**Assignment-3 & 4**

**Abhinav Reddy Ayyadapu**

**16354733**

1. **Use any open-source API to access some data in Jason format and then parse the Json data and display it as some kind of dashboard.**

* **API used:** Sportspage Feeds
* It needs to be subscribed for a Basic plan which we can pull 20 per day.

**A screenshot of a computer

Description automatically generated**

* We can run the requests code in our colab which generates the data of the API.

A screenshot of a computer

Description automatically generated



* **Generating data using dash**

**A screenshot of a computer

Description automatically generated**

* **Visualizations**

**There are two different visualizations:**

1. **AP Poll**

A graph of a number of people

Description automatically generated with medium confidence

1. **Coaches Poll**

**A graph of a person standing on a white surface

Description automatically generated with medium confidence**

**2)** **The data file diabetes.csv contains data of 768 patients.**

**Data\_Clean:**

**Given Dataset:**

**A screenshot of a data

Description automatically generated**

**In this data there are 8 attributes (Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age) and 1 response variable (Outcome). The response variable, Outcome, has binary value (1 indicating the outcome is diabetes and 0 means no diabetes).**

**Checking for missing values:**

**A screenshot of a computer code

Description automatically generated**

**Collecting the dataset information:**

**A screenshot of a computer

Description automatically generated**

**Data\_Analysis**

1. **set a seed (to ensure work reproducibility) and take a random sample of 25 observations and find the mean Glucose and highest Glucose values of this sample and compare these statistics with the population statistics of the same variable.**

**A screenshot of a computer

Description automatically generated**

**np.random.seed() :** Setting a seed for reproducibility ensures that the random numbers generated by a computer program are the same every time the program is run. This is particularly useful when you're testing or debugging code, or when you want to ensure that others can replicate your results exactly. This means that every time you generate random numbers using NumPy functions after setting the seed to 42, you will get the same sequence of random numbers.

* **The below shows the Bar Chart and the Pie Chart for the comparison of the results.**

**Bar Chart**

A graph of a comparison of glucose

Description automatically generated

**Pie Chart**

A diagram of glucose

Description automatically generated

1. **Find the 98th percentile of BMI of your sample and the population and compare the results using charts.**

**A close-up of a computer screen

Description automatically generated**

* **The below shows the Bar Chart and Pie Chart for the 98th percentile of BMI of the sample and the population.**

**Bar Chart**

A blue and green rectangles

Description automatically generated

**Pie Chart**

A blue and red circle with black text

Description automatically generated

**c. Using bootstrap (replace= True), create 500 samples (of 150 observation each) from the**

**population and find the average mean, standard deviation and percentile for BloodPressure and compare this with these statistics from the population for the same variable. Again, you should create charts for this comparison. Report on your findings.**

**A screenshot of a computer

Description automatically generated**

* **The below shows Bar graphs and Pie charts for the distribution of blood pressure statistics for a population and for bootstrap samples. The blood pressure statistics include the mean, percentile, and standard deviation.**

**Bar Graph**

A graph of a comparison of blood pressure

Description automatically generated

**Pie Chart**

A close-up of a pie chart

Description automatically generated