Principles of Data Science

Assignment-4

Name: Nikhil Jagadeesh Sriram

Student ID: 16352573

Source:

Diabetes_clean:

```
import pandas as pd
```

```
raw_data=pd.read_csv('diabetes.csv')
raw_data.head(10)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1

```
raw_data.isna()
clean_data=raw_data.dropna(inplace=False)
clean_data
```

Output:

```
raw_data.isna()
clean_data=raw_data.dropna(inplace=False)
clean_data
```

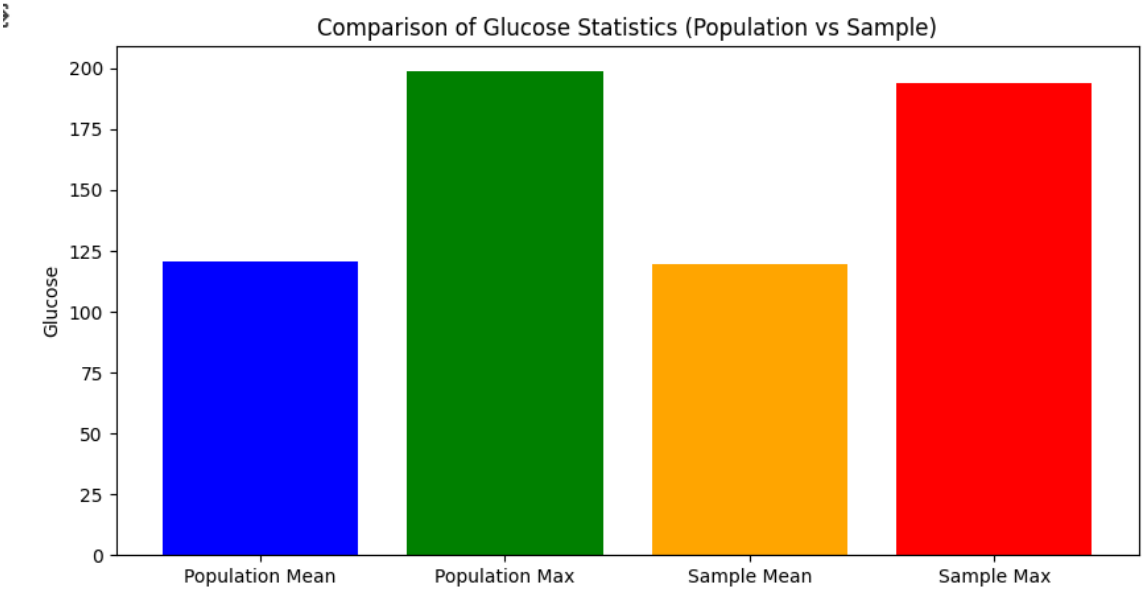
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

Diabetes_analysis:

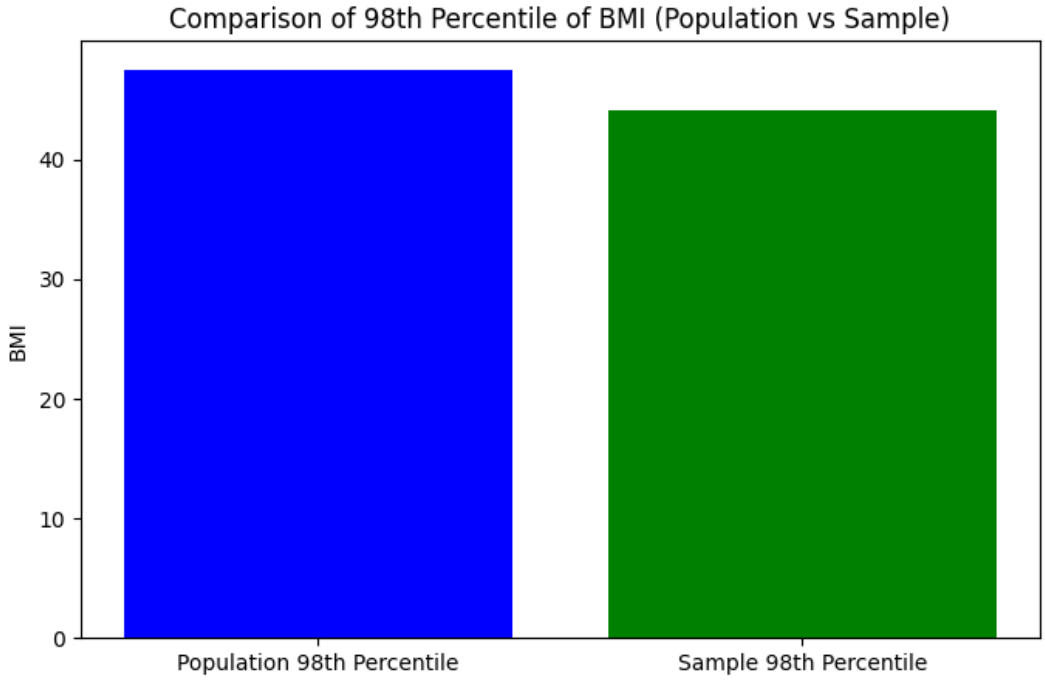
a) set a seed (to ensure work reproducibility) and take a random sample of 25 observations and find the mean Glucose and highest Glucose values of this sample and compare these statistics with the population statistics of the same variable. You should use charts for this comparison.

```
sample_data = data.sample(n=25)

glucose_mean = data['Glucose'].mean()
glucose_max = data['Glucose'].max()
sample_mean_glucose = sample_data['Glucose'].mean()
sample_max_glucose = sample_data['Glucose'].max()

plt.figure(figsize=(10, 5))
plt.bar(['Population Mean', 'Population Max', 'Sample Mean', 'Sample Max'],
        [glucose_mean, glucose_max, sample_mean_glucose, sample_max_glucose],
        color=['blue', 'green', 'orange', 'red'])  # Specify colors here
plt.title('Comparison of Glucose Statistics (Population vs Sample)')
plt.ylabel('Glucose')
plt.show()
```



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

```
[ ]  population_98thbmi = np.percentile(data['BMI'], 98)
     sample_98thbmi = np.percentile(sample_data['BMI'], 98)

     plt.figure(figsize=(8, 5))
     plt.bar(['Population 98th Percentile', 'Sample 98th Percentile'],
             [population_98thbmi, sample_98thbmi],
             color=['blue', 'green'])  # Specify colors here
     plt.title('Comparison of 98th Percentile of BMI (Population vs Sample)')
     plt.ylabel('BMI')
     plt.show()
```



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

```
population_mean = data['BloodPressure'].mean()
population_std = data['BloodPressure'].std()
population_percentile = np.percentile(data['BloodPressure'], 95)

n_bootstrap = 500
bootstrap_size = 150

bootstrap_means = []
bootstrap_stds = []
bootstrap_percentiles = []

for _ in range(n_bootstrap):
    bootstrap_sample = np.random.choice(data['BloodPressure'], size=bootstrap_size, replace=True)
    bootstrap_means.append(np.mean(bootstrap_sample))
    bootstrap_stds.append(np.std(bootstrap_sample))
    bootstrap_percentiles.append(np.percentile(bootstrap_sample, 95))

plt.figure(figsize=(15, 5))

plt.hist(bootstrap_means, bins=30, alpha=0.5, color='orange', label='Bootstrap Sample Means')
plt.axvline(x=population_mean, color='blue', linestyle='--', label='Population Mean')
plt.hist(bootstrap_stds, bins=30, alpha=0.5, color='green', label='Bootstrap Sample Std')
plt.axvline(x=population_std, color='purple', linestyle='--', label='Population Std')
plt.hist(bootstrap_percentiles, bins=30, alpha=0.5, color='yellow', label='Bootstrap Sample 95th Percentiles')
plt.axvline(x=population_percentile, color='red', linestyle='--', label='Population 95th Percentile')

plt.title('Comparison of BloodPressure Statistics')
plt.xlabel('BloodPressure')
plt.ylabel('Frequency')
plt.legend()

plt.tight_layout()
plt.show()
```