In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

In [3]:
```
df = pd.read_csv('Alumni Giving Regression (Edited).csv')
```

In [5]:
```
df.head(20)
```

Out[5]:

	A	B	C	D	E	F
0	24	0.42	0.16	0.59	0.81	0.08
1	19	0.49	0.04	0.37	0.69	0.11
2	18	0.24	0.17	0.66	0.87	0.31
3	8	0.74	0.00	0.81	0.88	0.11
4	8	0.95	0.00	0.86	0.92	0.28
5	20	0.39	0.06	0.35	0.69	0.15
6	20	0.35	0.17	0.60	0.79	0.05
7	18	0.28	0.18	0.58	0.83	0.23
8	18	0.34	0.12	0.57	0.78	0.11
9	14	0.49	0.09	0.71	0.85	0.14
10	21	0.33	0.11	0.27	0.67	0.08
11	14	0.47	0.06	0.91	0.96	0.27
12	19	0.33	0.07	0.61	0.75	0.08
13	21	0.46	0.12	0.78	0.84	0.17
14	16	0.36	0.21	0.67	0.88	0.09
15	19	0.46	0.16	0.67	0.85	0.34
16	22	0.32	0.11	0.54	0.78	0.08
17	16	0.51	0.13	0.76	0.91	0.28
18	18	0.34	0.19	0.64	0.83	0.07
19	17	0.62	0.15	0.91	0.97	0.12

In [15]:
```
# Compute the correlation matrix
correlation_matrix = df.corr()

# Display the correlation matrix
print(correlation_matrix)
```

```
          A         B         C         D         E         F
A  1.000000 -0.691900  0.414978 -0.604574 -0.521985 -0.549244
B -0.691900  1.000000 -0.581516  0.487248  0.376735  0.540427
C  0.414978 -0.581516  1.000000  0.017023  0.055766 -0.175102
D -0.604574  0.487248  0.017023  1.000000  0.934396  0.681660
E -0.521985  0.376735  0.055766  0.934396  1.000000  0.647625
F -0.549244  0.540427 -0.175102  0.681660  0.647625  1.000000
```

In [16]:
```
# Assuming 'E' is the target variable
X = df.drop('E', axis=1)  # Features (excluding the target variable)
y = df['E']  # Target variable
```

In [18]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, randor

# Create and fit the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the testing set
y_pred = model.predict(X_test)
```

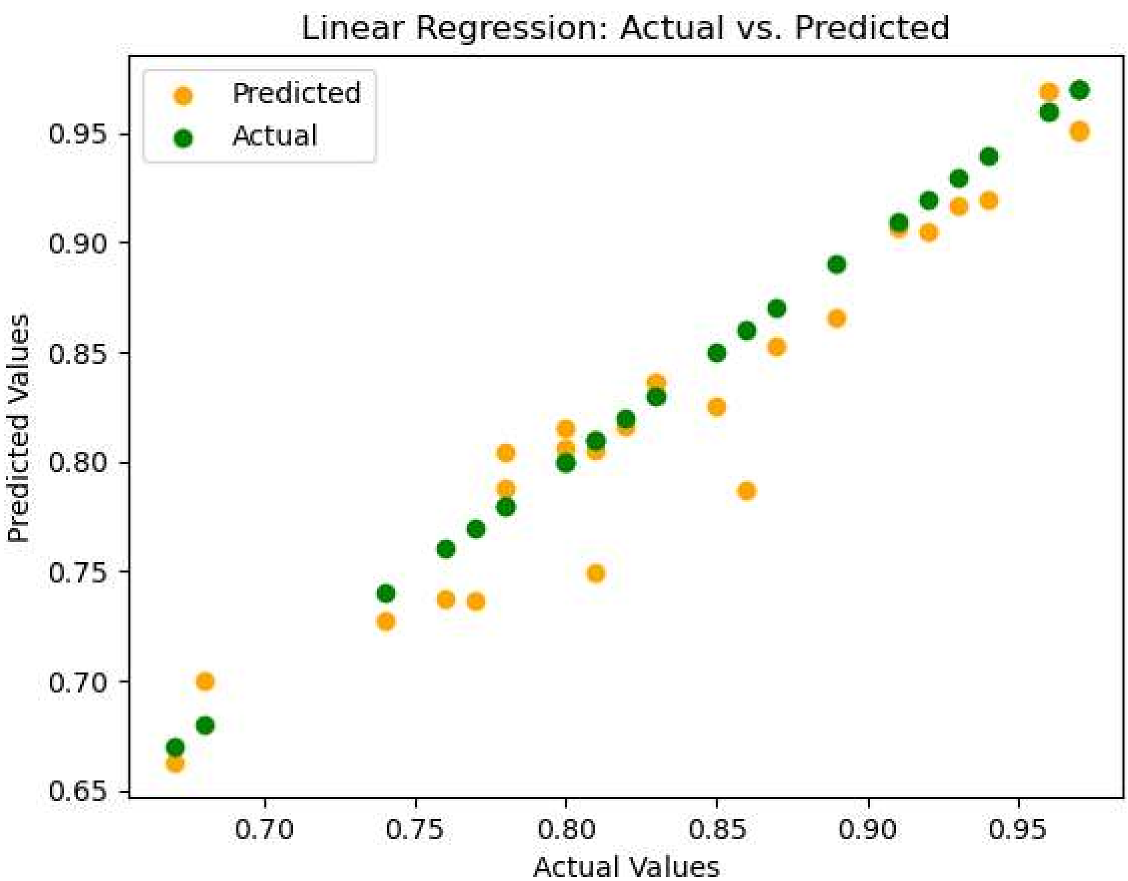
In [19]:
```

# Display the coefficients and intercept
print("Coefficients:", model.coef_)
print("Intercept:", model.intercept_)
```

```
Coefficients: [-2.34100334e-04 -9.20350726e-02 -3.06657997e-02  4.78845060e-0
1
  5.24553522e-02]
Intercept: 0.5678024945072702
```

In [20]:
```

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

```
Mean Squared Error: 0.0006232168181720903
R-squared: 0.9164929012611361
```

In [23]:
```
# Plotting the predicted vs. actual values

plt.scatter(y_test, y_pred, color='orange', label='Predicted')
plt.scatter(y_test, y_test, color='green', label='Actual')
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.title("Linear Regression: Actual vs. Predicted")
plt.legend()
plt.show()
```



In [ ]:

In [ ]:

In [ ]: