Data Transformations and Queries

Using Python

Dr. Austin Brown

Kennesaw State University

Simple Data Transformations/Queries

- **Objective**: Learn to perform basic data transformations such as selecting columns, filtering rows, and creating new columns.
- **Importance**: Essential for cleaning and preparing data for analysis.
- **Key Points**
  - **select() Function**: From the dplyr package, used to select specific columns.
  - **Syntax**: `select(data, column_names)`
  - **Example**: Code snippet showing how to select specific columns.

Simple Data Transformations/Queries

- In many instances, you may need to perform simple data transformations or queries to extract specific information from your dataset.
- This could involve selecting specific columns, filtering rows based on certain conditions, or creating new columns based on existing data.
- In Python, the `pandas` library provides a set of functions that make these operations easy and intuitive.

Selecting Columns in Python with `pandas` library

- In Google Colab, upload the HEART.csv file as we have done previously.
- Go ahead and import the data using `pd.read_csv` as we have done already.
- Now, suppose instead of working with the full dataframe, I want to only focus on a few specific columns:
  - `Chol_Status`
  - `BP_Status`
  - `Weight_Status`
  - `Smoking_Status`

Selecting Columns in Python with `pandas` library

- While there are multiple ways of creating a new dataframe which contains only these four columns, one of the most straightforward ways is to use the indexing method from the `pandas` library.
- The indexing method allows you to choose specific columns from a dataframe and create a new dataframe with only those columns.
- This can be useful when you have a large dataset with many columns, but you are only interested in a subset of them.
- Let's see how this works with the HEART dataset.

Selecting Columns in Python with pandas library

```
## Import pandas library ##
import pandas as pd
## Read in HEART.csv file ##
heart = pd.read_csv("HEART.csv")
## Select specific columns ##
selected_columns = heart[['Chol_Status',
                          'BP_Status',
                          'Weight_Status',
                          'Smoking_Status']]
## Check out first few rows ##
print(selected_columns.head())
```

Selecting Columns in Python with `pandas` library

- As we can see in the above code snippet, we first import the `pandas` library using the `import` statement.
- Next, we read in the HEART.csv file using the `pd.read_csv` function and store it in a dataframe called `heart`.
- We then use the indexing method to select the specific columns we are interested in and store the result in a new dataframe called `selected_columns`.

Filtering Rows in Python with `pandas` library

- Not only can we select columns, but we can also filter rows based on specific conditions.
- For example, in the HEART dataset, we may want to filter out all rows where the `Chol_Status` is `High`.
  - That is, we want to keep only the rows where `Chol_Status` is not `High`.
- To do this, we can use boolean indexing in `pandas`.

Filtering Rows in Python with pandas library

```
## Filter rows where Chol_Status is not High ##
filtered_rows = heart[heart['Chol_Status'] != 'High']
```

Filtering Rows in Python with `pandas` library

- In the code snippet above, we use boolean indexing to filter out rows where the `Chol_Status` is `High`.
- The syntax is `heart[heart['Chol_Status'] != 'High']`.
- This code will return a new dataframe called `filtered_rows` that contains only the rows where `Chol_Status` is not `High`.
- We defined "not equal to" as `!=` in the code snippet.

Creating New Columns in Python with `pandas` library

- Many times, you may need to create new columns based on existing data in your dataset.
- For example, in the HEART dataset, we may want to create a new column called `BMI` that calculates the Body Mass Index for each individual.
- To do this in Python, we can use simple arithmetic operations to create the new column.

Creating New Columns in Python with pandas library

```
## Add BMI Column to heart ##
heart['BMI'] = (heart['Weight'] / (heart['Height'] ** 2)) *
## Check Data Structure ##
print(heart.info())
## Check out first few rows of BMI ##
print(heart['BMI'].head())
```