In [18]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


file_name = "electricity_consumption_sa.csv"


try:
    df = pd.read_csv(file_name, sep=';')
    print("✅ File loaded successfully.")
except FileNotFoundError:
    print(f"❌ Error: File '{file_name}' not found. Check the file name and path.")
    raise

print("\n--- First 5 Rows ---")
print(df.head())
print("\n--- Column Info and Data Types ---")
df.info()
```

```
✅ File loaded successfully.

--- First 5 Rows ---
   Year    Region       Sector  Electricity Consumption (MWh)  \
0  2005   Eastern   Industrial                     28089814.0
1  2005   Western       Others                      2016908.0
2  2005     Total        Total                    153283561.0
3  2006  Southern  Residential                      9247978.0
4  2006   Central       Others                      2431666.0

                     Name Date_Object Periodicity
0    Industrial : Eastern  2005-01-01    Annually
1        Others : Western  2005-01-01    Annually
2           Total : Total  2005-01-01    Annually
3  Residential : Southern  2006-01-01    Annually
4        Others : Central  2006-01-01    Annually

--- Column Info and Data Types ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 630 entries, 0 to 629
Data columns (total 7 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Year                           630 non-null    int64
 1   Region                         630 non-null    object
 2   Sector                         630 non-null    object
 3   Electricity Consumption (MWh)  630 non-null    float64
 4   Name                           630 non-null    object
 5   Date_Object                    630 non-null    object
 6   Periodicity                    630 non-null    object
dtypes: float64(1), int64(1), object(5)
memory usage: 34.6+ KB
```

In [23]:
```python
# تحديد الأعمدة التي سنعمل عليها (هذه الخطوة تمت بالفعل في الخلية 2 السابقة)
DATE_COLUMN = 'Year'
REGION_COLUMN = 'Region'
CONSUMPTION_COLUMN = 'Electricity Consumption (MWh)'

# 1. فلترة البيانات على المنطقة الوسطى ('Central' هي التي تمثل الرياض إقليمياً)
RIYADH_NAME = 'Central'
# لضمان عدم حدوث تحذيرات df من (copy) نستخدم نسخة
```

```
df_riyadh = df[df[REGION_COLUMN] == RIYADH_NAME].copy()

# الطباعة للتحقق من النتيجة:
print(f"✅ تم فلترة البيانات بنجاح على منطقة {RIYADH_NAME}.")
print(f"عدد السجلات التي تخص الرياض: {len(df_riyadh)}")
print("\n(بجميع القطاعات) أول 5 صفوف لبيانات الرياض:")
print(df_riyadh.head())
```

```
✅ تم فلترة البيانات بنجاح على منطقة Central.
عدد السجلات التي تخص الرياض: 126

(بجميع القطاعات) أول 5 صفوف لبيانات الرياض:
    Year   Region        Sector  Electricity Consumption (MWh)  \
4   2006  Central        Others                      2431666.0
9   2007  Central    Industrial                      3748438.0
11  2007  Central        Others                      2604606.0
14  2008  Central  Agricultural                      2355609.0
16  2008  Central         Total                     55983973.0

                     Name Date_Object Periodicity
4         Others : Central  2006-01-01    Annually
9     Industrial : Central  2007-01-01    Annually
11        Others : Central  2007-01-01    Annually
14  Agricultural : Central  2008-01-01    Annually
16         Total : Central  2008-01-01    Annually
```

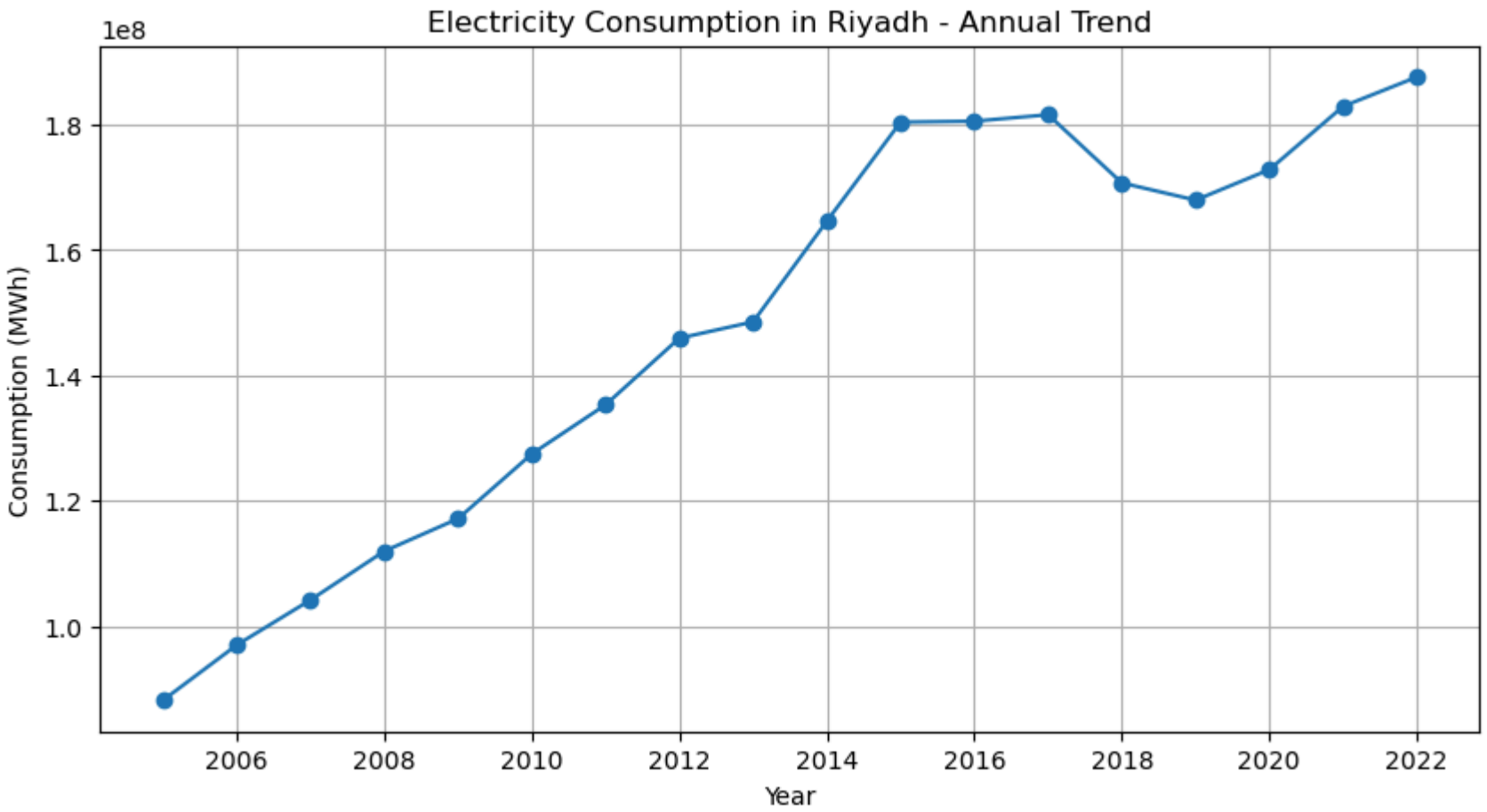
In [35]:
```
# الآن، إطار البيانات الحالي هو df_riyadh والمؤشر هو Year الذي يحتوي على كل القطاعات في الرياض

# 3. التجميع (Aggregation): حساب الاستهلاك الكلي السنوي لكل سنة عن طريق جمع كل القطاعات
CONSUMPTION_COLUMN = 'Electricity Consumption (MWh)'

# نقوم بتجميع البيانات بناءً على مؤشر السنة (Index) ثم جمع عمود الاستهلاك
df_clean = df_riyadh.groupby(level=0)[CONSUMPTION_COLUMN].sum().to_frame()

# 4. التأكد من النظافة
df_clean = df_clean.ffill().dropna()


# الطباعة للتحقق من النتيجة:
print(f"✅ Total annual consumption is collected. Final number of records: {len(df_clean)}")
print("\n data clean:")
```

```
print(df_clean.head(10)) # نطبع 10 صفوف لنرى التتابع الزمني

# للتأكد من انتظام السنوات (Index) طباعة المؤشر
print("\n Available years (indicator):")
print(df_clean.index.year)
```

```
✅ Total annual consumption is collected. Final number of records: 18

 data clean:
            Electricity Consumption (MWh)
Year
2005-01-01                   8.819913e+07
2006-01-01                   9.700675e+07
2007-01-01                   1.041922e+08
2008-01-01                   1.119679e+08
2009-01-01                   1.171662e+08
2010-01-01                   1.274917e+08
2011-01-01                   1.353263e+08
2012-01-01                   1.459458e+08
2013-01-01                   1.485649e+08
2014-01-01                   1.646362e+08

 Available years (indicator):
Index([2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016,
       2017, 2018, 2019, 2020, 2021, 2022],
      dtype='int32', name='Year')
```

In [36]:

```
plt.figure(figsize=(10,5))
plt.plot(df_clean.index, df_clean[CONSUMPTION_COLUMN], marker='o')
plt.title("Electricity Consumption in Riyadh - Annual Trend")
plt.xlabel("Year")
plt.ylabel("Consumption (MWh)")
plt.grid(True)
plt.show()
```



In [59]:
```
print(df_clean.head())
print(df_clean.tail())
print(df_clean.index)
```

```
            Electricity Consumption (MWh)
Year
2005-01-31                     88199134.0
2006-01-31                     97006748.0
2007-01-31                    104192164.0
2008-01-31                    111967946.0
2009-01-31                    117166198.0
            Electricity Consumption (MWh)
Year
2018-01-31                   1.707173e+08
2019-01-31                   1.679766e+08
2020-01-31                   1.728444e+08
2021-01-31                   1.829464e+08
2022-01-31                   1.876239e+08
DatetimeIndex(['2005-01-31', '2006-01-31', '2007-01-31', '2008-01-31',
               '2009-01-31', '2010-01-31', '2011-01-31', '2012-01-31',
               '2013-01-31', '2014-01-31', '2015-01-31', '2016-01-31',
               '2017-01-31', '2018-01-31', '2019-01-31', '2020-01-31',
               '2021-01-31', '2022-01-31'],
              dtype='datetime64[ns]', name='Year', freq='YE-JAN')
```

In [60]:

```
df_clean.index = df_clean.index.year
df_clean.index.name = "Year"
print(df_clean)
print(df_clean.index)
```

```
      Electricity Consumption (MWh)
Year
2005                   8.819913e+07
2006                   9.700675e+07
2007                   1.041922e+08
2008                   1.119679e+08
2009                   1.171662e+08
2010                   1.274917e+08
2011                   1.353263e+08
2012                   1.459458e+08
2013                   1.485649e+08
2014                   1.646362e+08
2015                   1.804233e+08
2016                   1.805722e+08
2017                   1.815990e+08
2018                   1.707173e+08
2019                   1.679766e+08
2020                   1.728444e+08
2021                   1.829464e+08
2022                   1.876239e+08
Index([2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016,
       2017, 2018, 2019, 2020, 2021, 2022],
      dtype='int32', name='Year')
```

In [81]:

```python
import pandas as pd

# قراءة الملفات
weather = pd.read_csv("riyadh_weather_annual.csv")
energy = pd.read_csv("electricity_consumption_sa.csv")

# اختيار فقط المنطقة الوسطى (تقريبًا تمثل الرياض)
energy_central = energy[energy["Region"] == "Central"]

# اختيار الأعمدة المهمة
energy_central = energy_central[["Year", "Electricity Consumption (MWh)"]]

# دمج الطقس مع استهلاك الكهرباء
merged = pd.merge(weather, energy_central, on="Year")

# حفظ الملف الناتج
merged.to_csv("riyadh_weather_energy_merged.csv", index=False)
```

```
print("✅ تم الدمج بنجاح! عدد السنوات:", len(merged))
print(merged.head())
```

```
✅ تم الدمج بنجاح! عدد السنوات: 126
   Year  Temperature (?C)  Humidity (%)  Rainfall (mm)  \
0  2005              26.2            44          100.7
1  2005              26.2            44          100.7
2  2005              26.2            44          100.7
3  2005              26.2            44          100.7
4  2005              26.2            44          100.7

   Electricity Consumption (MWh)
0                      2186793.0
1                     25109641.0
2                      6361952.0
3                      5095729.0
4                      2326012.0
```

In [82]:
```
import pandas as pd

# قراءة الملفات
weather = pd.read_csv("riyadh_weather_annual.csv")
energy = pd.read_csv("electricity_consumption_sa.csv")

# نختار المنطقة الوسطى فقط
energy_central = energy[energy["Region"] == "Central"]

# نجمع استهلاك الكهرباء لكل سنة (بغض النظر عن القطاع)
energy_central_sum = energy_central.groupby("Year", as_index=False)["Electricity Consumption (MWh)"].sum()

# ندمج مع بيانات الطقس
merged = pd.merge(weather, energy_central_sum, on="Year")

# نحفظ الملف الناتج
merged.to_csv("riyadh_weather_energy_merged.csv", index=False)

print("✅ تم الدمج بنجاح بعد التجميع! عدد السنوات:", len(merged))
print(merged.head())
```

```
✅ تم الدمج بنجاح بعد التجميع! عدد السنوات: 18
   Year  Temperature (?C)  Humidity (%)  Rainfall (mm)  \
0  2005              26.2            44          100.7
1  2006              26.5            43          110.2
2  2007              26.8            42          120.3
3  2008              27.0            41          130.4
4  2009              27.2            40          140.5

   Electricity Consumption (MWh)
0                     88199134.0
1                     97006748.0
2                    104192164.0
3                    111967946.0
4                    117166198.0
```

In [83]:

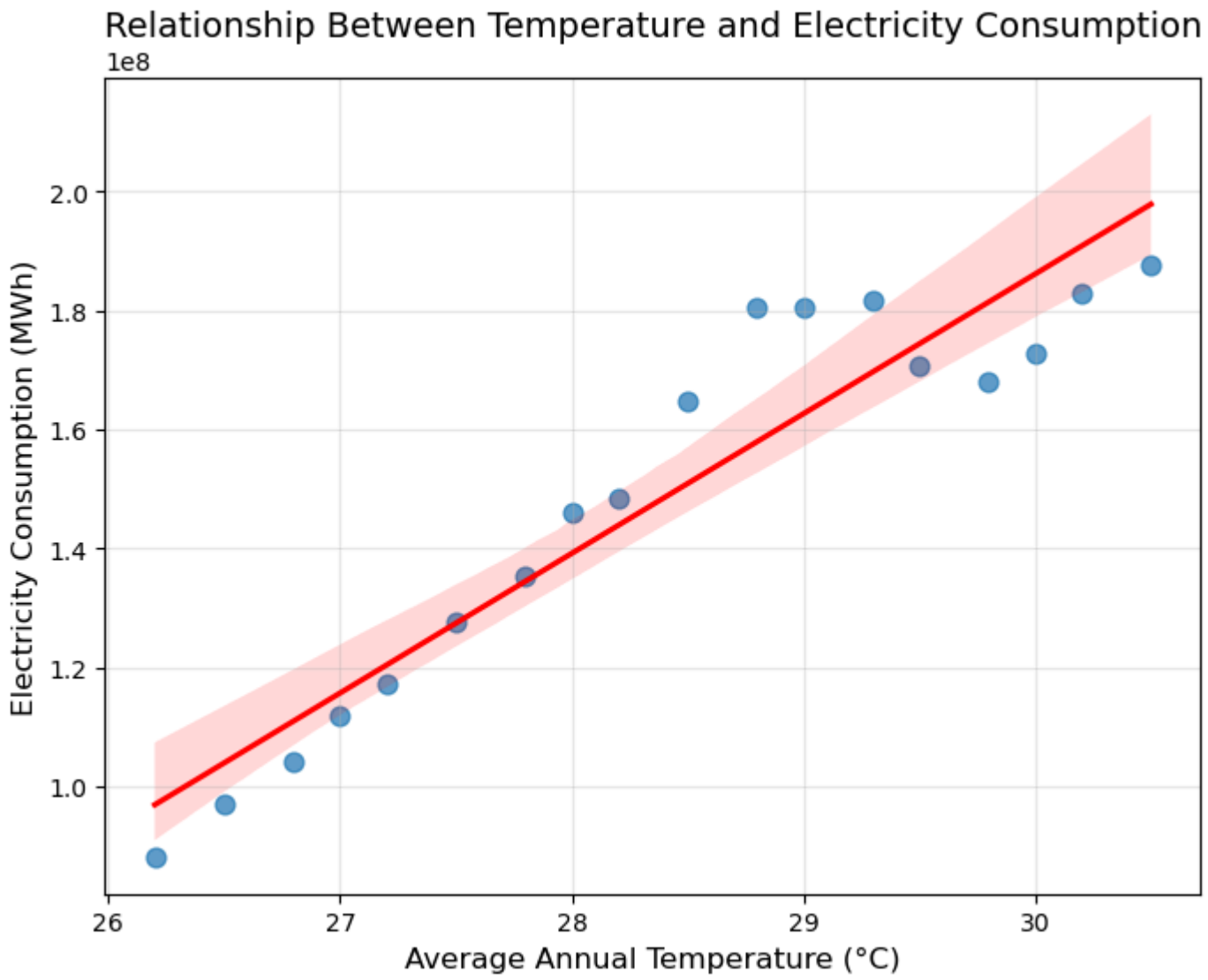
```python
import pandas as pd

# تحميل الملف المدمج
df = pd.read_csv("riyadh_weather_energy_merged.csv")

# عرض أول صفين للتأكد
print("عرض أول صفين من البيانات:")
print(df.head(2))
print("-" * 60)

# معلومات عامة
print("معلومات عامة:")
print(df.info())
print("-" * 60)

# إحصائيات وصفية
print("إحصائيات وصفية:")
print(df.describe())
print("-" * 60)

# أعلى وأقل سنة في استهلاك الكهرباء
max_year = df.loc[df["Electricity Consumption (MWh)"].idxmax(), "Year"]
min_year = df.loc[df["Electricity Consumption (MWh)"].idxmin(), "Year"]

print(f"🔺 أعلى استهلاك كهرباء في سنة: {max_year}")
print(f"🔻 أقل استهلاك كهرباء في سنة: {min_year}")
```

```python
# المتوسطات العامة
mean_temp = df["Temperature (?C)"].mean()
mean_humidity = df["Humidity (%)"].mean()
mean_rain = df["Rainfall (mm)"].mean()
mean_energy = df["Electricity Consumption (MWh)"].mean()

print("\n📊 المتوسطات العامة:")
print(f"درجة الحرارة: {mean_temp:.2f}°C")
print(f"الرطوبة: {mean_humidity:.2f}%")
print(f"الأمطار: {mean_rain:.2f} مم")
print(f"استهلاك الكهرباء: {mean_energy:,.0f} ميجا واط ساعة")
```

```
عرض أول صفين من البيانات:
   Year  Temperature (?C)  Humidity (%)  Rainfall (mm)  \
0  2005              26.2            44          100.7
1  2006              26.5            43          110.2

   Electricity Consumption (MWh)
0                     88199134.0
1                     97006748.0
------------------------------------------------------------
معلومات عامة:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18 entries, 0 to 17
Data columns (total 5 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Year                           18 non-null     int64
 1   Temperature (?C)               18 non-null     float64
 2   Humidity (%)                   18 non-null     int64
 3   Rainfall (mm)                  18 non-null     float64
 4   Electricity Consumption (MWh)  18 non-null     float64
dtypes: float64(3), int64(2)
memory usage: 852.0 bytes
None
------------------------------------------------------------
إحصائيات وصفية:
              Year  Temperature (?C)  Humidity (%)  Rainfall (mm)  \
count    18.000000         18.000000     18.000000      18.000000
mean   2013.500000         28.377778     35.500000     185.483333
std       5.338539          1.339252      5.338539      53.417427
min    2005.000000         26.200000     27.000000     100.700000
25%    2009.250000         27.275000     31.250000     143.025000
50%    2013.500000         28.350000     35.500000     185.450000
75%    2017.750000         29.450000     39.750000     227.875000
max    2022.000000         30.500000     44.000000     270.800000

       Electricity Consumption (MWh)
count                   1.800000e+01
mean                    1.480667e+08
std                     3.322990e+07
min                     8.819913e+07
25%                     1.197476e+08
```

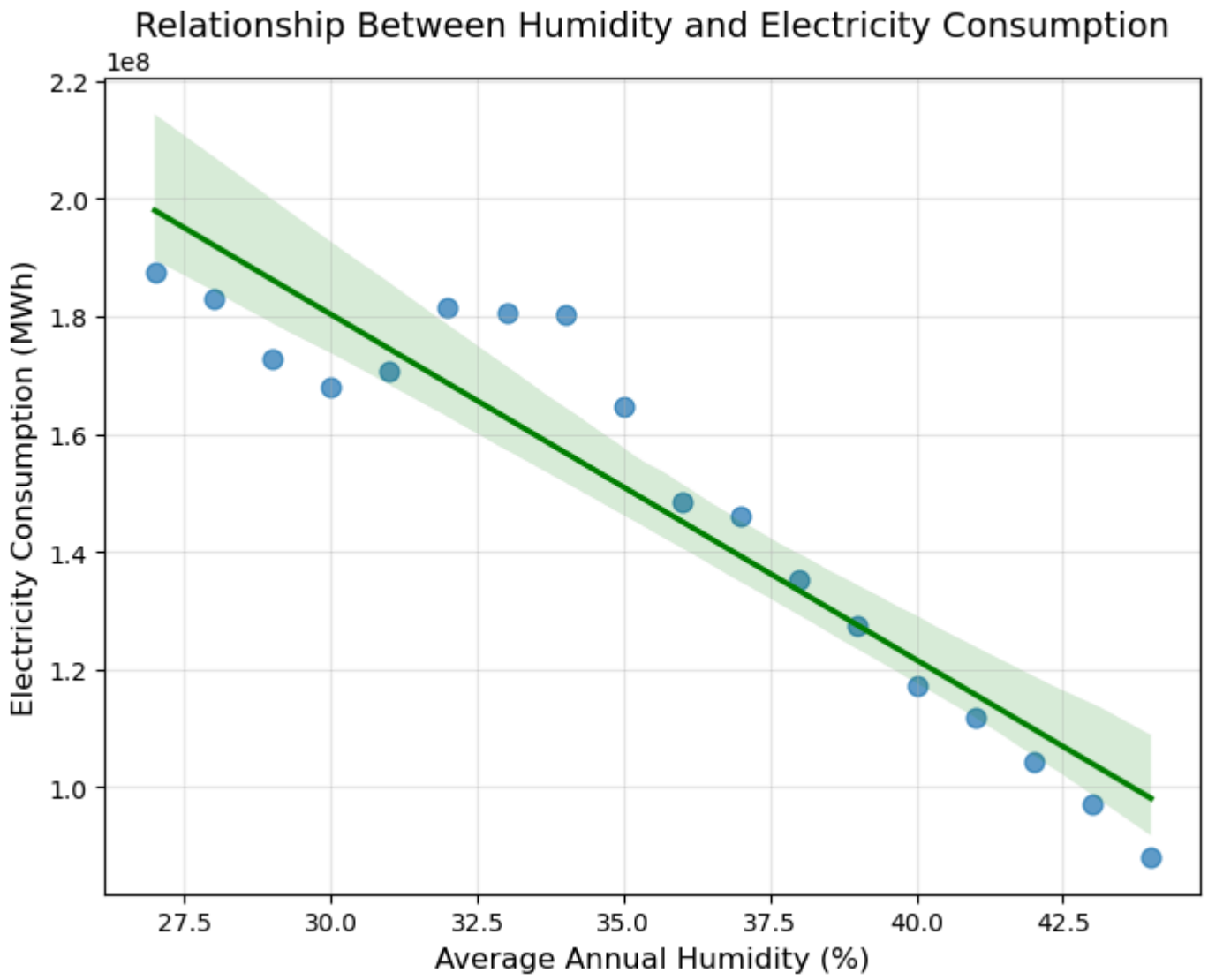
```
50%                    1.566005e+08
75%                    1.785286e+08
max                    1.876239e+08
------------------------------------------------------------
🔺 أعلى استهلاك كهرباء في سنة: 2022
🔻 أقل استهلاك كهرباء في سنة: 2005

📊 المتوسطات العامة:
درجة الحرارة: 28.38°C
الرطوبة: 35.50%
الأمطار: 185.48 مم
استهلاك الكهرباء: 148,066,670 ميجا واط ساعة
```

In [88]:
```python
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8,6))
sns.regplot(
    data=merged,
    x="Temperature (?C)",
    y="Electricity Consumption (MWh)",
    scatter_kws={"s": 60, "alpha": 0.7},
    line_kws={"color": "red"}
)
plt.title("Relationship Between Temperature and Electricity Consumption", fontsize=14)
plt.xlabel("Average Annual Temperature (°C)", fontsize=12)
plt.ylabel("Electricity Consumption (MWh)", fontsize=12)
plt.grid(True, alpha=0.3)
plt.show()
```



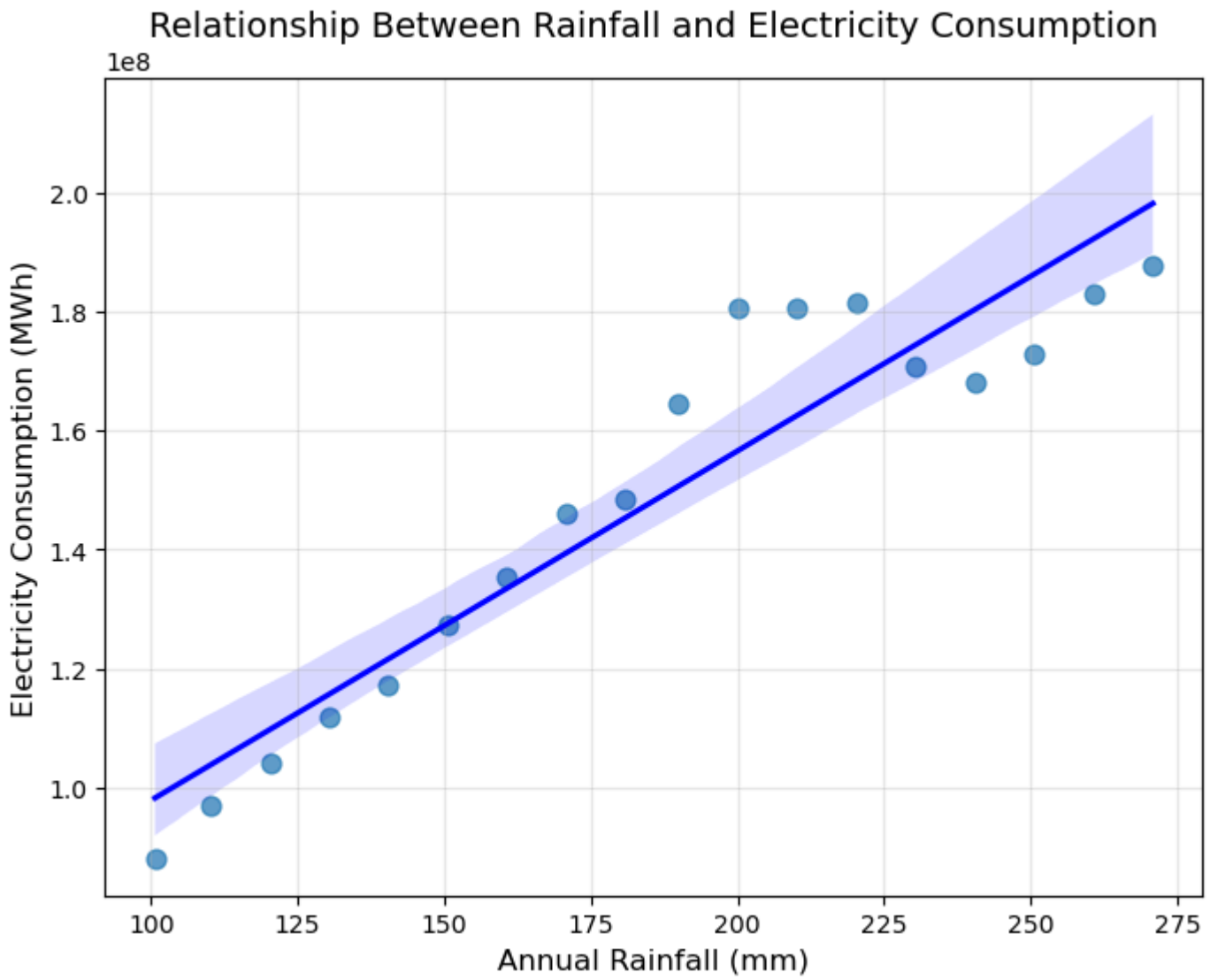
In [89]:
```
plt.figure(figsize=(8,6))
sns.regplot(
    data=merged,
    x="Humidity (%)",
    y="Electricity Consumption (MWh)",
    scatter_kws={"s": 60, "alpha": 0.7},
```

```
    line_kws={"color": "green"}
)
plt.title("Relationship Between Humidity and Electricity Consumption", fontsize=14)
plt.xlabel("Average Annual Humidity (%)", fontsize=12)
plt.ylabel("Electricity Consumption (MWh)", fontsize=12)
plt.grid(True, alpha=0.3)
plt.show()
```



In [90]:
```
plt.figure(figsize=(8,6))
sns.regplot(
    data=merged,
    x="Rainfall (mm)",
    y="Electricity Consumption (MWh)",
    scatter_kws={"s": 60, "alpha": 0.7},
```

```
    line_kws={"color": "blue"}
)
plt.title("Relationship Between Rainfall and Electricity Consumption", fontsize=14)
plt.xlabel("Annual Rainfall (mm)", fontsize=12)
plt.ylabel("Electricity Consumption (MWh)", fontsize=12)
plt.grid(True, alpha=0.3)
plt.show()
```



In [86]:
```
import pandas as pd

# Load the merged dataset
df = pd.read_csv("riyadh_weather_energy_merged.csv")

# Select relevant columns
```

```
corr_columns = ["Temperature (?C)", "Humidity (%)", "Rainfall (mm)", "Electricity Consumption (MWh)"]

# Compute correlation
correlation = df[corr_columns].corr()

print("📊 Correlation Matrix:")
print(correlation)
```

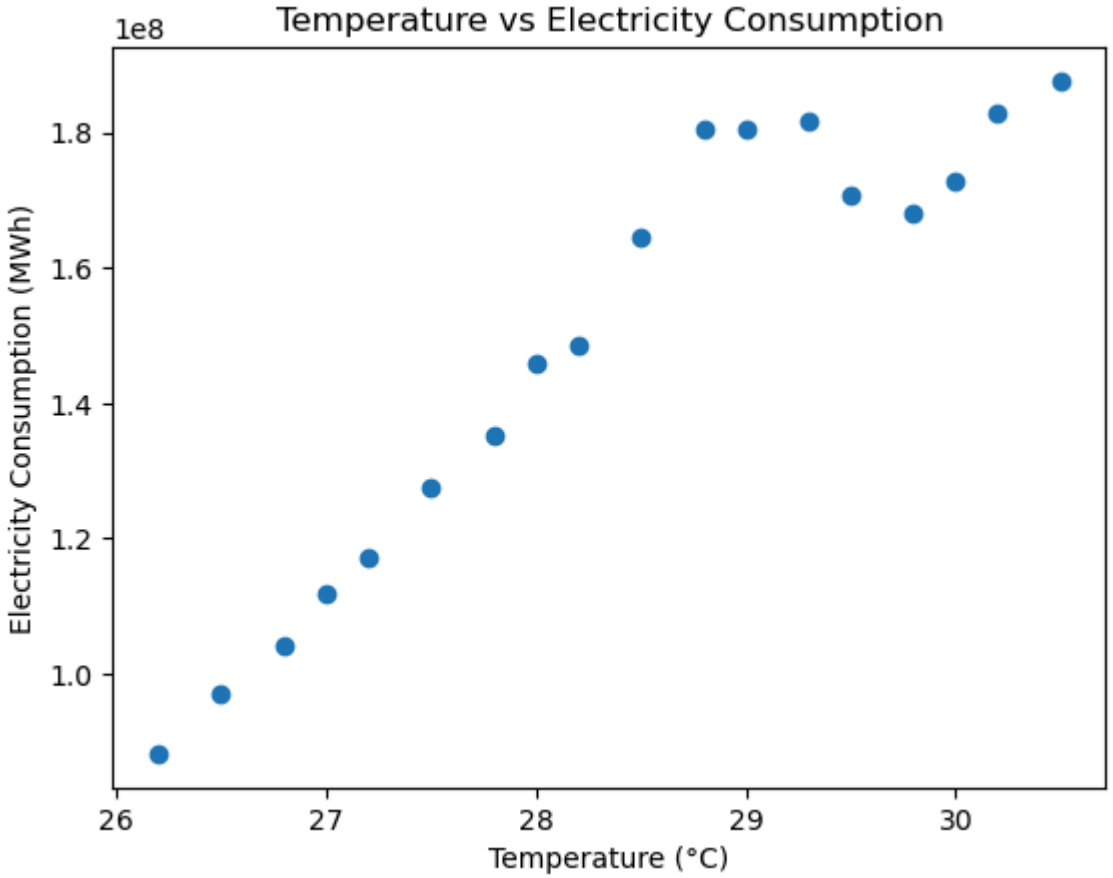
```
📊 Correlation Matrix:
                            Temperature (?C)  Humidity (%)  Rainfall (mm)  \
Temperature (?C)                    1.000000     -0.999638       0.999567
Humidity (%)                       -0.999638      1.000000      -0.999988
Rainfall (mm)                       0.999567     -0.999988       1.000000
Electricity Consumption (MWh)       0.946423     -0.944088       0.943259

                            Electricity Consumption (MWh)
Temperature (?C)                                 0.946423
Humidity (%)                                    -0.944088
Rainfall (mm)                                    0.943259
Electricity Consumption (MWh)                    1.000000
```

In [96]:

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("riyadh_weather_energy_merged.csv")
annual_summary = df.groupby('Year').mean()
print(annual_summary[['Temperature (?C)', 'Electricity Consumption (MWh)']])

# أعلى وأدنى سنة استهلاك
max_year = df.loc[df['Electricity Consumption (MWh)'].idxmax(), 'Year']
min_year = df.loc[df['Electricity Consumption (MWh)'].idxmin(), 'Year']
print(f"أعلى سنة استهلاك: {max_year}, أقل سنة: {min_year}")

# Scatter plot حرارة مقابل استهلاك
plt.scatter(df['Temperature (?C)'], df['Electricity Consumption (MWh)'])
plt.xlabel('Temperature (°C)')
plt.ylabel('Electricity Consumption (MWh)')
plt.title('Temperature vs Electricity Consumption')
plt.show()
```

Year	Temperature (?C)	Electricity Consumption (MWh)
2005	26.2	8.819913e+07
2006	26.5	9.700675e+07
2007	26.8	1.041922e+08
2008	27.0	1.119679e+08
2009	27.2	1.171662e+08
2010	27.5	1.274917e+08
2011	27.8	1.353263e+08
2012	28.0	1.459458e+08
2013	28.2	1.485649e+08
2014	28.5	1.646362e+08
2015	28.8	1.804233e+08
2016	29.0	1.805722e+08
2017	29.3	1.815990e+08
2018	29.5	1.707173e+08
2019	29.8	1.679766e+08
2020	30.0	1.728444e+08
2021	30.2	1.829464e+08
2022	30.5	1.876239e+08

أعلى سنة استهلاك: 2022, أقل سنة: 2005

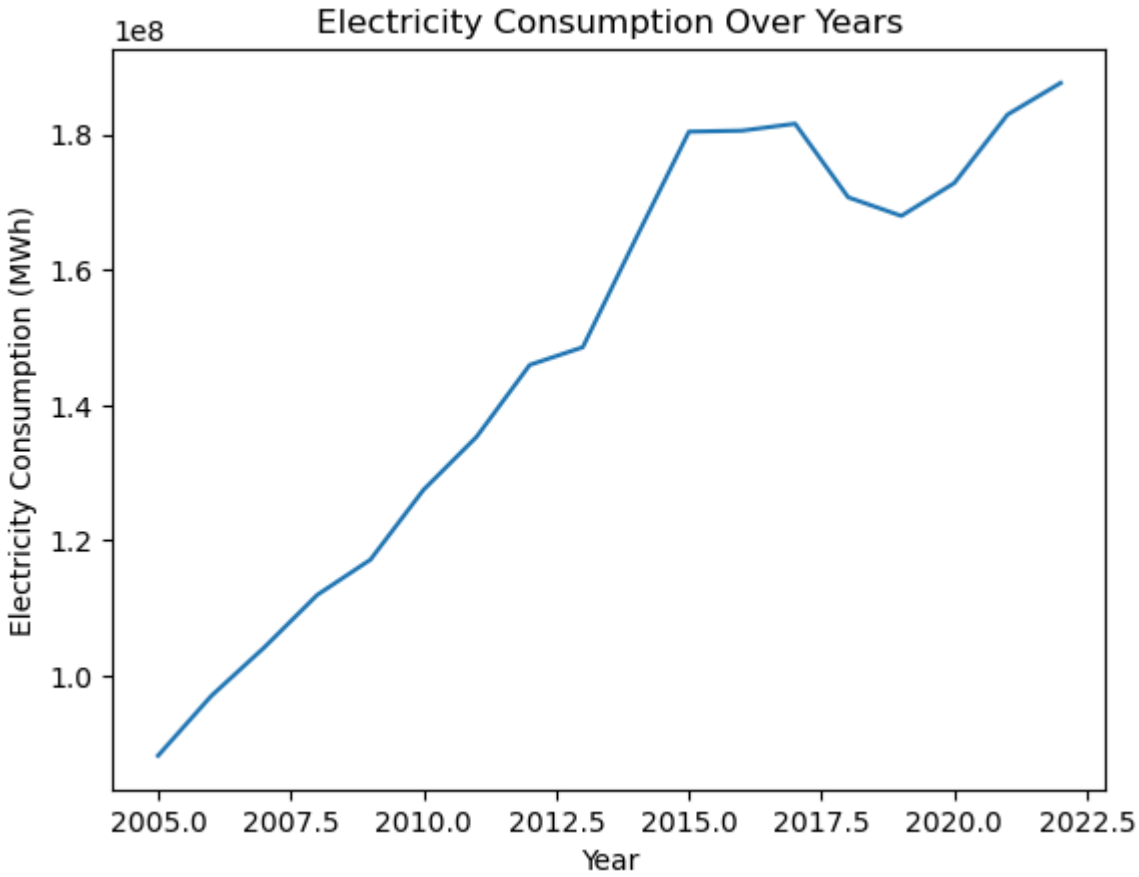


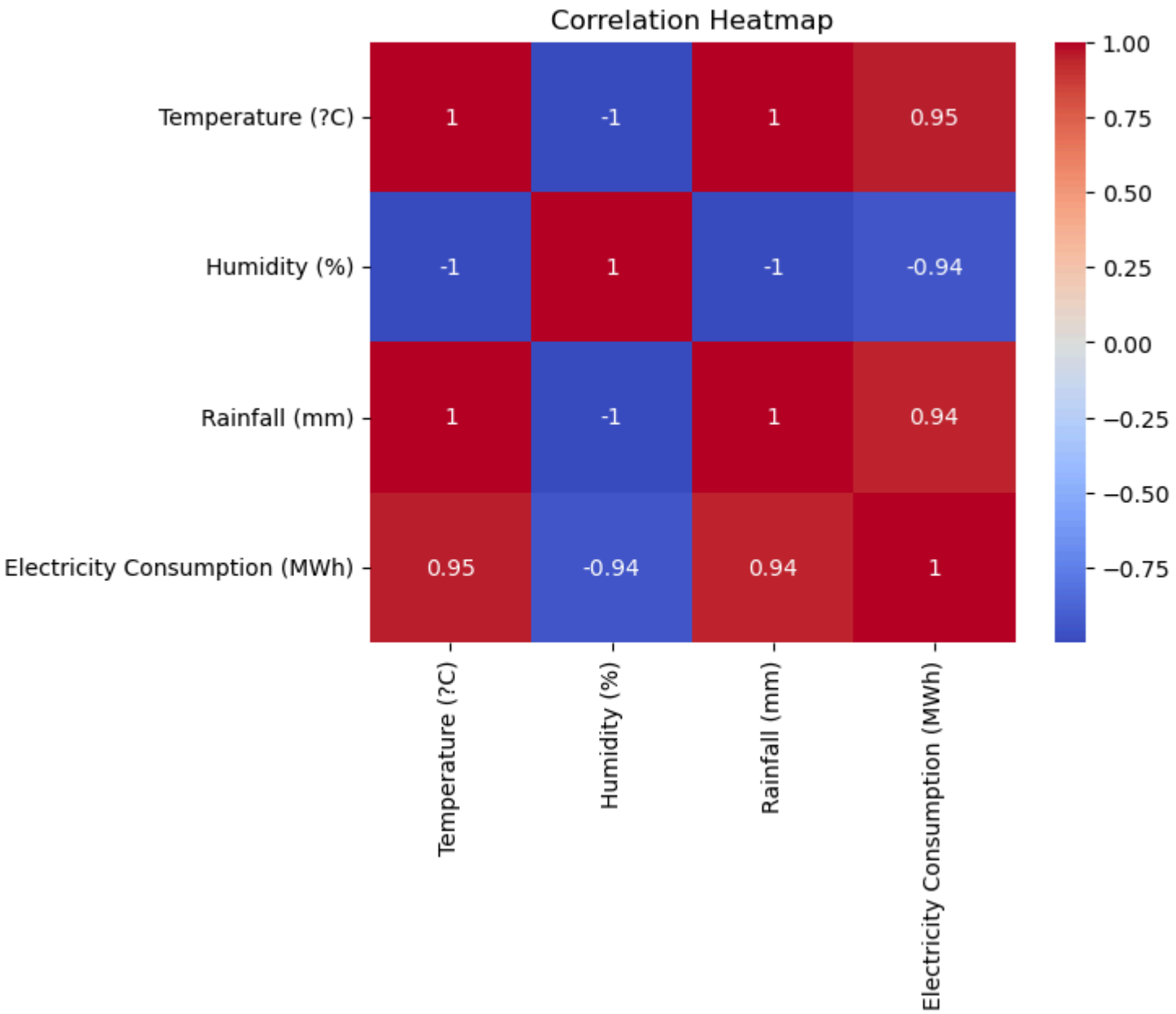
In [97]:
```
corr_columns = ["Temperature (?C)", "Humidity (%)", "Rainfall (mm)", "Electricity Consumption (MWh)"]
correlation = df[corr_columns].corr()
print(correlation['Electricity Consumption (MWh)'])
```

```
Temperature (?C)                0.946423
Humidity (%)                   -0.944088
Rainfall (mm)                   0.943259
Electricity Consumption (MWh)   1.000000
Name: Electricity Consumption (MWh), dtype: float64
```

In [98]:
```
import seaborn as sns

# مخطط زمني
```

```
plt.plot(df['Year'], df['Electricity Consumption (MWh)'])
plt.xlabel('Year')
plt.ylabel('Electricity Consumption (MWh)')
plt.title('Electricity Consumption Over Years')
plt.show()

# Heatmap
sns.heatmap(correlation, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```





In [99]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

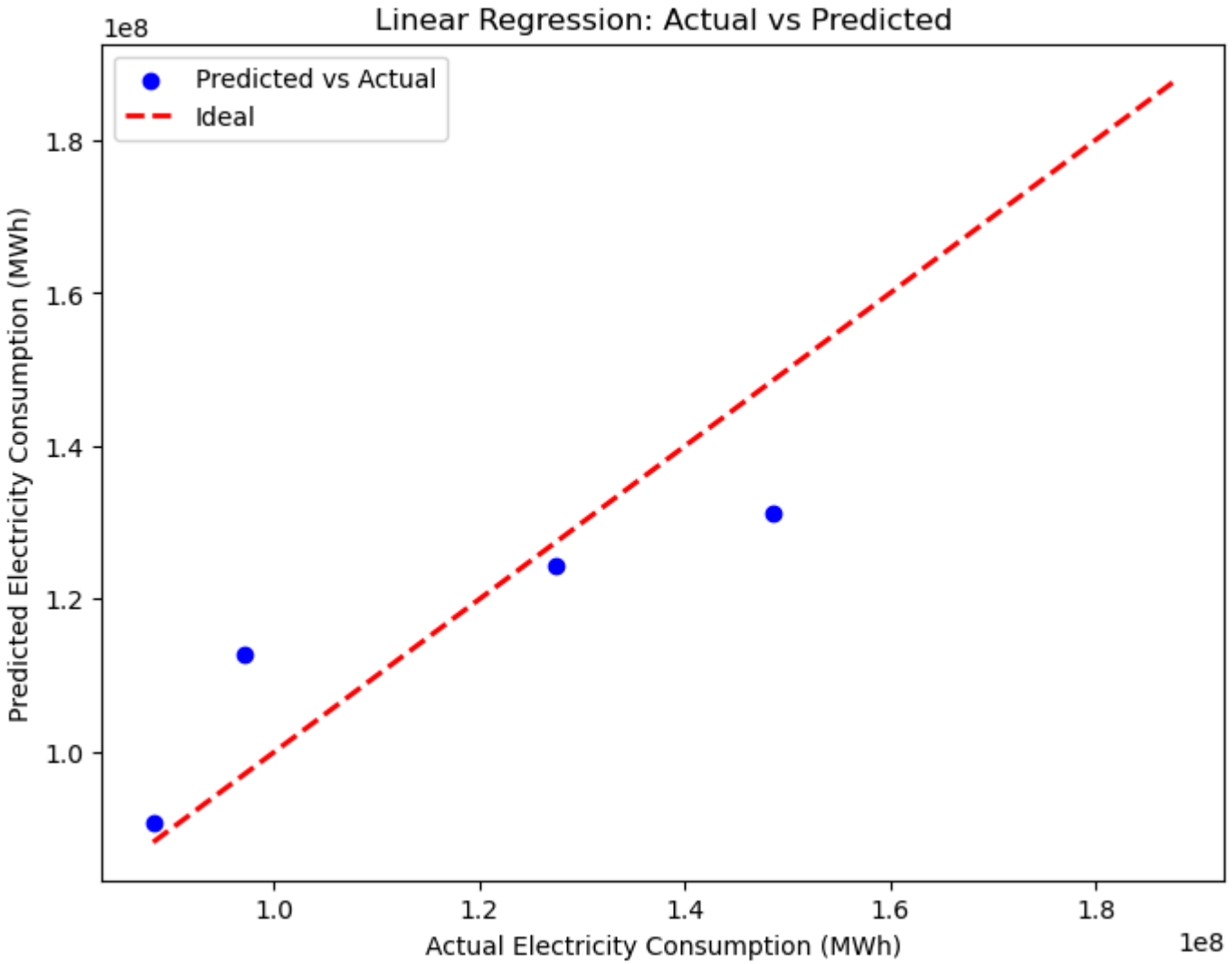
```
from sklearn.metrics import mean_absolute_error, r2_score

X = df[['Temperature (?C)', 'Humidity (%)', 'Rainfall (mm)']]
y = df['Electricity Consumption (MWh)']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print("MAE:", mean_absolute_error(y_test, y_pred))
print("R2:", r2_score(y_test, y_pred))
```

```
MAE: 9689517.34182097
R2: 0.7571582167001402
```

In [101…

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# تحميل البيانات
df = pd.read_csv("riyadh_weather_energy_merged.csv")

# المتغيرات المستقلة والهدف
X = df[['Temperature (?C)', 'Humidity (%)', 'Rainfall (mm)']]
y = df['Electricity Consumption (MWh)']

# تقسيم البيانات
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# تدريب نموذج Linear Regression
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# رسم القيم الحقيقية مقابل التوقعات
plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred, color='blue', label='Predicted vs Actual')
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--', lw=2, label='Ideal')
```

```
plt.xlabel('Actual Electricity Consumption (MWh)')
plt.ylabel('Predicted Electricity Consumption (MWh)')
plt.title('Linear Regression: Actual vs Predicted')
plt.legend()
plt.show()
```



In [113…

```
import pandas as pd
from sklearn.linear_model import LinearRegression
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
import numpy as np

# 1️⃣ تحميل البيانات
df = pd.read_csv("riyadh_weather_energy_merged.csv")

# المتغيرات المستقلة والهدف
X = df[['Temperature (?C)', 'Humidity (%)', 'Rainfall (mm)']]
y = df['Electricity Consumption (MWh)']

# تقسيم البيانات
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# تدريب نموذج Linear Regression
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# ============================
# 2️⃣ حساب مؤشرات الأداء
# ============================
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)

print(f"MAE: {mae:,.2f}")
print(f"RMSE: {rmse:,.2f}")
print(f"R²: {r2:.4f}")
```

```
MAE: 9,689,517.34
RMSE: 11,878,921.68
R²: 0.7572
```

In [115…

```
import numpy as np

# القيم الحقيقية والمتوقعة من النموذج
# (من الكود اللي سويته y_pred و y_test هذي عادة تكون)
accuracy = 100 - np.mean(np.abs((y_test - y_pred) / y_test)) * 100

print(f"📈 نسبة دقة التنبؤ: {accuracy:.2f}%")
```

📈 نسبة دقة التنبؤ: 91.70%

In [12]:
```
import pandas as pd
from sklearn.linear_model import LinearRegression

# 1️⃣ تحميل البيانات
df = pd.read_csv("riyadh_weather_energy_merged.csv")

# ============================
# نموذج الاتجاه فقط (Trend)
# ============================
X_year = df[['Year']]  # DataFrame باسم العمود
y_consumption = df['Electricity Consumption (MWh)']

model_year = LinearRegression()
model_year.fit(X_year, y_consumption)

# ============================
# نموذج باستخدام متوسط الطقس (Average Weather)
# ============================
mean_temp = df['Temperature (?C)'].mean()
mean_humidity = df['Humidity (%)'].mean()
mean_rainfall = df['Rainfall (mm)'].mean()

X_weather = df[['Temperature (?C)', 'Humidity (%)', 'Rainfall (mm)']]
y_weather = df['Electricity Consumption (MWh)']

model_weather = LinearRegression()
model_weather.fit(X_weather, y_weather)

# ============================
# دالة للتنبؤ بأي سنة
# ============================
def predict_year(year):
    # Trend model
    pred_trend = model_year.predict(pd.DataFrame([[year]], columns=['Year']))[0]

    # Average Weather model
    future_weather = pd.DataFrame([{
        'Temperature (?C)': mean_temp,
        'Humidity (%)': mean_humidity,
```

```
        'Rainfall (mm)': mean_rainfall
    }])
    pred_weather = model_weather.predict(future_weather)[0]

    print(f"Predicted Electricity Consumption for {year}:")
    print(f" - Trend only: {pred_trend:,.0f} MWh")
    print(f" - Average Weather: {pred_weather:,.0f} MWh")

    return pred_trend, pred_weather

# ============================
# مثال: توقع سنة 2030
# ============================
pred_trend, pred_weather = predict_year(2026)
```

```
Predicted Electricity Consumption for 2026:
 - Trend only: 221,522,990 MWh
 - Average Weather: 148,066,670 MWh
```

In [17]:

```
results = []

for year in range(2023, 2041):
    pred_trend, pred_weather = predict_year(year)
    results.append({
        "Year": year,
        "Trend only (MWh)": pred_trend,
        "Average Weather (MWh)": pred_weather
    })

forecast_df = pd.DataFrame(results)
forecast_df.to_csv("riyadh_energy_forecast_2023_2040.csv", index=False)

print("✅ تم إنشاء ملف التوقعات من 2023 إلى 2040 بنجاح")
print(forecast_df.head())
print(forecast_df.tail())
```

```
Predicted Electricity Consumption for 2023:
 - Trend only: 203,893,474 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2024:
 - Trend only: 209,769,979 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2025:
 - Trend only: 215,646,485 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2026:
 - Trend only: 221,522,990 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2027:
 - Trend only: 227,399,496 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2028:
 - Trend only: 233,276,002 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2029:
 - Trend only: 239,152,507 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2030:
 - Trend only: 245,029,013 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2031:
 - Trend only: 250,905,519 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2032:
 - Trend only: 256,782,024 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2033:
 - Trend only: 262,658,530 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2034:
 - Trend only: 268,535,035 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2035:
 - Trend only: 274,411,541 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2036:
 - Trend only: 280,288,047 MWh
```

```
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2037:
 - Trend only: 286,164,552 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2038:
 - Trend only: 292,041,058 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2039:
 - Trend only: 297,917,564 MWh
 - Average Weather: 148,066,670 MWh
Predicted Electricity Consumption for 2040:
 - Trend only: 303,794,069 MWh
 - Average Weather: 148,066,670 MWh
✅ تم إنشاء ملف التوقعات من 2023 إلى 2040 بنجاح
   Year  Trend only (MWh)  Average Weather (MWh)
0  2023      2.038935e+08           1.480667e+08
1  2024      2.097700e+08           1.480667e+08
2  2025      2.156465e+08           1.480667e+08
3  2026      2.215230e+08           1.480667e+08
4  2027      2.273995e+08           1.480667e+08
    Year  Trend only (MWh)  Average Weather (MWh)
13  2036      2.802880e+08           1.480667e+08
14  2037      2.861646e+08           1.480667e+08
15  2038      2.920411e+08           1.480667e+08
16  2039      2.979176e+08           1.480667e+08
17  2040      3.037941e+08           1.480667e+08
```

In [20]:
```
combined_df.to_csv("riyadh_energy_combined_actual_forecast.csv", index=False)
```

In [ ]:
```

```