A Objective

```
In [ ]:
```

B Snyopsis & Algorithm

```
In [ ]:
```

C Code and Output

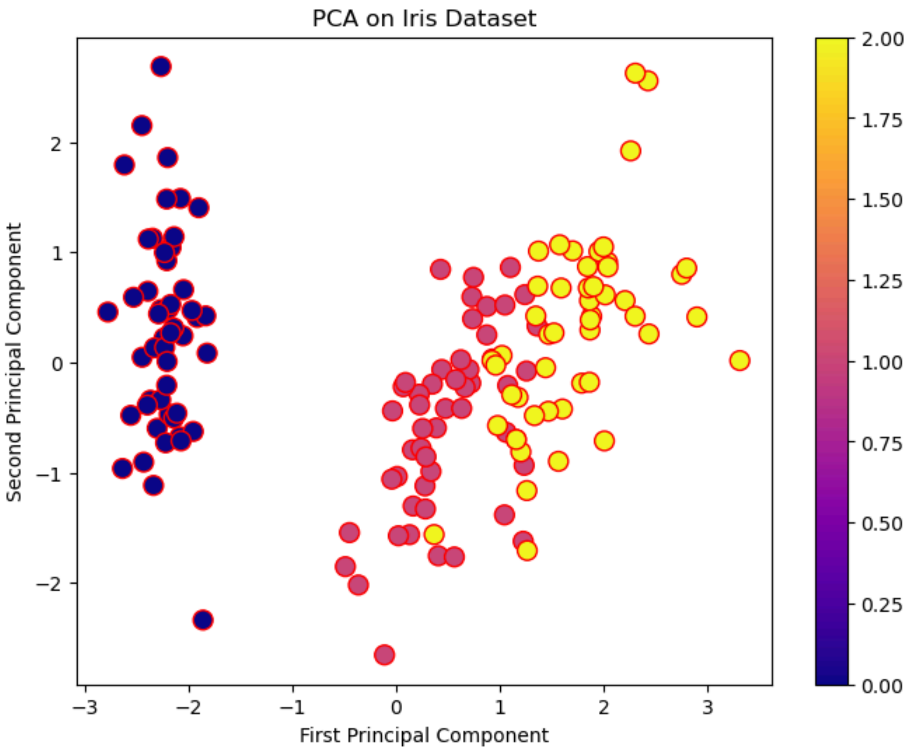
```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        from sklearn.decomposition import PCA
        from sklearn.datasets import load_iris
        from sklearn.preprocessing import StandardScaler
```

```
In [2]: # Load the Iris dataset
        iris = load_iris()
        X = iris.data
        y = iris.target
```

```
In [3]: # Standardize the dataset
        scaler = StandardScaler()
        X_scaled = scaler.fit_transform(X)
```

```
In [4]: # Apply PCA and reduce to 2 components
        pca = PCA(n_components=2)
        X_pca = pca.fit_transform(X_scaled)
```

```
In [9]: # Plot the PCA results
        plt.figure(figsize=(8, 6))
        plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y, cmap='plasma', edgecolor='red', s=100)
        plt.xlabel('First Principal Component')
        plt.ylabel('Second Principal Component')
        plt.title('PCA on Iris Dataset')
        plt.colorbar()
        plt.show()
```



In [6]:

```
# Print explained variance ratio
print("Explained Variance Ratio:", pca.explained_variance_ratio_)
```

```
Explained Variance Ratio: [0.72962445 0.22850762]
```

D Conclusions & Discussion

In [ ]:

```
#Discuss what inferences you have made after the successful completion of the Experiment.
```

E Viva Voce Questions

In [1]:

```
#1 What is Dimensionality Reduction?
```

In [2]:

```
#2 Define PCA?
```

In [3]:

```
#3 Define Eigen Values.
```

In [4]:

```
#4 Define Eigen Vectors.
```

In [5]:

```
#5 Define Orthogonality.
```