In [1]:
```
import numpy as np
# Correlation Analysis with Sample X, Y values
X = np.array([10, 20, 30, 40, 50])
Y= np.array([25, 45, 65, 85,105,])
```

```
c:\users\vamsi2001\appdata\local\programs\python\python39\lib\site-packages\num
py\_distributor_init.py:30: UserWarning: loaded more than 1 DLL from .libs:
c:\users\vamsi2001\appdata\local\programs\python\python39\lib\site-packages\num
py\.libs\libopenblas.EL2C6PLE4ZYW3ECEVIV3OXXGRN2NRFM2.gfortran-win_amd64.dll
c:\users\vamsi2001\appdata\local\programs\python\python39\lib\site-packages\num
py\.libs\libopenblas.XWYDX2IKJW2NMTWSFYNGFUWKQU3LYTCZ.gfortran-win_amd64.dll
  warnings.warn("loaded more than 1 DLL from .libs:"
```

In [2]:
```
# Compute Mean
X_mean = np.mean(X)
Y_mean = np.mean(Y)
print(f"Mean of X: {X_mean}")
print(f"Mean of Y: {Y_mean}")

```

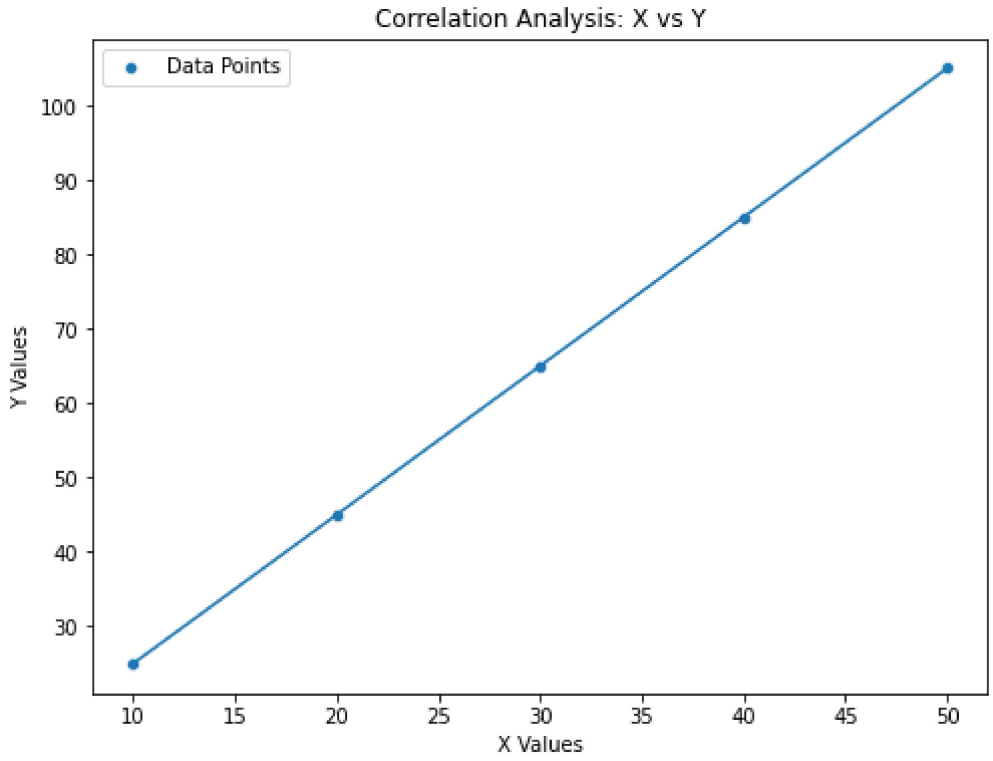
```
Mean of X: 30.0
Mean of Y: 65.0
```

In [3]:
```
# Compute deviations and correlation coefficient
num = np.sum((X - X_mean) * (Y - Y_mean))
denom = np.sqrt(np.sum((X - X_mean) ** 2) * np.sum((Y - Y_mean) ** 2))
r = num / denom
print(f"Correlation Coefficient (r): {r}")
```

```
Correlation Coefficient (r): 1.0
```

In [4]:

```
# Plot Correlation Line
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(8,6))
sns.scatterplot(x=X, y=Y, label='Data Points')
plt.plot(X,Y)
plt.xlabel("X Values")
plt.ylabel("Y Values")
plt.title("Correlation Analysis: X vs Y")
plt.legend()
plt.show()
```



To find correlation on iris Dataset using corr()

In [8]:
```
1 import pandas as pd
2 data=sns.load_dataset('iris')
3 data.head()
4 data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

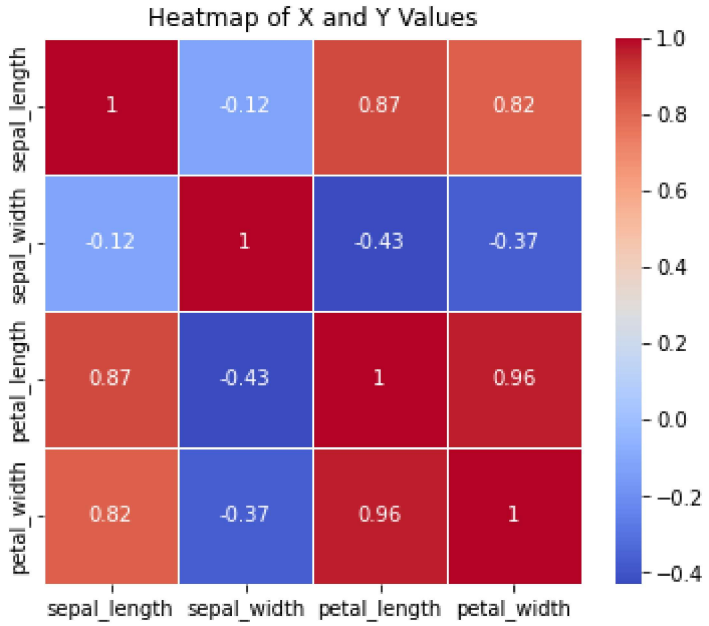
In [9]:
```
1 data.corr()# method='pearson' or 'spearman'
```

Out[9]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

In [10]:

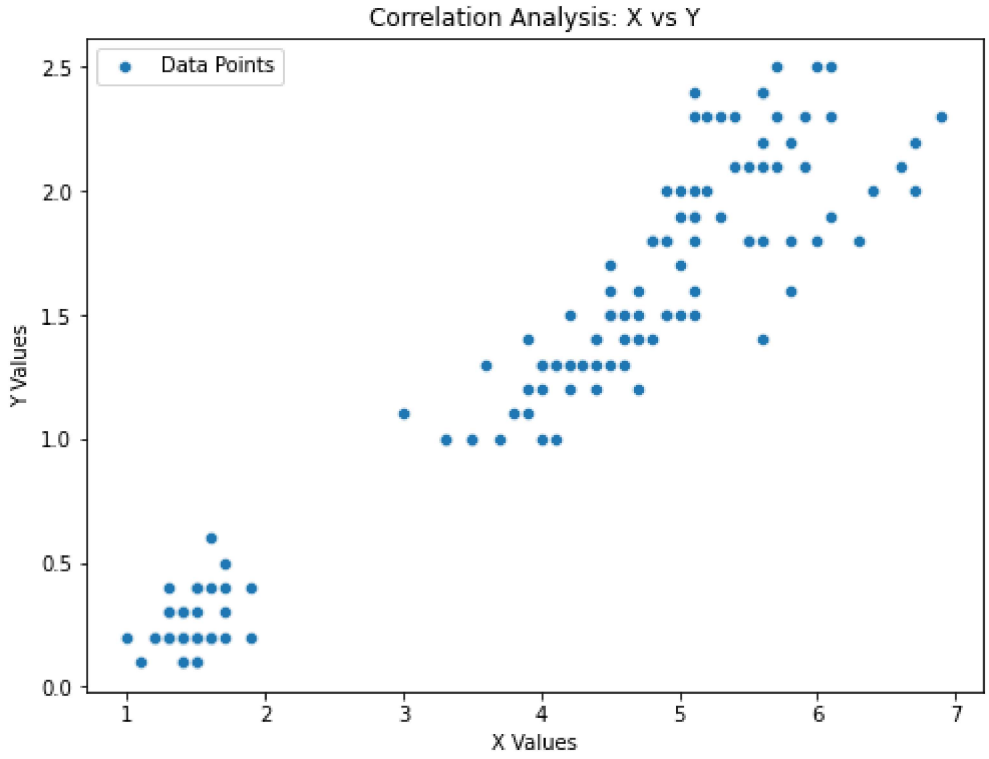
```
import pandas as pd
import matplotlib.pyplot as plt
plt.figure(figsize=(6,5))
sns.heatmap(data.corr() ,annot=True, cmap='coolwarm', linewidths=0.5)
plt.title("Heatmap of X and Y Values")
plt.show()
```



In [11]:

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(8,6))
sns.scatterplot(x=data['petal_length'], y=data['petal_width'], label='Data P
#plt.plot(data['petal_length'],data['petal_width'])
plt.xlabel("X Values")
plt.ylabel("Y Values")
plt.title("Correlation Analysis: X vs Y")
plt.legend()
plt.show()

```



In [13]:
```
import numpy as np
from scipy.stats import pearsonr, spearmanr

# Sample data
x = np.array([1, 2, 3, 4, 5, 6, 7])
y = np.array([3, 5, 7, 9, 11, 13, 15])

# Pearson correlation
pearson_corr, _ = pearsonr(x, y)

# Spearman correlation
spearman_corr, _ = spearmanr(x, y)

print(f"Pearson correlation coefficient: {pearson_corr:.4f}")
print(f"Spearman correlation coefficient: {spearman_corr:.4f}")
print(spearmanr(x, y))
```

```
Pearson correlation coefficient: 1.0000
Spearman correlation coefficient: 1.0000
SpearmanrResult(correlation=1.0, pvalue=0.0)
```

In [14]:
```
import seaborn as sns
df=sns.load_dataset('iris')
df.head()
```

Out[14]:

	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

In [17]:
```
pr,_=pearsonr(df['sepal_length'],df['sepal_width'])
sr,_=spearmanr(df['sepal_length'],df['sepal_width'])
print("pearson coff",pr)
print("spearman coff",sr)
```

```
pearson coff -0.11756978413300206
spearman coff -0.166777658283235
```

In [19]:
```
sr,_=spearmanr(df['petal_length'],df['petal_width'])
pr,_=pearsonr(df['petal_length'],df['petal_width'])
print("pearson coff",pr)
print("spearman coff",sr)
```

```
pearson coff 0.9628654314027963
spearman coff 0.9376668235763412
```

In [ ]:
```
1
```