

Pandas-styles

```
In [8]: from sklearn.datasets import load_boston
import pandas as pd
```

```
In [9]: boston = load_boston()
```

```
In [10]: df = pd.DataFrame(boston['data'], columns=boston['feature_names'])
df['price'] = boston['target']
```

```
In [11]: hdf = df.head(10)
hdf
```

Out[11]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.35
5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3.0	222.0	18.7	394.12	5.21
6	0.08829	12.5	7.87	0.0	0.524	6.012	66.6	5.5605	5.0	311.0	15.2	395.60	12.43
7	0.14455	12.5	7.87	0.0	0.524	6.172	96.1	5.9505	5.0	311.0	15.2	396.90	19.15
8	0.21124	12.5	7.87	0.0	0.524	5.631	100.0	6.0821	5.0	311.0	15.2	386.63	29.93
9	0.17004	12.5	7.87	0.0	0.524	6.004	85.9	6.5921	5.0	311.0	15.2	386.71	17.10

```
In [14]: hdf.style.highlight_max()
```

```
Out[14]:
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	
0	0.006320	18.000000	2.310000	0.000000	0.538000	6.575000	65.200000	4.090000	1.000000	2:
1	0.027310	0.000000	7.070000	0.000000	0.469000	6.421000	78.900000	4.967100	2.000000	2:
2	0.027290	0.000000	7.070000	0.000000	0.469000	7.185000	61.100000	4.967100	2.000000	2:
3	0.032370	0.000000	2.180000	0.000000	0.458000	6.998000	45.800000	6.062200	3.000000	2:
4	0.069050	0.000000	2.180000	0.000000	0.458000	7.147000	54.200000	6.062200	3.000000	2:
5	0.029850	0.000000	2.180000	0.000000	0.458000	6.430000	58.700000	6.062200	3.000000	2:
6	0.088290	12.500000	7.870000	0.000000	0.524000	6.012000	66.600000	5.560500	5.000000	3
7	0.144550	12.500000	7.870000	0.000000	0.524000	6.172000	96.100000	5.950500	5.000000	3
8	0.211240	12.500000	7.870000	0.000000	0.524000	5.631000	100.000000	6.082100	5.000000	3
9	0.170040	12.500000	7.870000	0.000000	0.524000	6.004000	85.900000	6.592100	5.000000	3

```
In [15]: hdf.style.background_gradient()
```

```
Out[15]:
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	
0	0.006320	18.000000	2.310000	0.000000	0.538000	6.575000	65.200000	4.090000	1.000000	2:
1	0.027310	0.000000	7.070000	0.000000	0.469000	6.421000	78.900000	4.967100	2.000000	2:
2	0.027290	0.000000	7.070000	0.000000	0.469000	7.185000	61.100000	4.967100	2.000000	2:
3	0.032370	0.000000	2.180000	0.000000	0.458000	6.998000	45.800000	6.062200	3.000000	2:
4	0.069050	0.000000	2.180000	0.000000	0.458000	7.147000	54.200000	6.062200	3.000000	2:
5	0.029850	0.000000	2.180000	0.000000	0.458000	6.430000	58.700000	6.062200	3.000000	2:
6	0.088290	12.500000	7.870000	0.000000	0.524000	6.012000	66.600000	5.560500	5.000000	3
7	0.144550	12.500000	7.870000	0.000000	0.524000	6.172000	96.100000	5.950500	5.000000	3
8	0.211240	12.500000	7.870000	0.000000	0.524000	5.631000	100.000000	6.082100	5.000000	3
9	0.170040	12.500000	7.870000	0.000000	0.524000	6.004000	85.900000	6.592100	5.000000	3

In [16]: `hdf.style.bar()`

Out[16]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	
0	0.006320	18.000000	2.310000	0.000000	0.538000	6.575000	65.200000	4.090000	1.000000	2:
1	0.027310	0.000000	7.070000	0.000000	0.469000	6.421000	78.900000	4.967100	2.000000	2:
2	0.027290	0.000000	7.070000	0.000000	0.469000	7.185000	61.100000	4.967100	2.000000	2:
3	0.032370	0.000000	2.180000	0.000000	0.458000	6.998000	45.800000	6.062200	3.000000	2:
4	0.069050	0.000000	2.180000	0.000000	0.458000	7.147000	54.200000	6.062200	3.000000	2:
5	0.029850	0.000000	2.180000	0.000000	0.458000	6.430000	58.700000	6.062200	3.000000	2:
6	0.088290	12.500000	7.870000	0.000000	0.524000	6.012000	66.600000	5.560500	5.000000	3
7	0.144550	12.500000	7.870000	0.000000	0.524000	6.172000	96.100000	5.950500	5.000000	3
8	0.211240	12.500000	7.870000	0.000000	0.524000	5.631000	100.000000	6.082100	5.000000	3
9	0.170040	12.500000	7.870000	0.000000	0.524000	6.004000	85.900000	6.592100	5.000000	3

In [17]:

```
def odd_green(val):
    if val % 2 != 0:
        return 'color: green'
    return ''
```

In [18]: `hdf.style.applymap(odd_green)`

Out[18]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	
0	0.006320	18.000000	2.310000	0.000000	0.538000	6.575000	65.200000	4.090000	1.000000	2:
1	0.027310	0.000000	7.070000	0.000000	0.469000	6.421000	78.900000	4.967100	2.000000	2:
2	0.027290	0.000000	7.070000	0.000000	0.469000	7.185000	61.100000	4.967100	2.000000	2:
3	0.032370	0.000000	2.180000	0.000000	0.458000	6.998000	45.800000	6.062200	3.000000	2:
4	0.069050	0.000000	2.180000	0.000000	0.458000	7.147000	54.200000	6.062200	3.000000	2:
5	0.029850	0.000000	2.180000	0.000000	0.458000	6.430000	58.700000	6.062200	3.000000	2:
6	0.088290	12.500000	7.870000	0.000000	0.524000	6.012000	66.600000	5.560500	5.000000	3
7	0.144550	12.500000	7.870000	0.000000	0.524000	6.172000	96.100000	5.950500	5.000000	3
8	0.211240	12.500000	7.870000	0.000000	0.524000	5.631000	100.000000	6.082100	5.000000	3
9	0.170040	12.500000	7.870000	0.000000	0.524000	6.004000	85.900000	6.592100	5.000000	3

In [20]:

```
def orange10(val):
    if val > 10:
        return 'background-color: orange'
    return ''
```

```
In [21]: hdf.style.applymap(odd_green).applymap(orange10)
```

Out[21]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	
0	0.006320	18.000000	2.310000	0.000000	0.538000	6.575000	65.200000	4.090000	1.000000	2:
1	0.027310	0.000000	7.070000	0.000000	0.469000	6.421000	78.900000	4.967100	2.000000	2:
2	0.027290	0.000000	7.070000	0.000000	0.469000	7.185000	61.100000	4.967100	2.000000	2:
3	0.032370	0.000000	2.180000	0.000000	0.458000	6.998000	45.800000	6.062200	3.000000	2:
4	0.069050	0.000000	2.180000	0.000000	0.458000	7.147000	54.200000	6.062200	3.000000	2:
5	0.029850	0.000000	2.180000	0.000000	0.458000	6.430000	58.700000	6.062200	3.000000	2:
6	0.088290	12.500000	7.870000	0.000000	0.524000	6.012000	66.600000	5.560500	5.000000	3
7	0.144550	12.500000	7.870000	0.000000	0.524000	6.172000	96.100000	5.950500	5.000000	3
8	0.211240	12.500000	7.870000	0.000000	0.524000	5.631000	100.000000	6.082100	5.000000	3
9	0.170040	12.500000	7.870000	0.000000	0.524000	6.004000	85.900000	6.592100	5.000000	3

```
In [22]: hdf.style.applymap(odd_green, subset=['ZN', 'RAD'])
```

Out[22]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	
0	0.006320	18.000000	2.310000	0.000000	0.538000	6.575000	65.200000	4.090000	1.000000	2:
1	0.027310	0.000000	7.070000	0.000000	0.469000	6.421000	78.900000	4.967100	2.000000	2:
2	0.027290	0.000000	7.070000	0.000000	0.469000	7.185000	61.100000	4.967100	2.000000	2:
3	0.032370	0.000000	2.180000	0.000000	0.458000	6.998000	45.800000	6.062200	3.000000	2:
4	0.069050	0.000000	2.180000	0.000000	0.458000	7.147000	54.200000	6.062200	3.000000	2:
5	0.029850	0.000000	2.180000	0.000000	0.458000	6.430000	58.700000	6.062200	3.000000	2:
6	0.088290	12.500000	7.870000	0.000000	0.524000	6.012000	66.600000	5.560500	5.000000	3
7	0.144550	12.500000	7.870000	0.000000	0.524000	6.172000	96.100000	5.950500	5.000000	3
8	0.211240	12.500000	7.870000	0.000000	0.524000	5.631000	100.000000	6.082100	5.000000	3
9	0.170040	12.500000	7.870000	0.000000	0.524000	6.004000	85.900000	6.592100	5.000000	3

```
In [24]: def top20(col):
            is_top20 = col > col.quantile(.8)
            return ['font-weight: bold' if v else '' for v in is_top20]
```

In [25]: `hdf.style.apply(top20)`

Out[25]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	
0	0.006320	18.000000	2.310000	0.000000	0.538000	6.575000	65.200000	4.090000	1.000000	2:
1	0.027310	0.000000	7.070000	0.000000	0.469000	6.421000	78.900000	4.967100	2.000000	2:
2	0.027290	0.000000	7.070000	0.000000	0.469000	7.185000	61.100000	4.967100	2.000000	2:
3	0.032370	0.000000	2.180000	0.000000	0.458000	6.998000	45.800000	6.062200	3.000000	2:
4	0.069050	0.000000	2.180000	0.000000	0.458000	7.147000	54.200000	6.062200	3.000000	2:
5	0.029850	0.000000	2.180000	0.000000	0.458000	6.430000	58.700000	6.062200	3.000000	2:
6	0.088290	12.500000	7.870000	0.000000	0.524000	6.012000	66.600000	5.560500	5.000000	3
7	0.144550	12.500000	7.870000	0.000000	0.524000	6.172000	96.100000	5.950500	5.000000	3
8	0.211240	12.500000	7.870000	0.000000	0.524000	5.631000	100.000000	6.082100	5.000000	3
9	0.170040	12.500000	7.870000	0.000000	0.524000	6.004000	85.900000	6.592100	5.000000	3

In [36]: `import qgrid`

In [35]: `pip install --upgrade qgrid`

```
Collecting qgrid
  Downloading qgrid-1.3.1.tar.gz (889 kB)
Requirement already satisfied, skipping upgrade: notebook>=4.0.0 in c:\users\
\anal\anaconda3\lib\site-packages (from qgrid) (6.0.3)
Requirement already satisfied, skipping upgrade: pandas>=0.18.0 in c:\users\da
nal\anaconda3\lib\site-packages (from qgrid) (1.0.1)
Requirement already satisfied, skipping upgrade: ipywidgets>=7.0.0 in c:\user
s\anal\anaconda3\lib\site-packages (from qgrid) (7.5.1)
Requirement already satisfied, skipping upgrade: ipython-genutils in c:\users
\anal\anaconda3\lib\site-packages (from notebook>=4.0.0->qgrid) (0.2.0)
Requirement already satisfied, skipping upgrade: Send2Trash in c:\users\anal\
anaconda3\lib\site-packages (from notebook>=4.0.0->qgrid) (1.5.0)
Requirement already satisfied, skipping upgrade: prometheus-client in c:\user
s\anal\anaconda3\lib\site-packages (from notebook>=4.0.0->qgrid) (0.7.1)
Requirement already satisfied, skipping upgrade: jupyter-client>=5.3.4 in
c:\users\anal\anaconda3\lib\site-packages (from notebook>=4.0.0->qgrid) (5.
3.4)
Requirement already satisfied, skipping upgrade: jupyter-core>=4.6.1 in c:\us
ers\anal\anaconda3\lib\site-packages (from notebook>=4.0.0->qgrid) (4.6.1)
Requirement already satisfied, skipping upgrade: jupyterlab in c:\users\anal\
```

In []: `qgrid.show_grid(hdf)` *#This method doesn't work here.*

