Basic Data Analysis

Using Python

Dr. Austin Brown

Kennesaw State University

Introduction to Data Analysis

- **Objective**: Learn to generate basic descriptive statistics and visualizations.
- **Importance**: Essential for understanding and summarizing data.

Introduction to Data Analysis

- So far, we have learned how to import, inspect, and perform some basic transformations on data.
- After this is complete, we can now focus on analyzing the data to gain insights and answer questions.
- At a foundational stage, this involves generating descriptive statistics and creating visualizations to summarize and present the data.

Introduction to Data Analysis

- Remember in a previous module on understanding column contents, we learned that we generally have two different types of data:
  - Numeric data: Data that represents quantities or numbers.
  - Categorical data: Data that represents categories or groups.
- This difference is not arbitrary: it has implications for the types of analyses we can perform and methods we have available to us.

Introduction to Data Analysis

- Let's once again use the HEART.csv file to demonstrate some basic data analysis techniques.
- As before, go ahead and upload the file to Google Colab folder and import the data using the `pd.read_csv()` function.
- Now, let's explore some basic analysis of numeric data.

Analyzing Numeric Data

- One of the first steps in analyzing numeric data is to calculate summary statistics.
- In the heart dataset, suppose we want to calculate the sample mean, median, and standard deviation of the `Height` and `Weight` columns.
- While there are several ways to do this, one of the simplest is to use the `describe()` method from the `pandas` library.
- The `describe()` method provides a concise summary of the data, including the mean, median, standard deviation, and other key statistics.

Analyzing Numeric Data

```
## Import pandas library ##
import pandas as pd
## Read in HEART.csv file ##
heart = pd.read_csv("HEART.csv")
## Calculate summary statistics ##
heart[['Height', 'Weight']].describe()
```

Analyzing Numeric Data

- Perhaps the most common method for visualizing a numeric variable is to create a histogram.
- A histogram is a graphical representation of the distribution of a numeric variable.
- The widths of the bars represent the intervals into which the data is grouped, while the heights of the bars represent the frequency of observations in each interval.
- It is a quick, visual tool for understanding common and uncommon values in a dataset.
- In Python, we can use the `hist()` method to create a histogram.

Analyzing Numeric Data

```
## Import matplotlib library for plotting ##
import matplotlib.pyplot as plt
## Create a histogram for Height #
plt.hist(heart['Height'])
plt.title('Histogram of Height')
plt.xlabel('Height (in)')
plt.ylabel('Frequency')
plt.show()
```

Analyzing Categroical Data

- For categorical data, one of the most common ways to summarize the data is to create a frequency table.
- A frequency table is a tabular representation of the number of times each category appears in the data.
- In Python, we can use the `value_counts()` method to generate a frequency table for a categorical variable.

Analyzing Categorical Data

```
## Calculate frequency of Weight_Status variable ##
heart['Weight_Status'].value_counts()
```

Analyzing Categorical Data

- A common way to visualize categorical data is to create a bar plot.
- A bar plot is a graphical representation of the frequency of each category in a dataset, similar to a histogram for numeric data.
- In Python, we can use the `plot(kind='bar')` method to create a bar plot for a categorical variable.

Analyzing Categorical Data

```
## Create a bar plot for Weight_Status variable ##
heart['Weight_Status'].value_counts().plot(kind='bar')
plt.title('Bar plot of Weight Status')
plt.xlabel('Weight Status')
plt.ylabel('Frequency')
plt.show()
```