Data Cleaning and Exploration Using Csvkit: Takeaways

Syntax

- Installing CSVkit:

```
sudo pip install csvkit
```

- Consolidating rows from multiple CSV files into one new file:

```
csvstack file1.csv file2.csv file3.csv > final.csv
```

- Adding a new column:

```
csvstack -n origin file1.csv file2.csv file3.csv > final.csv
```

- Specifying a grouping value for each filename:

```
csvstack -g 1,2,3 file2.csv file3.csv > final.csv
```

- Returning standard input in a pretty formatted table representation:

```
head -10 final.csv | csvlook
```

- Displaying all column names along with a unique integer identifier:

```
csvcut -n Combined_hud.csv
```

- Displaying the first five values of a specific column:

```
csvcut -c 1 Combined_hud.csv | head -5
```

- Calculating summary statistics for a column:

```
csvcut -c 2 Combined_hud.csv | csvstat
```

- Calculating the mean value for all columns:

```
csvstat --mean Combined_hud.csv
```

- Finding all rows in a column that match a specific pattern:

```
csvgrep -c 2 -m -9 Combined_hud.csv
```

- Selecting rows that do not match a specific pattern:

```
csvgrep -c 2 -m -9 -i Combined_hud.csv
```

Concepts

- `csvkit` supercharges your workflow by adding command line tools specifically for working with `CSV` files.
- `csvstack` stacks rows from multiple CSV files.
- `csvlook` renders CSV in pretty table format.
- `csvcut` selects specific columns from a CSV file.
- `csvstat` calculates descriptive statistics for some or all columns.
- `csvgrep` filters tabular data using specific criteria.

Resources

- CSVkit documentation
- Working with CSVkit