Module 7: Review, Assignment, Pandas cont.

October 20, 2025

Review

Things We've Covered:

- Strings
- Lists
- Dictionaries
- Tuples
- Built-in Functions
- Functions

Things We've Covered, cont.:

- Boolean expressions
- For loops
- Classes
  - Attributes
  - Methods

Things We've Covered, cont.:

Project Organization

- Setting up our project folder
- Conda installation
- Installing packages with `pip`

Things We've Covered, cont.:

- Numpy
  - ndarray
  - random number generator
- Matplotlib
  - pyplot
  - fig, ax = plt.subplots()

Things We've Covered, cont.:

- Pandas
  - Dataframes
  - Series
  - Groupby

Assignment

Pandas, cont.

Loading Data

You can also use Pandas to load data from files.

```
df = pd.read_csv('~/workspace/mskcc-python/module_07/gapminder.tsv', sep='\t')
```

`sep` refers to the separator. For this file, the columns are delimited using tabs.

Some Useful methods

- `df.head()` : returns a view of your data. By default, it'll show you the first five rows of your data.
- `df.tail()` : similar to head but instead, it will return the last five rows of your data.

These both accept arguments. If you would like to return the first ten rows, you would write:

```
df.head(10)
```

Some Useful Attributes:

- `df.columns` : returns a list of column names
- `df.shape` : returns the number of rows and number of columns of your data.

Subsetting your dataframe: `.iloc`

If you would like to return a subset of your dataset, you can do so by using the `.iloc` indexer

```
df.iloc[:100]
```

This will return the first 100 rows subset of your dataframe.

This uses the dataframe index to select the rowse of interest. The default index will be a `range(len(df))`. There are other possible types of indices but for now, we will just consider this one.

Subsetting your dataframe: `.iloc`, cont.

You can also use it to return a subset of the columns too using their index.

```
df.iloc[:100, :2]
```

`.iloc` uses numeric indexers. This means you cannot use the column names to subset your dataframe with `.iloc`.

Subsetting your dataframe: `.loc`

You can also index your dataframe using the indexer, `.loc`.
This indexer uses the row names and column names for indexing.

By default, the row names are the same as the index.

```
df.loc[:100]
```

Subsetting your dataframe: `.loc`, cont.

This comes into play when you want to include columns in your indexer.

```
df.loc[:100, ['continent', 'country']]
```

Summary Statistics

.mean()
.mean()
.describe()

Let's check out some of these summary statistics for our dataframe.

Q: What do you notice about these values? Is there a way to understand the difference average age for a given country?

Groupby

We use `.groupby` method when we want to do calculations based on a particular group:

```
df.groupby('country')['lifeExp'].mean()
```

By default, it will arrange the countries in alphabetic order. But if we wanted to group based on the continent as well:

```
df.groupby(['continent', 'country'])['lifeExp'].mean()
```

Aggregate

If you would like to calculate diferrent summary statistics for your dataframe, you can do so by using the `.agg()` method:

```
.agg(
    "column_of_interest": ["mean", "median", "skew", ...]
)
```

Let's use our dataframe to compose an aggregation:

```
df.groupby(['continent']).agg({
    'lifeExp': ['mean', 'median', 'max']
})
```

Renaming Columns

- .rename()

```
.rename(columns = {"old_name": "new_name"})
```

Renaming Columns, cont.

```
df.rename(
    columns = {
        'lifeExp': 'life_expectancy'
})
```

To prevent overwriting your dataframe, Pandas creates a copy of your dataframe.

If you want to modify your existing dataframe, you would need to use the argument `inplace` and set it `inplace=True`.

Exercise

Let's do some more exploration of the Gapminder dataset.

- Create a breakdown by year and country of the average life expectancy and the average GDP per capita
- Create another breakdown by year and continent
- Create an average life expectancy and GDP per capita per year
- Create a figure with two subplots with the two calculated averages