DATA 8

Spring 2018

Lecture 12

Table Examples

Slides created by John DeNero (denero@berkeley.edu) and Ani Adhikari (adhikari@berkeley.edu)

Announcements

Combining Table Methods

Important Table Methods

```
t.select(column, …) or t.drop(column, …)
t.take([row, …]) or t.exclude([row, …])

t.sort(column, descending=False, distinct=False)
t.where(column, are.condition(...))

t.apply(function, column, …)
t.group(column) or t.group(column, function)
t.group([column, …]) or t.group([column, …], function)
t.pivot(cols, rows) or t.pivot(cols, rows, vals, function)

t.join(column, other_table, other_table_column)
```

http://data8.org/datascience/tables.html

Discussion Question

Generate a table with one row per cafe that has the name and discounted price of its cheapest discounted drink

drinks

Drink	Cafe	Price
Milk Tea	Tea One	4
Espresso	Nefeli	2
Coffee	Nefeli	3
Espresso	Abe's	2

discounts

Coupon	Location
5%	Tea One
50%	Nefeli
25%	Tea One

(Demo)

cheapest

Cafe	Drink	Discounted Price
Nefeli	Espresso	1
Tea One	Milk Tea	3

Spring 2016 Midterm, Q2(b)

(b) **(8 pt)** Each row of the `trip` table from lecture describes a single bicycle rental in the San Francisco area. Durations are integers representing times in seconds. The first three rows out of 338343 appear below.

Start	End	Duration
Ferry Building	SF Caltrain	765
San Antonio Shopping Center	Mountain View City Hall	1036
Post at Kearny	2nd at South Park	307

Write a Python expression below each of the following descriptions that computes its value. The first one is provided for you. You *may* use up to two lines and introduce variables.

- The average duration of a rental.

```
total_duration = sum(trip.column(2))
total_duration / trip.num_rows
```

- The name of the station where the most rentals ended (assume no ties).
- The number of stations for which the average duration ending at that station was more than 10 minutes.

(Demo)

Advanced Where

Comparison Operators

The result of a comparison expression is a **`bool`** value

```
x = 2                    y = 3
```

Assignment statements

```
x > 1                    x > y                    y >= 3

x == y                   x != 2                   2 < x < 5
```

Comparison expressions

`t.where(array_of_bool_values)` returns a table with only the rows of **`t`** for which the corresponding **`bool`** is **`True`**.

(Demo)

ZIP Codes

(Demo)