

NAME-SANJAY HARITWAL  
REG. NO-21BDS0124

# Assignment-2

+ Code + Text

✓ RAM  
Disk

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data = np.random.randint(1, 101, size=(4, 4))
df=pd.DataFrame(data,columns=['Random value 1','Random value 2','Random value 3','Random value 4'])
column_summary = df['Random value 2'].describe()
print(column_summary)
plt.hist(df['Random value 2'], bins=10)
plt.xlabel('Variable Name')
plt.ylabel('Frequency')
plt.title('Histogram of Variable')
plt.show()
sns.kdeplot(df['Random value 2'], shade=True, label='PDF')
sns.kdeplot(df['Random value 2'], cumulative=True, label='CDF')
plt.title('PDF and CDF of Variable')
plt.legend()
plt.show()
mean_value = df['Random value 2'].mean()
median_value = df['Random value 2'].median()
mode_value = df['Random value 2'].mode().iloc[0]
std_deviation = df['Random value 2'].std()

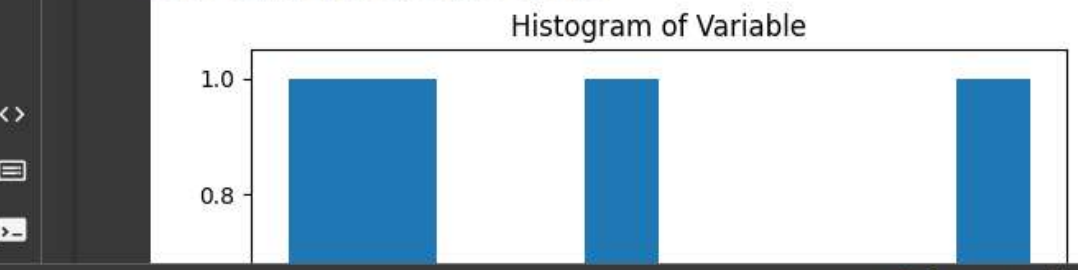
print(f"Mean: {mean_value}")
```

✓ 8s completed at 10:47 PM

```
median_value = df['Random value 2'].median()
mode_value = df['Random value 2'].mode().iloc[0]
std_deviation = df['Random value 2'].std()

print(f"Mean: {median_value}")
print(f"Median: {mode_value}")
print(f"Mode: {mode_value}")
print(f"Standard Deviation: {std_deviation}")
```

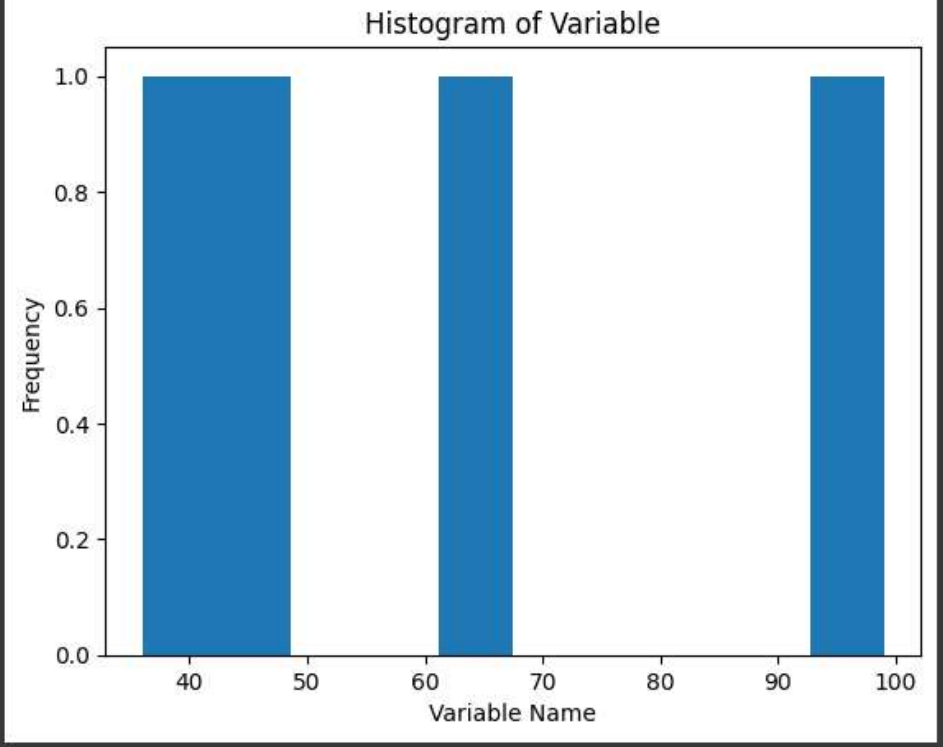
```
count    4.000000
mean     62.000000
std      27.604347
min       36.000000
25%      44.250000
50%      56.500000
75%      74.250000
max       99.000000
Name: Random value 2, dtype: float64
```



8s completed at 10:47 PM



Name: Random value 2, dtype: float64

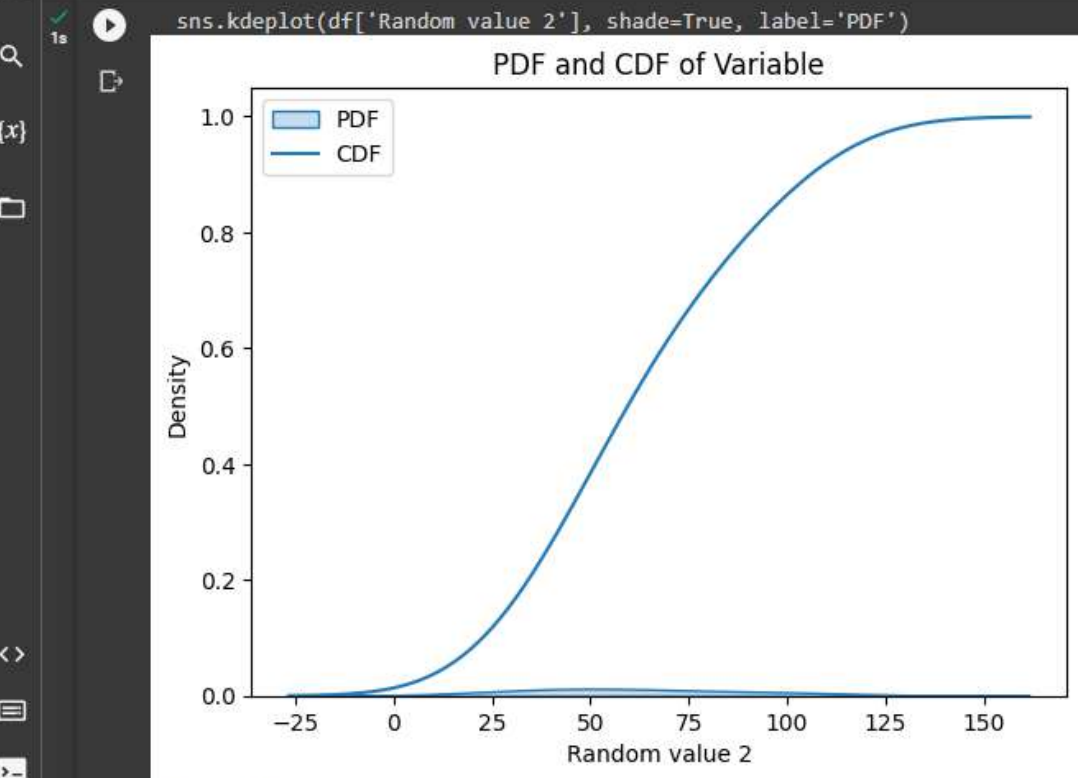


<ipython-input-6-f04b4c52a0ce>:14: FutureWarning:

8s completed at 10:47 PM

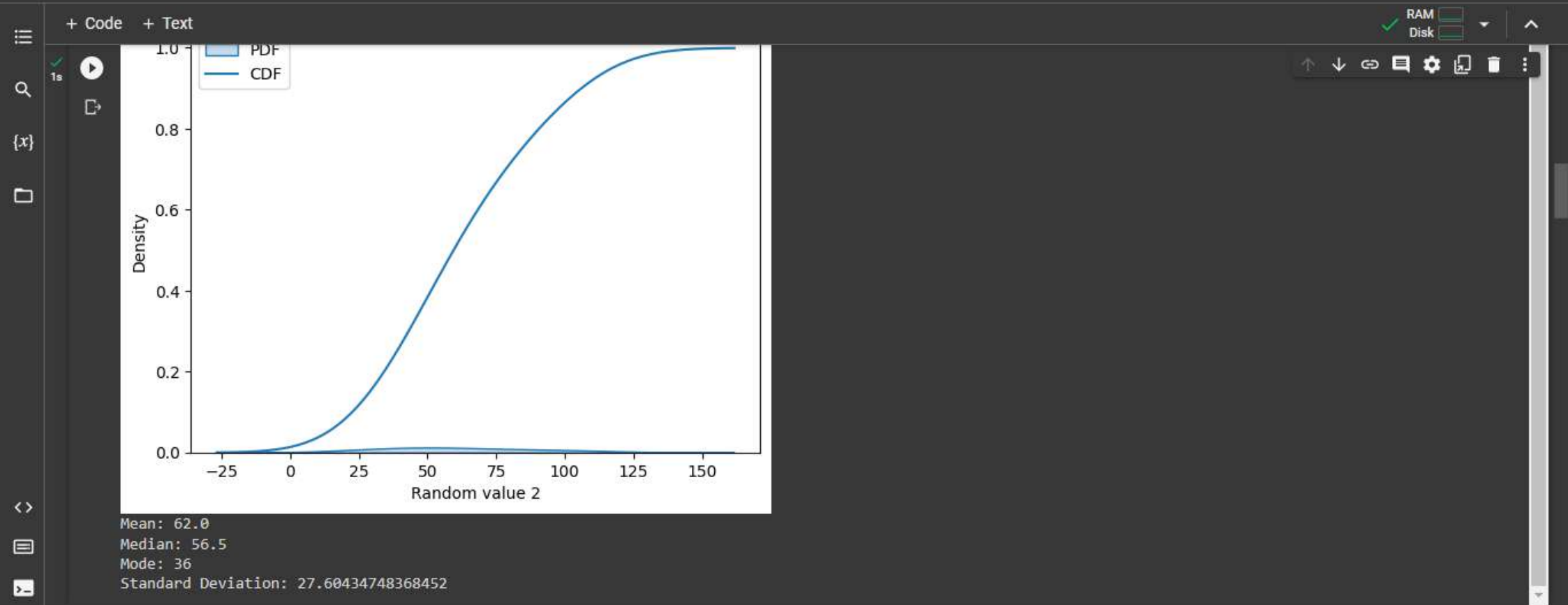
+ Code + Text

RAM Disk



Mean: 62.0

8s completed at 10:47 PM



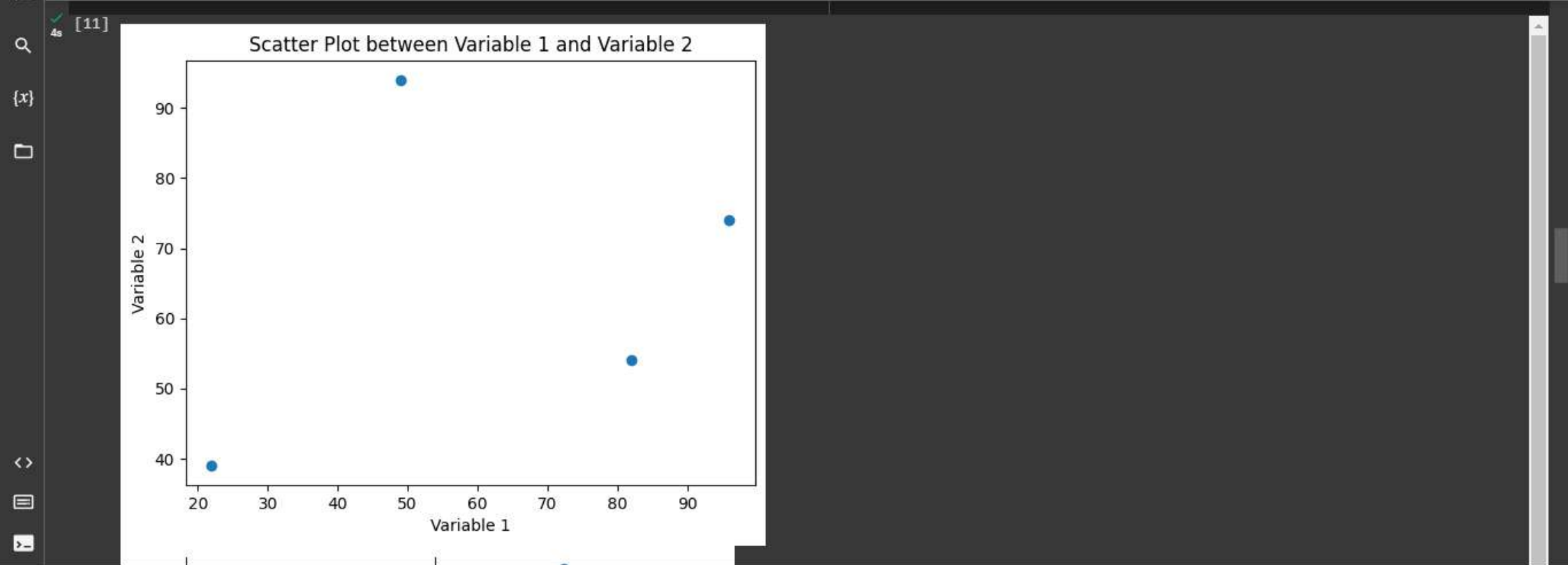


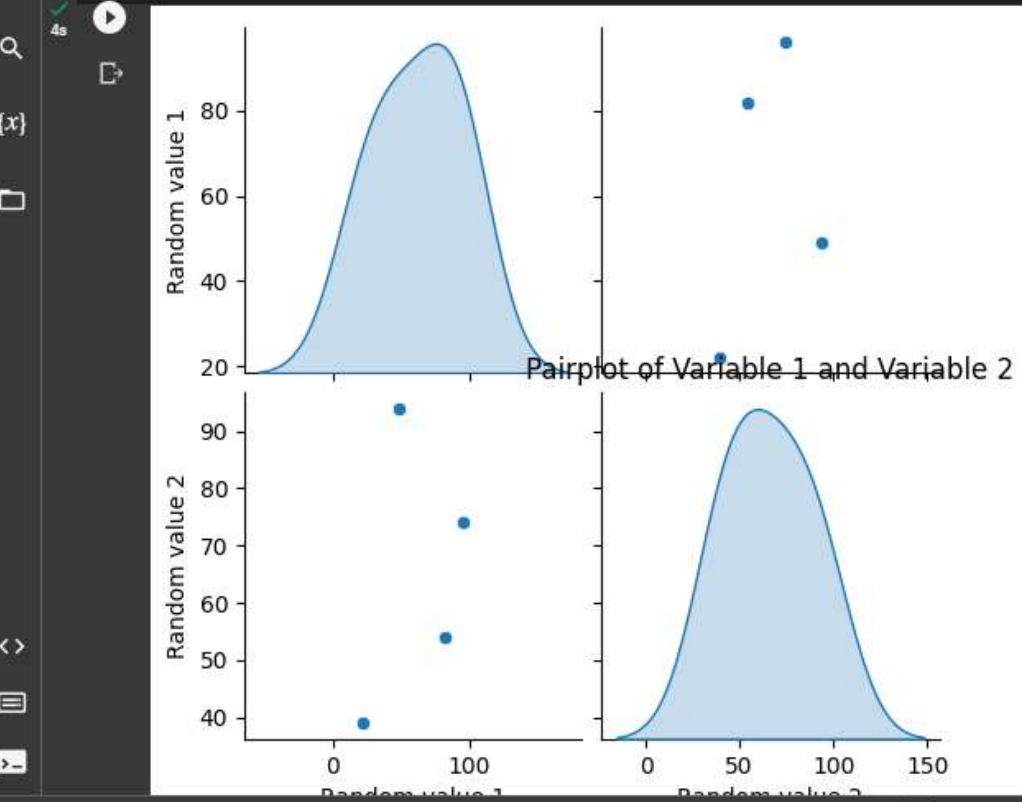
+ Code + Text

RAM Disk

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data = np.random.randint(1, 101, size=(4, 4))
df=pd.DataFrame(data,columns=['Random value 1','Random value 2','Random value 3','Random value 4'])
plt.scatter(df['Random value 1'], df['Random value 2'])
plt.xlabel('Variable 1')
plt.ylabel('Variable 2')
plt.title('Scatter Plot between Variable 1 and Variable 2')
plt.show()
sns.pairplot(df[['Random value 1', 'Random value 2']], diag_kind='kde')
plt.title('Pairplot of Variable 1 and Variable 2')
plt.show()
correlation_coefficient = df['Random value 1'].corr(df['Random value 2'])
print(f'Correlation Coefficient: {correlation_coefficient}')
sns.boxplot(x='Random value 1', y='Random value 2', data=df)
plt.xlabel('Categorical Variable')
plt.ylabel('Continuous Variable')
plt.title('Box Plot of Variable 2 by Category in Variable 1')
plt.show()
correlation_matrix = df[['Random value 1', 'Random value 2', 'Random value 3', 'Random value 4']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix Heatmap')
plt.show()
```

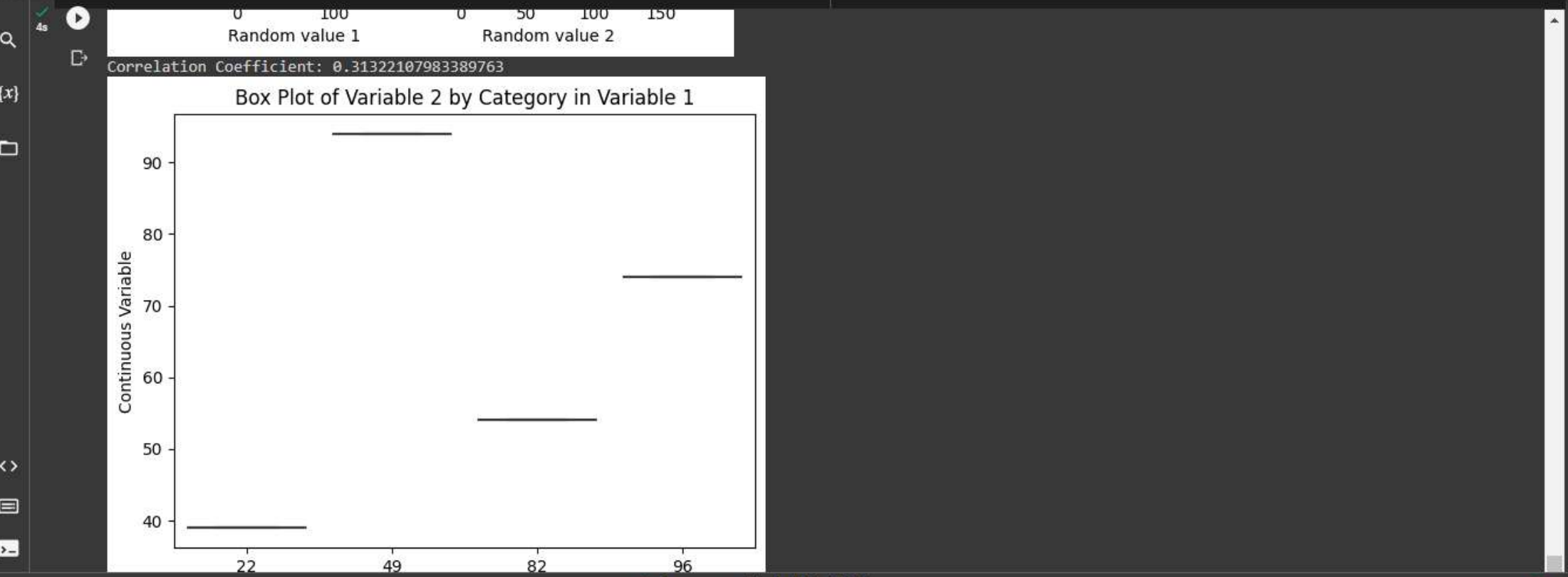
8s completed at 10:47 PM

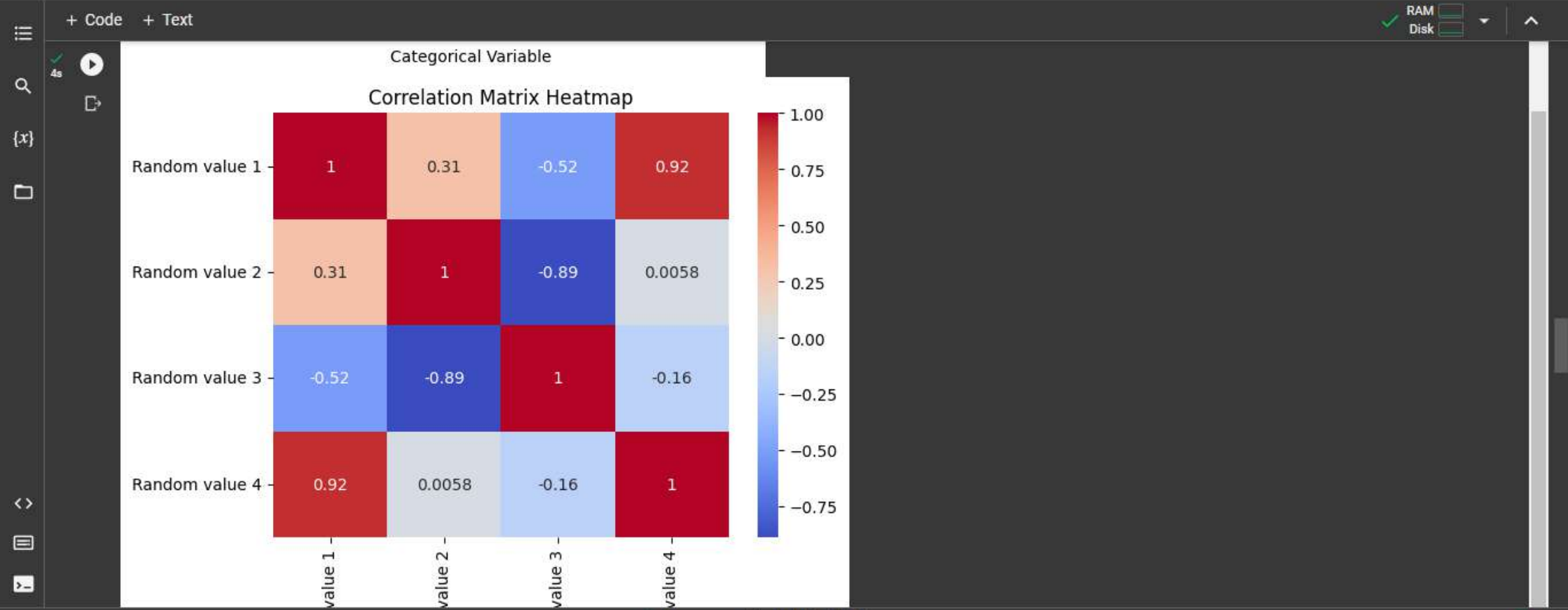


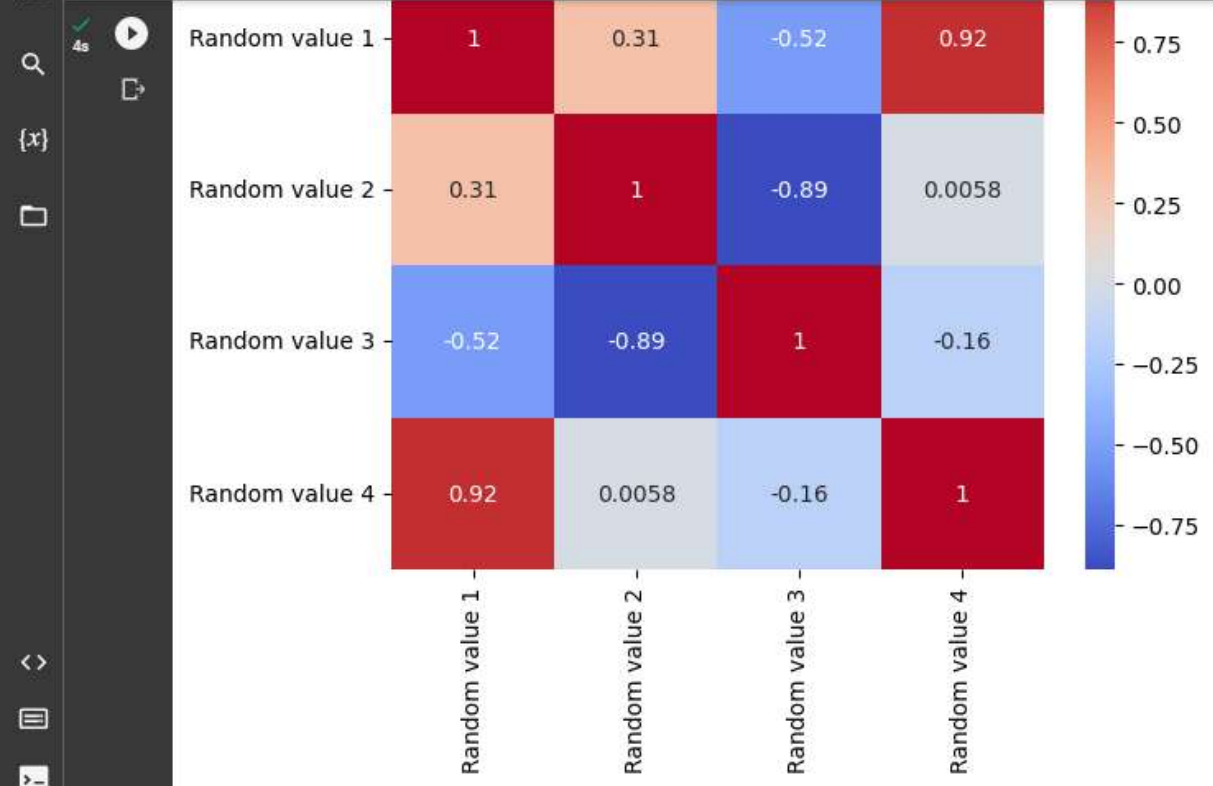


8s completed at 10:47 PM









```
+ Code + Text
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data = np.random.randint(1, 101, size=(4, 4))
df=pd.DataFrame(data,columns=['Random value 1','Random value 2','Random value 3','Random value 4'])
iris = sns.load_dataset('iris')
print(iris.head())
print(iris.describe())
correlation_matrix = iris.corr()
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Matrix")
plt.show()
sns.pairplot(iris, hue='species', markers=["o", "s", "D"])
plt.title("Pairplot of Iris Dataset")
plt.show()
plt.figure(figsize=(12, 6))
plt.subplot(2, 2, 1)
sns.violinplot(x='species', y='sepal_length', data=iris)
plt.subplot(2, 2, 2)
sns.violinplot(x='species', y='sepal_width', data=iris)
plt.subplot(2, 2, 3)
sns.violinplot(x='species', y='petal_length', data=iris)
plt.subplot(2, 2, 4)
sns.violinplot(x='species', y='petal_width', data=iris)
```



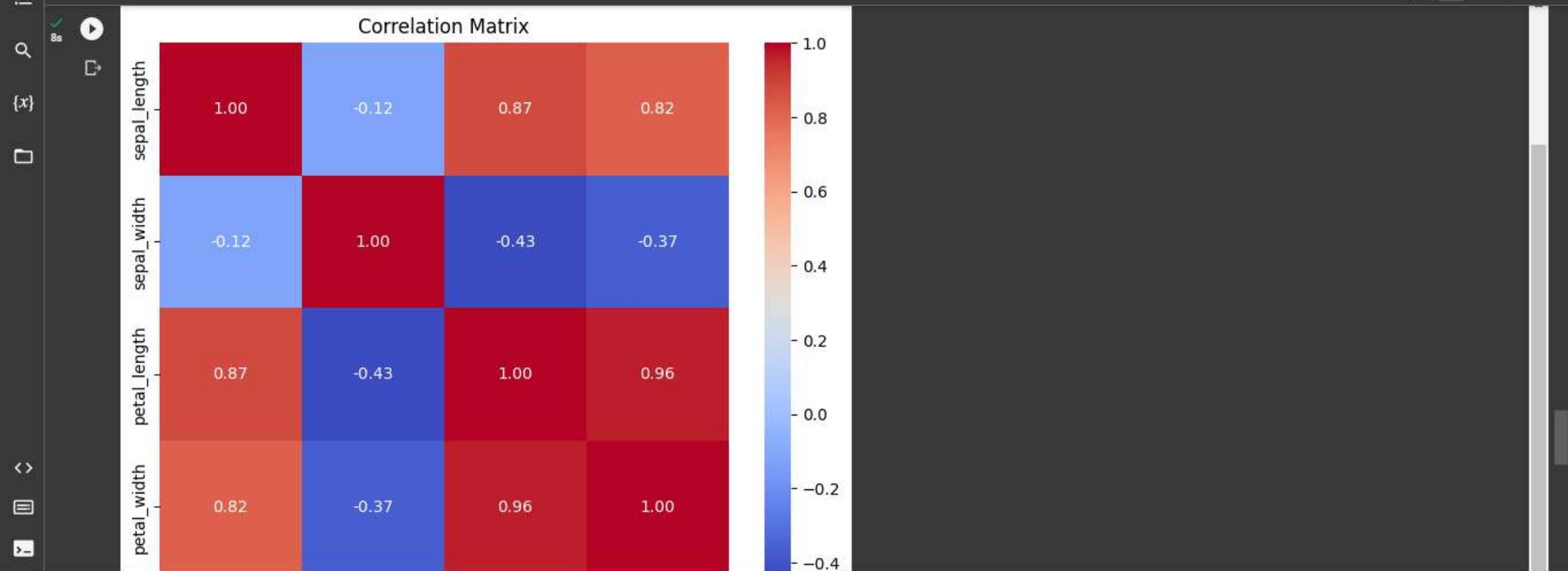
```
plt.subplot(2, 2, 2)
sns.violinplot(x='species', y='sepal_width', data=iris)
plt.subplot(2, 2, 3)
sns.violinplot(x='species', y='petal_length', data=iris)
plt.subplot(2, 2, 4)
sns.violinplot(x='species', y='petal_width', data=iris)
plt.tight_layout()
plt.show()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
count	150.000000	150.000000	150.000000	150.000000	
mean	5.843333	3.057333	3.758000	1.199333	
std	0.828066	0.435866	1.765298	0.762238	
min	4.300000	2.000000	1.000000	0.100000	
25%	5.100000	2.800000	1.600000	0.300000	
50%	5.800000	3.000000	4.350000	1.300000	
75%	6.400000	3.300000	5.100000	1.800000	
max	7.900000	4.400000	6.900000	2.500000	

<ipython-input-13-13270e16935b>:10: FutureWarning: The default value of numeric\_only in DataFrame.corr is deprecated. In a future version, it will default to False  
correlation\_matrix = iris.corr()

Correlation Matrix 8s completed at 10:47 PM





8s completed at 10:47 PM

