



Numerical Data Transformations

- `h2o_frame[x].abs()`
- `h2o_frame[x].acos()`
- `h2o_frame[x].acosh()`
- `h2o_frame[x].asin()`
- `h2o_frame[x].asinh()`
- `h2o_frame[x].atan()`
- `h2o_frame[x].atanh()`
- `h2o_frame[x].ceil()`
- `h2o_frame[x].cos()`
- `h2o_frame[x].cosh()`
- `h2o_frame[x].cospi()`
- `h2o_frame[x].cut(breaks, ...)`
- `h2o_frame[x].digamma()`
- `h2o_frame[x].exp()`
- `h2o_frame[x].expm1()`
- `h2o_frame[x].floor()`
- `h2o_frame[x].gamma()`
- `h2o_frame[x].lgamma()`

- `h2o_frame[x].log()`
- `h2o_frame[x].log10()`
- `h2o_frame[x].log1p()`
- `h2o_frame[x].log2()`
- `h2o_frame[x].round(digits=0)`
- `h2o_frame[x].scale(center=True, scale=True)`
- `h2o_frame[x].sign()`
- `h2o_frame[x].signif(digits=6)`
- `h2o_frame[x].sin()`
- `h2o_frame[x].sinh()`
- `h2o_frame[x].sinpi()`
- `h2o_frame[x].sqrt()`
- `h2o_frame[x].tan()`
- `h2o_frame[x].tanh()`
- `h2o_frame[x].tanpi()`
- `h2o_frame[x].trigamma()`
- `h2o_frame[x].trunc()`

Numeric Data Transformations

Transformation for skewed data with positive and negative values

$$\text{pseudoLog10}(x) = \text{asinh}(x/2) / \log(10)$$

```
1 import math
2
3 def pseudo_log10(x):
4     return math.asinh(x / 2) / math.log(10)
5
6 print("pseudo_log10(\261{}) = \261{:0.6f}".format(100000, pseudo_log10(100000)))
7 print("pseudo_log10(\261{}) = \261{:0.6f}".format(10000, pseudo_log10(10000)))
8 print("pseudo_log10(\261{}) = \261{:0.6f}".format(1000, pseudo_log10(1000)))
9 print("pseudo_log10(\261{}) = \261{:0.6f}".format(100, pseudo_log10(100)))
10 print("pseudo_log10(\261{}) = \261{:0.6f}".format(10, pseudo_log10(10)))
11 print("pseudo_log10(\261{}) = \261{:0.6f}".format(1, pseudo_log10(1)))
12 print("pseudo_log10({}) = {}".format(0, pseudo_log10(0)))
```

```
pseudo_log10(±100000) = ±5.000000
pseudo_log10(±10000)  = ±4.000000
pseudo_log10(±1000)   = ±3.000000
pseudo_log10(±100)    = ±2.000043
pseudo_log10(±10)     = ±1.004279
pseudo_log10(±1)      = ±0.208988
pseudo_log10(0)       = 0.0
```

Numeric Data Transformations

- `h2o_frame[x].abs()`
- `h2o_frame[x].acos()`
- `h2o_frame[x].acosh()`
- `h2o_frame[x].asin()`
- `h2o_frame[x].asinh()`
- `h2o_frame[x].atan()`
- `h2o_frame[x].atanh()`
- `h2o_frame[x].ceil()`
- `h2o_frame[x].cos()`
- `h2o_frame[x].cosh()`
- `h2o_frame[x].cospi()`
- `h2o_frame[x].cut(breaks, ...)`
- `h2o_frame[x].digamma()`
- `h2o_frame[x].exp()`
- `h2o_frame[x].expm1()`
- `h2o_frame[x].floor()`
- `h2o_frame[x].gamma()`
- `h2o_frame[x].lgamma()`
- `h2o_frame[x].log()`
- `h2o_frame[x].log10()`
- `h2o_frame[x].log1p()`
- `h2o_frame[x].log2()`
- `h2o_frame[x].round(digits=0)`
- `h2o_frame[x].scale(center=True, scale=True)`
- `h2o_frame[x].sign()`
- `h2o_frame[x].signif(digits=6)`
- `h2o_frame[x].sin()`
- `h2o_frame[x].sinh()`
- `h2o_frame[x].sinpi()`
- `h2o_frame[x].sqrt()`
- `h2o_frame[x].tan()`
- `h2o_frame[x].tanh()`
- `h2o_frame[x].tanpi()`
- `h2o_frame[x].trigamma()`
- `h2o_frame[x].trunc()`