```
import agate
```

Let's add some data.

```
salaries = agate.Table.from_csv('Data/nusalaries1415.csv')
```

```
print(salaries)
```

```
|--------------------+---------------|
|  column_names      | column_types  |
|--------------------+---------------|
|  ID                | Number        |
|  Campus            | Text          |
|  DepartmentNumber  | Date          |
|  DepartmentName    | Text          |
|  CostElement       | Number        |
|  Name              | Text          |
|  Title             | Text          |
|  Position          | Number        |
|  Class             | Number        |
|  Term              | Number        |
|  FTE               | Number        |
|  Salary            | Number        |
|--------------------+---------------|
```

Now we just want UNL, so we need to filter those out.

```
unl = salaries.where(lambda row: row['Campus'] is 'University
of Nebraska-Lincoln')
```

```
print(len(unl.rows))
```

```
0
```

Uh oh. How could this be? The answer is almost always a bad filter condition. In this case, it's not title cased, it should be all caps. And there's a space between the dash on both sides.

```
unl = salaries.where(lambda row: row['Campus'] is 'UNIVERSITY
OF NEBRASKA - LINCOLN')
```

```
print(len(unl.rows))
```

```
0
```

Now, what the hell? That should work, right? Well, not exactly. We need to set our row equal to UNIVERISTY ... and we can't use the regular `=` to do it. We need to use `==` which in Python is actually equal to. The single equal sign is for assigning varables.

```
unl = salaries.where(lambda row: row['Campus'] == 'UNIVERSITY
OF NEBRASKA - LINCOLN')
```

```
print(len(unl.rows))
```

```
6948
```

That's better.