H_2O.ai

Row & Column Selection

Extracting a single column

- `h2o_frame["x"] # column name x`
- `h2o_frame[2]   # 3rd col`
- `h2o_frame[-2]  # 2nd col from end`
- `h2o_frame[:, -1] # Last column`

Filtering rows

- `h2o_frame[0:5, :]`
- `h2o_frame[h2o_frame["x"] > 1, :]`

Extracting multiple columns

- `h2o_frame[["x","y","z"]]`
- `h2o_frame[[1, 5, 6]]`

Filtering rows for select columns

```
h2o_frame[0:50, [1,2,3]]

med = h2o_frame["a"].median()
h2o_frame[h2o_frame["a"] > med, "z"]
```

Pandas-like convention for slicing and dicing data

(0-based indexes)

Filters & Logical Operations

- Logical Operators
  - `h2o_frame[x].logical_negation()`
  - `h2o_frame[x] & h2o_frame[y]`
  - `h2o_frame[x] | h2o_frame[y]`

- Comparison Operators
  - `h2o_frame[x] {==, !=, <, <=, >=, >} value`
  - `h2o_frame[x] {==, !=, <, <=, >=, >} h2o_frame[y]`

- Logical Data Summaries
  - `h2o_frame[x].all()         # includes NAs`
  - `h2o_frame[x].any()         # includes NAs`
  - `h2o_frame[x].any_na_rm() # disregards NAs`

Row & Column Selection

Pandas-like convention for slicing and dicing data

(0-based indexes)

Extracting a single column

- h2o_frame["x"] # column name x
- h2o_frame[2]   # 3rd col
- h2o_frame[-2] # 2nd col from end
- h2o_frame[:, -1] # Last column

Filtering rows

- h2o_frame[0:5, :]
- h2o_frame[h2o_frame["x"] > 1, :]

Extracting multiple columns

- h2o_frame[["x","y","z"]]
- h2o_frame[[1, 5, 6]]

Filtering rows for select columns

```
h2o_frame[0:50, [1,2,3]]

med = h2o_frame["a"].median()
h2o_frame[h2o_frame["a"] > med, "z"]
```