H_2O.ai

String Munging

- String Matching
  - `h2o_frame[x].countmatches()`
  - `h2o_frame[x].grep()`
  - `h2o_frame[x].match()`

- Substitute pattern match
  - `h2o_frame[x].gsub() # Replace all occurrences`
  - `h2o_frame[x].sub()  # Replace first occurrence`

- String Cleaning
  - `h2o_frame[x].substring()`
  - `h2o_frame[x].strsplit()`
  - `h2o_frame[x].trim()`
  - `h2o_frame[x].lstrip()`
  - `h2o_frame[x].rstrip()`

Joins Between Two H2OFrames

```
h2o_frame.merge(other, all_x=False, all_y=False, by_x=None,
by_y=None, method='auto')
```

Arguments

`h2o_frame` left/self data set in the join.

`other` right/other data set in the join.

`all_x` If True, include all rows from the left/self frame.

`all_y` If True, include all rows from the right/other frame.

`by_x` list of columns in the left/self frame to use as a merge key.

`by_y` list of columns in the right/other frame to use as a merge key.

`method` string representing the merge method, one of auto(default), radix or hash.

String Munging

- String Matching
  - `h2o_frame[x].countmatches()`
  - `h2o_frame[x].grep()`
  - `h2o_frame[x].match()`

- Substitute pattern match
  - `h2o_frame[x].gsub() # Replace all occurrences`
  - `h2o_frame[x].sub()  # Replace first occurrence`

- String Cleaning
  - `h2o_frame[x].substring()`
  - `h2o_frame[x].strsplit()`
  - `h2o_frame[x].trim()`
  - `h2o_frame[x].lstrip()`
  - `h2o_frame[x].rstrip()`