Alphalens Tear Sheets for

Factor - Historical Returns

for Period: 1/1/2018 to 4/20/2020 (last 28 Months)

In [1]:
```
# Import all required libraries
# Quantopian Libraries
from quantopian.pipeline import Pipeline
from quantopian.research import run_pipeline
from quantopian.pipeline.filters import QTradableStocksUS
from quantopian.pipeline.data import factset, USEquityPricing
from quantopian.pipeline.classifiers.fundamentals import Sector
from quantopian.pipeline.factors import Returns, SimpleMovingAverage, CustomFa
ctor, RSI
# Alpha Lens libraries
from alphalens.performance import mean_information_coefficient
from alphalens.utils import get_clean_factor_and_forward_returns
from alphalens.tears import create_information_tear_sheet, create_returns_tear
_sheet
```

Define Your Alpha Factor Here

Spend your time in this cell, creating good factors. Then simply run the rest of the notebook to analyze

In [2]:
```
# Define the factor (Historic Returns) for the Pipeline function
# The Date Range used is Jan-1,2018 to Apr-30,2020 (ie 28 months)
def make_pipeline():

    return Pipeline(
        columns={
        'hist_returns' : Returns(window_length=2) ,
        },
        screen=QTradableStocksUS()
    )

# Set the Start and End Dates for Factor Data
start_date='2018-01-01'
end_date='2020-04-30'

# Now run the Pipe
factor_data = run_pipeline(make_pipeline(), start_date, end_date)
```

Pipeline Execution Time: 5.54 Seconds

In [3]:
```
# Display factor data and notice the columns: Date, Equity and Historical Returns
print(type(factor_data))
print(len(factor_data))
factor_data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
1279633
```

Out[3]:

		hist_returns
	Equity(2 [HWM])	-0.012855
	Equity(24 [AAPL])	-0.010347
2018-01-02 00:00:00+00:00	Equity(31 [ABAX])	-0.005823
	Equity(41 [ARCB])	-0.016506
	Equity(52 [ABM])	-0.004752

In [4]:
```
# Set the Start and End Dates and Get Pricing Data for the equities in the Factor Data
start_date='2018-01-01'
end_date='2020-05-31'  # End Date for Pricing Data should be > End Date for Factor Data for Forward Returns
pricing_data = get_pricing(factor_data.index.levels[1], start_date, end_date, fields='open_price')
```

In [5]:
```
# Display Pricing data and notice the columns: Date, Equity and Price
print(type(pricing_data))
print(len(pricing_data))
pricing_data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
606
```

Out[5]:

	Equity(2 [HWM])	Equity(24 [AAPL])	Equity(31 [ABAX])	Equity(41 [ARCB])	Equity(52 [ABM])	Equity(53 [ABMD])	Equity(62 [ABT])	Equity([GOL
2018-01-02 00:00:00+00:00	26.835	164.326	49.347	34.650	35.880	188.13	55.748	14.2
2018-01-03 00:00:00+00:00	27.130	166.615	49.845	36.161	35.809	193.29	56.504	14.7
2018-01-04 00:00:00+00:00	27.709	166.624	49.477	35.820	35.990	198.00	56.993	14.5
2018-01-05 00:00:00+00:00	28.741	167.493	50.791	35.820	36.656	200.78	56.552	14.5
2018-01-08 00:00:00+00:00	29.164	168.372	54.872	35.187	37.151	208.24	56.370	14.4

5 rows × 2748 columns

In [6]:
```
# Merge the Factor Data and Pricing Data using the Function from Alphalens
merged_data = get_clean_factor_and_forward_returns(
    factor = factor_data['hist_returns'],
    prices = pricing_data,
    quantiles=5,
    periods=(1,),
    max_loss=100,
)
```

```
Dropped 0.0% entries from factor data: 0.0% in forward returns computation an
d 0.0% in binning phase (set max_loss=0 to see potentially suppressed Excepti
ons).
max_loss is 10000.0%, not exceeded: OK!
```

In [7]:
```
# Display Merged data and notice the columns
print(type(merged_data))
print(len(merged_data))
merged_data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
1279096
```

Out[7]:

		1D	factor	factor_quantile
date	**asset**			
	Equity(2 [HWM])	0.010993	-0.012855	2
	Equity(24 [AAPL])	0.013930	-0.010347	2
2018-01-02 00:00:00+00:00	**Equity(31 [ABAX])**	0.010092	-0.005823	3
	Equity(41 [ARCB])	0.043608	-0.016506	1
	Equity(52 [ABM])	-0.001979	-0.004752	3

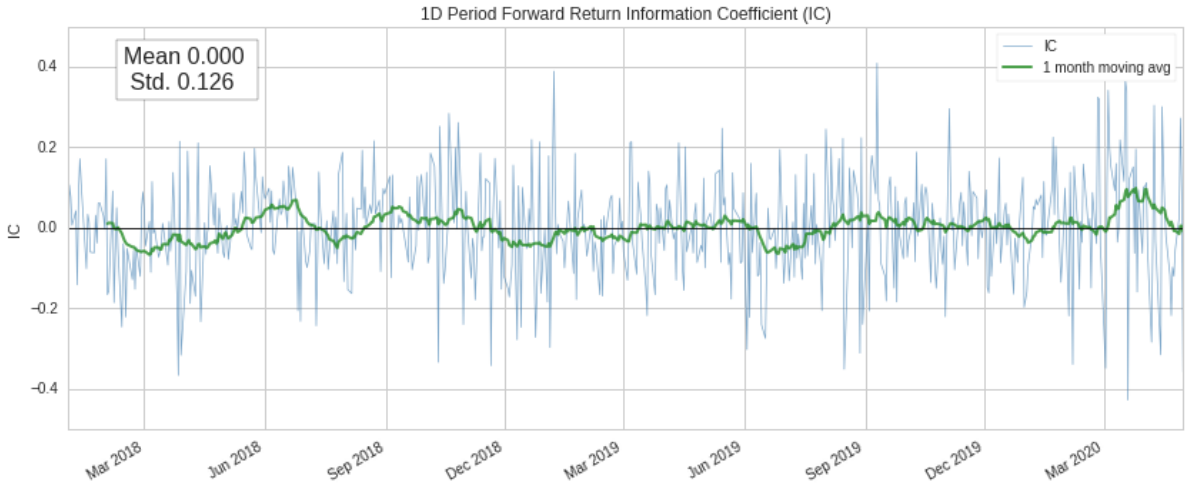
In [8]:
```
# Create Information Tear Sheets using Alphalens
create_information_tear_sheet(merged_data)
```

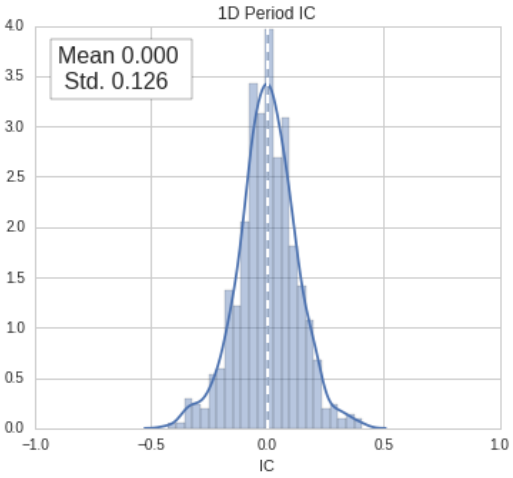
```
Information Analysis
```

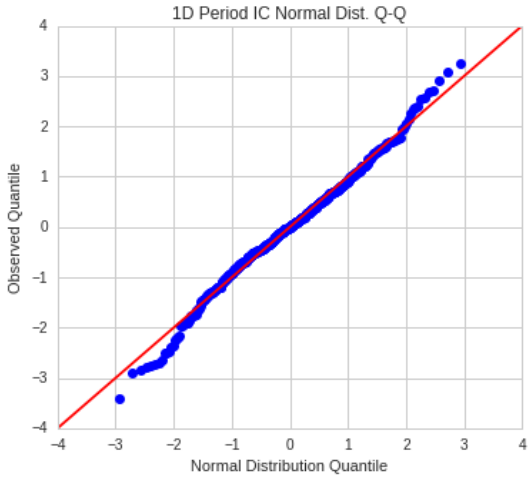
	1D
IC Mean	0.000
IC Std.	0.126
Risk-Adjusted IC	0.002
t-stat(IC)	0.045
p-value(IC)	0.964
IC Skew	-0.099
IC Kurtosis	0.653

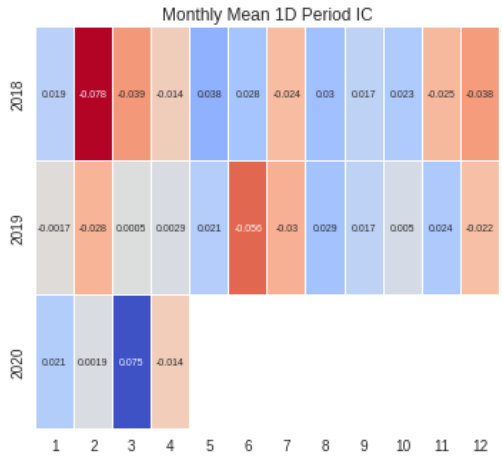
```
/venvs/py35/lib/python3.5/site-packages/statsmodels/nonparametric/kdetools.p
y:20: VisibleDeprecationWarning: using a non-integer number instead of an int
eger will result in an error in the future
  y = X[:m/2+1] + np.r_[0,X[m/2+1:],0]*1j
```

```
<matplotlib.figure.Figure at 0x7f24e2dc4a20>
```


1D Period Forward Return Information Coefficient (IC)
Mean 0.000
Std. 0.126
IC
1 month moving avg
IC
0.4
0.2
0.0
−0.2
−0.4
Mar 2018
Jun 2018
Sep 2018
Dec 2018
Mar 2019
Jun 2019
Sep 2019
Dec 2019
Mar 2020


1D Period IC
Mean 0.000
Std. 0.126
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
1.0
IC


1D Period IC Normal Dist. Q-Q
Observed Quantile
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4
Normal Distribution Quantile

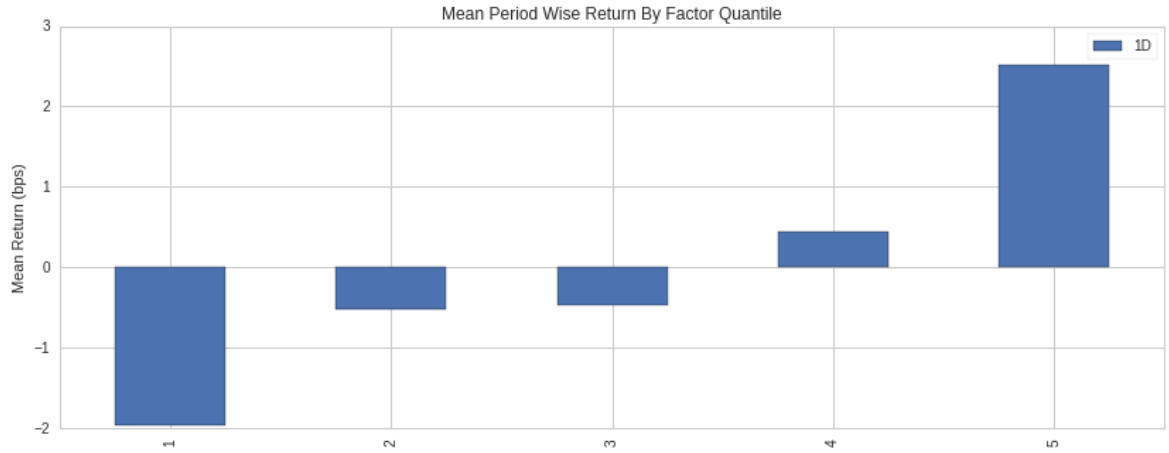

Monthly Mean 1D Period IC
2018
2019
2020
0.019
-0.078
-0.039
-0.014
0.038
0.028
-0.024
0.03
0.017
0.023
-0.025
-0.038
-0.0017
-0.028
0.0065
0.0029
0.021
-0.056
-0.03
0.029
0.017
0.005
0.024
-0.022
0.021
0.0019
0.075
-0.014
1
2
3
4
5
6
7
8
9
10
11
12

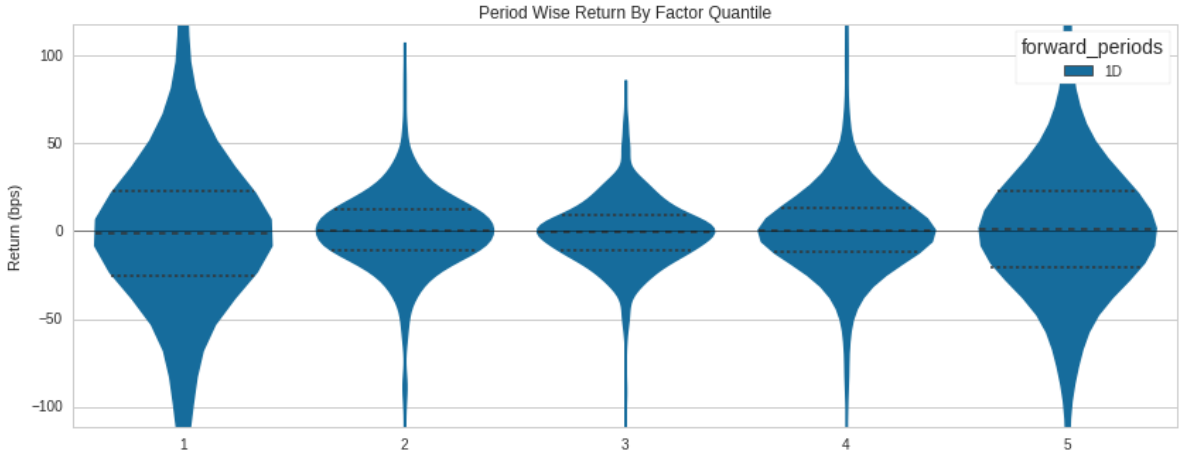
In [9]:
```
# Create Returns Tear Sheets using Alphalens
create_returns_tear_sheet(merged_data)
```

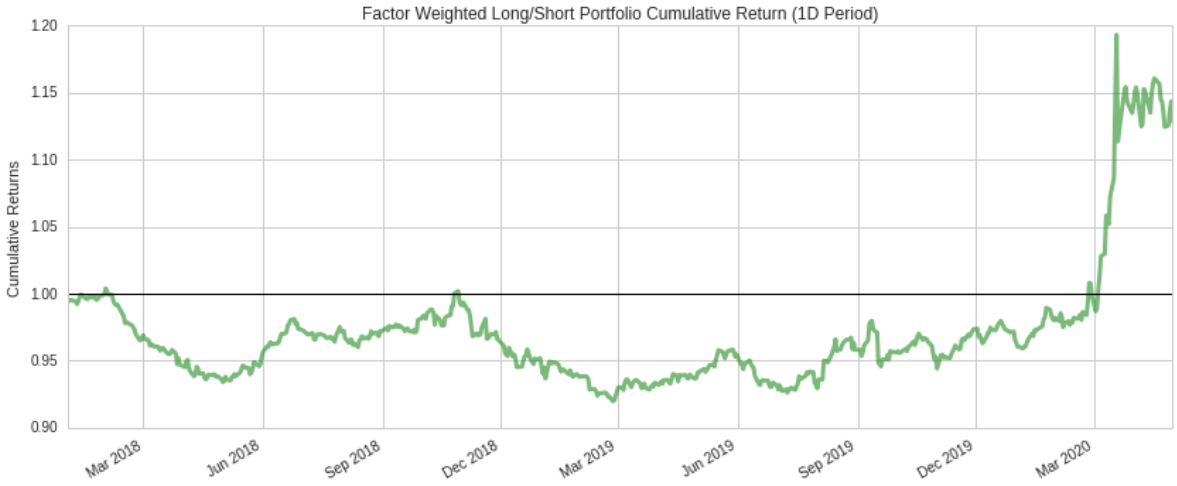
Returns Analysis

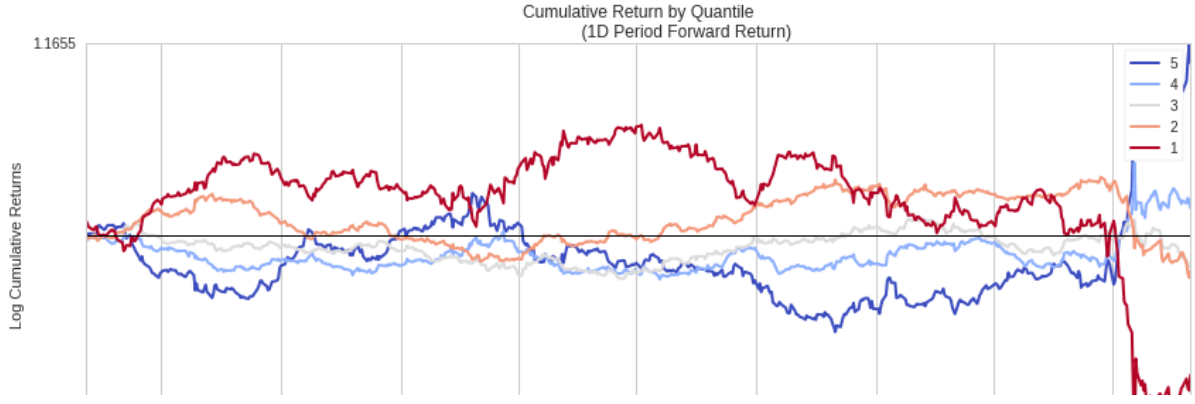
	1D
Ann. alpha	0.057
beta	-0.137
Mean Period Wise Return Top Quantile (bps)	2.518
Mean Period Wise Return Bottom Quantile (bps)	-1.964
Mean Period Wise Spread (bps)	4.482

<matplotlib.figure.Figure at 0x7f24dde33ac8>

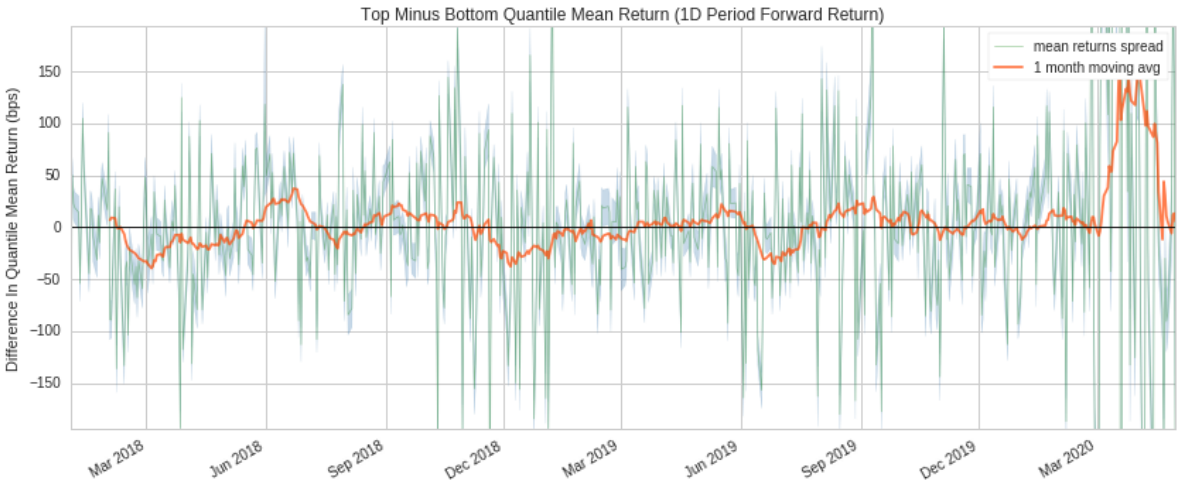

Mean Period Wise Return By Factor Quantile
Mean Return (bps)
1D
3
2
1
0
−1
−2
1
2
3
4
5


Period Wise Return By Factor Quantile
forward_periods
1D
Return (bps)
100
50
0
−50
−100
1
2
3
4
5


Factor Weighted Long/Short Portfolio Cumulative Return (1D Period)
Cumulative Returns
1.20
1.15
1.10
1.05
1.00
0.95
0.90
Mar 2018
Jun 2018
Sep 2018
Dec 2018
Mar 2019
Jun 2019
Sep 2019
Dec 2019
Mar 2020


Cumulative Return by Quantile
(1D Period Forward Return)
Log Cumulative Returns
1.1655
5
4
3
2
1


0.8178
Mar 2018
Jun 2018
Sep 2018
Dec 2018
Mar 2019
Jun 2019
Sep 2019
Dec 2019
Mar 2020


Top Minus Bottom Quantile Mean Return (1D Period Forward Return)
Difference In Quantile Mean Return (bps)
150
100
50
0
−50
−100
−150
mean returns spread
1 month moving avg
Mar 2018
Jun 2018
Sep 2018
Dec 2018
Mar 2019
Jun 2019
Sep 2019
Dec 2019
Mar 2020

In [ ]: