d, Rebal Freq 60m', 'Lookback 5d, Rebal Freq 60m', 'Lookback 3d, Rebal Freq 30m', 'Lookback 5d, Rebal Freq 30m']]
data = stats.loc[['Annualized Returns', 'Worst Drawdown', 'Lookback 3d, Rebal Freq 60m', 'Lookback 5d, Rebal Freq 60m', 'Lookback 3d, Rebal Freq 30m', 'Lookback 5d, Rebal Freq 30m']]
```

---

```
plt.figure(figsize=(12, 6))
fig = plt.figure(figsize=(12, 6))
data = stats.loc[['Annualized Returns', 'Worst Drawdown', 'Lookback 3d, Rebal Freq 60m', 'Lookback 5d, Rebal Freq 60m', 'Lookback 3d, Rebal Freq 30m', 'Lookback 5d, Rebal Freq 30m']]
data = stats.loc[['Annualized Returns', 'Worst Drawdown', 'Lookback 3d, Rebal Freq 60m', 'Lookback 5d, Rebal Freq 60m', 'Lookback 3d, Rebal Freq 30m', 'Lookback 5d, Rebal Freq 30m']]
```