DS-670 Weighted ...

```
%pyspark
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
import timeit
```

FINISHED

```
%pyspark
df = DataFrame({'category': ['a','a','a','a','b','b','b','b'], 'data': np.random.randn(8),
df
```

FINISHED

	category	data	weights
0	a	-2.529232	0.433668
1	a	-0.132285	0.107664
2	a	1.784336	0.273748
3	a	0.991449	0.826221
4	b	-0.844074	0.548237
5	b	0.060158	0.203452
6	b	0.627685	0.619496
7	b	-0.612839	0.071068

```
%pyspark
start = timeit.timeit()

df = DataFrame({'category': ['a','a','a','a','b','b','b','b'], 'data': np.random.randn(8),
df
end = timeit.timeit()
print(end - start)
```

FINISHED

```
2.09808349609e-05
```

```
%pyspark

grouped = df.groupby('category')
get_wavg = lambda g: np.average(g['data'], weights=g['weights'])
grouped.apply(get_wavg)
```

FINISHED

```
category
a   -0.628943
b   -0.224998
dtype: float64
```

```
%pyspark
start = timeit.timeit()
grouped = df.groupby('category')
get_wavg = lambda g: np.average(g['data'], weights=g['weights'])
grouped.apply(get_wavg)
end = timeit.timeit()
print(end - start)
```

FINISHED

```
0.00119829177856
```

```
%pyspark
# This data is from Yahoo Finance data set. It has the S&P 500 stock market from 2000-03-3(
close_px = pd.read_csv('/Users/Mmohamar/Downloads/table.csv', parse_dates=True, index_col=(
close_px.info()
```

FINISHED

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 4276 entries, 2017-03-29 to 2000-03-30
Data columns (total 6 columns):
Open         4276 non-null float64
High         4276 non-null float64
Low          4276 non-null float64
Close        4276 non-null float64
Volume       4276 non-null int64
Adj Close    4276 non-null float64
dtypes: float64(5), int64(1)
memory usage: 233.8 KB
```

```
%pyspark
close_px[-4:]
```

FINISHED

```
                 Open         High          Low        Close      Volume  \
Date
2000-04-04  1505.979980  1526.449951  1416.410034  1494.729980  1515460000
2000-04-03  1498.579956  1507.189941  1486.959961  1505.969971  1021700000
2000-03-31  1487.920044  1519.810059  1484.380005  1498.579956  1227400000
2000-03-30  1508.520020  1517.380005  1474.630005  1487.920044  1193400000
              Adj Close
Date
2000-04-04  1494.729980
2000-04-03  1505.969971
2000-03-31  1498.579956
2000-03-30  1487.920044
```

```
%pyspark
start = timeit.timeit()
close_px[-4:]
end = timeit.timeit()
print(end - start)
```

FINISHED

0.0219159126282

```
%pyspark
rets = close_px.pct_change().dropna()
spx_corr = lambda x: x.corrwith(x['Close'])
by_year = rets.groupby(lambda x: x.year)
by_year.apply(spx_corr)
```

FINISHED

	Open	High	Low	Close	Volume	Adj Close
2000	-0.009966	0.611983	0.617876	1.0	-0.163299	1.0
2001	0.013880	0.589494	0.581842	1.0	0.005475	1.0
2002	-0.024179	0.632065	0.514824	1.0	0.017073	1.0
2003	-0.139924	0.521516	0.483188	1.0	0.223159	1.0
2004	0.003695	0.556858	0.577184	1.0	-0.067367	1.0
2005	-0.103747	0.551393	0.555781	1.0	-0.042278	1.0
2006	0.010322	0.573907	0.574508	1.0	-0.032910	1.0
2007	-0.138275	0.452698	0.630547	1.0	-0.209898	1.0
2008	-0.092710	0.484550	0.560502	1.0	-0.098464	1.0
2009	-0.037061	0.620987	0.600489	1.0	0.093216	1.0
2010	0.058897	0.588977	0.606921	1.0	-0.133493	1.0
2011	-0.086071	0.579418	0.608427	1.0	0.044650	1.0
2012	0.062405	0.658301	0.635668	1.0	0.017892	1.0
2013	-0.034737	0.574304	0.663911	1.0	-0.145132	1.0
2014	0.082195	0.599605	0.720652	1.0	-0.174004	1.0
2015	0.123495	0.673705	0.693994	1.0	-0.118394	1.0
2016	0.050678	0.674122	0.694677	1.0	-0.131279	1.0

```
%pyspark
start = timeit.timeit()
rets = close_px.pct_change().dropna()
spx_corr = lambda x: x.corrwith(x['Close'])
by_year = rets.groupby(lambda x: x.year)
by_year.apply(spx_corr)
end = timeit.timeit()
print(end - start)
```

FINISHED

```
-0.0220007896423
```

```
%pyspark
# Annual correlation of Apple with Microsoft
by_year.apply(lambda g: g['High'].corr(g['Low']))
```

FINISHED

```
2000    0.620230
2001    0.650804
2002    0.714291
2003    0.578030
2004    0.575314
2005    0.622040
2006    0.577264
2007    0.607131
2008    0.723168
2009    0.677464
2010    0.594464
2011    0.681434
2012    0.614203
2013    0.627911
2014    0.733010
2015    0.755769
2016    0.742851
2017    0.707768
```

FINISHED

```
%pyspark
start = timeit.timeit()
by_year.apply(lambda g: g['High'].corr(g['Low']))
end = timeit.timeit()
print(end - start)
```

```
0.000125169754028
```

FINISHED

```
%pyspark
# applying Ordinary Least Squares (OLS) regression on each chunk of data

import numpy as np
import statsmodels.api as sm
def regression(data, yvar, xvars):
  Y = data[yvar]
  X = data[xvars]
  X['intercept'] = 1.
  result = sm.OLS(Y,X).fit()
  return result.params
```

FINISHED

```
%pyspark
by_year.apply(regression,'Close',['Volume'])
```

	Volume	intercept
2000	-0.008068	0.001088
2001	0.000269	0.000510
2002	0.001276	0.001059
2003	0.011591	-0.000947
2004	-0.002866	-0.000264
2005	-0.001568	-0.000176
2006	-0.000981	-0.000404
2007	-0.008549	0.000143
2008	-0.009129	0.002320
2009	0.008566	-0.000764
2010	-0.006471	-0.000240
2011	0.002458	0.000015
2012	0.000559	-0.000520
2013	-0.005324	-0.000783
2014	-0.007359	-0.000341
2015	-0.003146	0.000233
2016	-0.006037	-0.000332

```
%pyspark
import timeit
start = timeit.timeit()
by_year.apply(regression,'Close',['Volume'])
end = timeit.timeit()
print(end - start)
```

FINISHED

```
0.0260169506073
```

```
%pyspark
```

FINISHED