Exploring Data with Pandas: Fundamentals: Takeaways

Syntax

DATA EXPLORATION METHODS

- Describing a series object:

```
revs = f500["revenues"]
summary_stats = revs.describe()
```

- Unique value counts for a column:

```
country_freqs = f500['country'].value_counts()
```

ASSIGNMENT WITH PANDAS

- Creating a new column:

```
top5_rank_revenue["year_founded"] = 0
```

- Replacing a specific value in a dataframe:

```
f500.loc["Dow Chemical","ceo"] = "Jim Fitterling"
```

BOOLEAN INDEXING IN PANDAS

- Filtering a dataframe down on a specific value in a column:

```
kr_bool = f500["country"] == "South Korea"
top_5_kr = f500[kr_bool].head()
```

- Updating values using Boolean filtering:

```
f500.loc[f500["previous_rank"] == 0, "previous_rank"] = np.nan
prev_rank_after = f500["previous_rank"].value_counts(dropna=False).head()
```

Concepts

- Because pandas is designed to operate like NumPy, a lot of concepts and methods from Numpy are supported. Examples include vectorized operations, methods for calculating summary statistics, and boolean indexing.
- **Method chaining** is a way to combine multiple methods together in a single line. When writing code, you should always assess whether method chaining will make your code harder to read. If it does, it's always preferable to break the code into more than one line.
- Because series and dataframes are two distinct objects, they have their own unique methods. However, there are many times where both series and dataframe objects have a method of the same name that behaves in similar ways.

Resources

- Series Statistics Methods
- Dataframe Statistics Methods
- Boolean Indexing