In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import adfuller
import plotly.express as px
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_absolute_error

import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
data = pd.read_csv('Temperatures.csv', encoding='latin-1')
data
```

Out[2]:

	Area Code	Area	Months Code	Months	Element Code	Element	Unit	Y1961	Y1962	Y1963	...	Y2010	Y2011	Y2012	Y2013	Y201
0	2	Afghanistan	7001	January	7271	Temperature change	°C	0.777	0.062	2.744	...	3.601	1.179	-0.583	1.233	1.75
1	2	Afghanistan	7001	January	6078	Standard Deviation	°C	1.950	1.950	1.950	...	1.950	1.950	1.950	1.950	1.95
2	2	Afghanistan	7002	February	7271	Temperature change	°C	-1.743	2.465	3.919	...	1.212	0.321	-3.201	1.494	-3.18
3	2	Afghanistan	7002	February	6078	Standard Deviation	°C	2.597	2.597	2.597	...	2.597	2.597	2.597	2.597	2.59
4	2	Afghanistan	7003	March	7271	Temperature change	°C	0.516	1.336	0.403	...	3.390	0.748	-0.527	2.246	-0.07
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
9651	5873	OECD	7018	Jun□Jul□Aug	6078	Standard Deviation	°C	0.247	0.247	0.247	...	0.247	0.247	0.247	0.247	0.24
9652	5873	OECD	7019	Sep□Oct□Nov	7271	Temperature change	°C	0.036	0.461	0.665	...	0.958	1.106	0.885	1.041	0.99
9653	5873	OECD	7019	Sep□Oct□Nov	6078	Standard Deviation	°C	0.378	0.378	0.378	...	0.378	0.378	0.378	0.378	0.37
9654	5873	OECD	7020	Meteorological year	7271	Temperature change	°C	0.165	-0.009	0.134	...	1.246	0.805	1.274	0.991	0.81
9655	5873	OECD	7020	Meteorological year	6078	Standard Deviation	°C	0.260	0.260	0.260	...	0.260	0.260	0.260	0.260	0.26

9656 rows × 66 columns

In [3]:

```
year_columns = [col for col in data.columns if col.startswith('Y')]
temperatures = data[year_columns].mean()

data = temperatures.reset_index()
data.columns = ['Year', 'Average Temperature']
data['Year'] = data['Year'].str[1:]
```

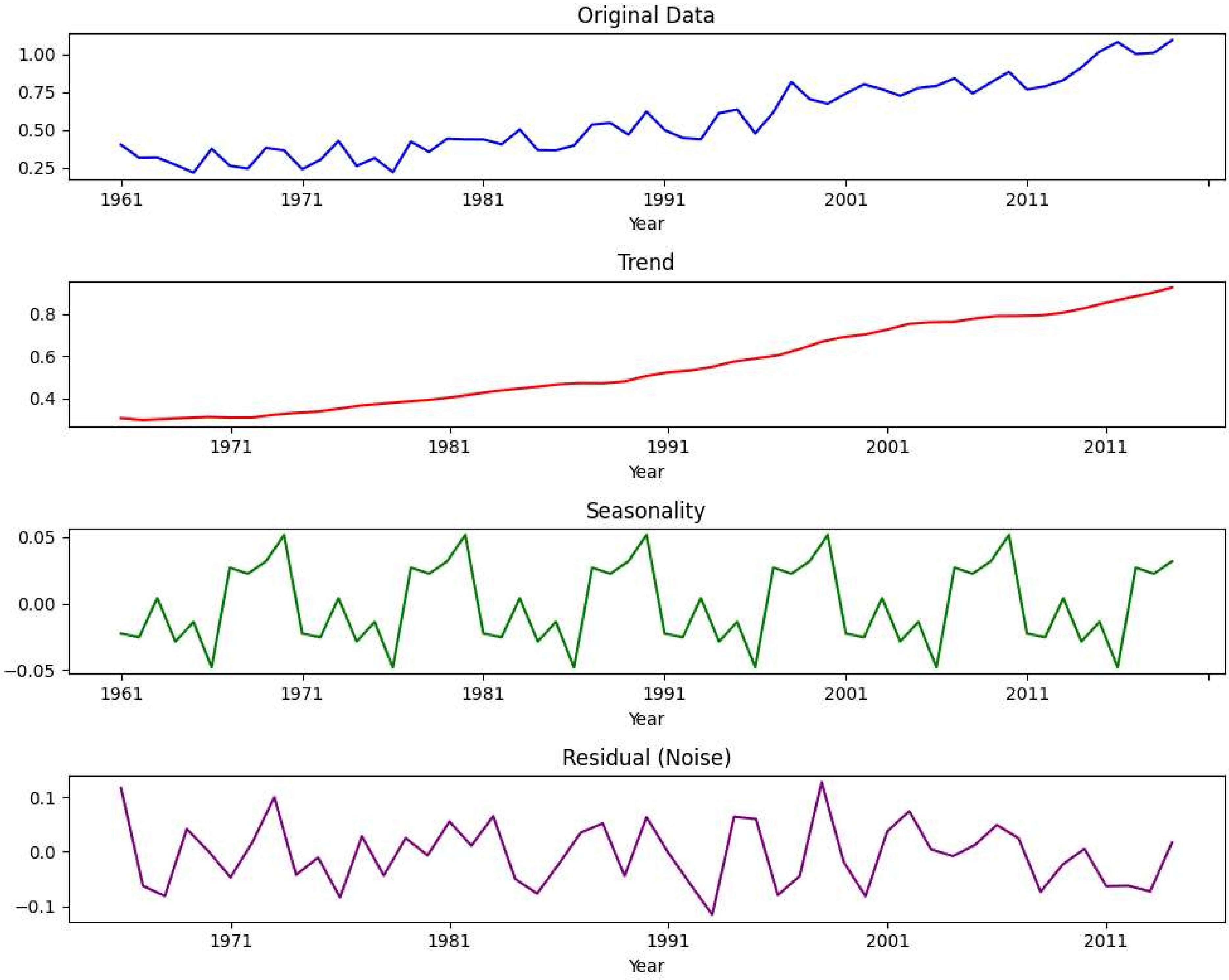
```
data.to_csv("average_temperatures.csv", index=False)
print(data)
```

```
    Year  Average Temperature
0   1961             0.402433
1   1962             0.315527
2   1963             0.317393
3   1964             0.269382
4   1965             0.217839
5   1966             0.376419
6   1967             0.263239
7   1968             0.244870
8   1969             0.382172
9   1970             0.365322
10  1971             0.240934
11  1972             0.302553
12  1973             0.427691
13  1974             0.261849
14  1975             0.314653
15  1976             0.221112
16  1977             0.422978
17  1978             0.355488
18  1979             0.442465
19  1980             0.438270
20  1981             0.437693
21  1982             0.404857
22  1983             0.503748
23  1984             0.366971
24  1985             0.365511
25  1986             0.398096
26  1987             0.535514
27  1988             0.546662
28  1989             0.469231
29  1990             0.621797
30  1991             0.499991
31  1992             0.447798
32  1993             0.439094
33  1994             0.611078
34  1995             0.635836
35  1996             0.477239
36  1997             0.617341
37  1998             0.818264
38  1999             0.704445
39  2000             0.674191
40  2001             0.741673
41  2002             0.802509
42  2003             0.769485
43  2004             0.726237
```

```
44  2005                0.777465
45  2006                0.791795
46  2007                0.842554
47  2008                0.742614
48  2009                0.814177
49  2010                0.884504
50  2011                0.768488
51  2012                0.788930
52  2013                0.829647
53  2014                0.913872
54  2015                1.018816
55  2016                1.081491
56  2017                1.003342
57  2018                1.010832
58  2019                1.094599
```

In [4]:

```
data.set_index("Year", inplace=True)

decomposition = seasonal_decompose(data["Average Temperature"], model="additive", period=10)

fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, figsize=(10, 8))

data["Average Temperature"].plot(ax=ax1, title="Original Data", color='blue')

decomposition.trend.plot(ax=ax2, title="Trend", color='red')
decomposition.seasonal.plot(ax=ax3, title="Seasonality", color='green')
decomposition.resid.plot(ax=ax4, title="Residual (Noise)", color='purple')

plt.tight_layout()
plt.show()
```


Original Data
1.00
0.75
0.50
0.25
1961
1971
1981
1991
2001
2011
Year
Trend
0.8
0.6
0.4
1971
1981
1991
2001
2011
Year
Seasonality
0.05
0.00
−0.05
1961
1971
1981
1991
2001
2011
Year
Residual (Noise)
0.1
0.0
−0.1
1971
1981
1991
2001
2011
Year

In [5]:

```python
result = adfuller(data["Average Temperature"])
print(f"ADF Statistic: {result[0]}")
print(f"p-value: {result[1]}")
```

```
ADF Statistic: 1.1279797201576414
p-value: 0.9954484841402742
```

In [6]:

```python
data.reset_index(inplace=True)
data["Temp_diff"] = data["Average Temperature"].diff()

def plot_temperature(df: pd.DataFrame, y: str, title: str) -> None:
    fig = px.line(df, x='Year', y=y, labels={'Year': 'Year'})
    fig.update_layout(template="simple_white", font=dict(size=18),
                      title_text=title, width=650, title_x=0.5, height=400)
    return fig.show()

plot_temperature(df=data, y='Temp_diff', title='Differenced Temperature Data')
```

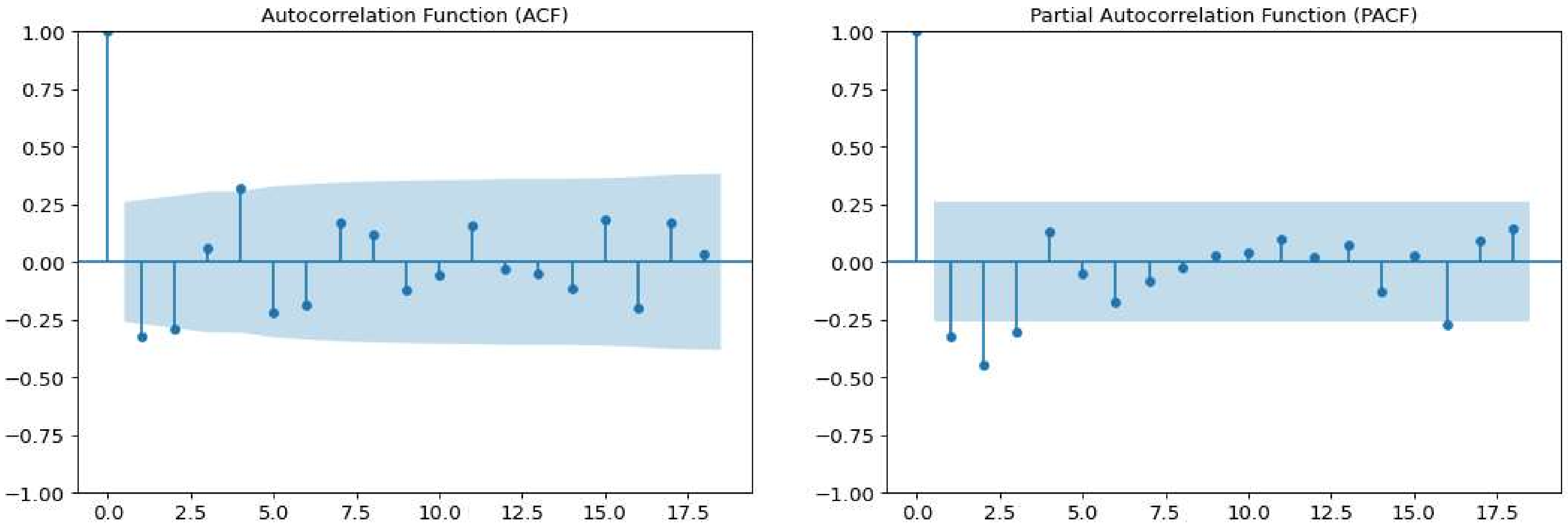
In [ ]:

```python
data["Temp_diff2"] = data["Temp_diff"].diff()
data.dropna(inplace=True)

result_diff2 = adfuller(data["Temp_diff2"])
print(f"ADF Statistic: {result_diff2[0]}")
print(f"p-value: {result_diff2[1]}")
```

```
ADF Statistic: -4.759518603359893
p-Value: 6.496008212398227e-05
```

In [8]:

```python
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 5), dpi=80)

plot_acf(data['Temp_diff'].dropna(), ax=ax1)
plot_pacf(data['Temp_diff'].dropna(), ax=ax2, method='ywm')

ax1.set_title("Autocorrelation Function (ACF)")
ax2.set_title("Partial Autocorrelation Function (PACF)")

ax1.tick_params(axis='both', labelsize=12)
ax2.tick_params(axis='both', labelsize=12)

plt.show()
```



In [9]:

```
data.set_index("Year", inplace=True)
data.index = data.index.astype(int)

sarima = SARIMAX(data['Average Temperature'],
                 order=(1,0,1),
                 seasonal_order=(1,0,1,10))
model_fit = sarima.fit()

predictions = model_fit.predict()
```

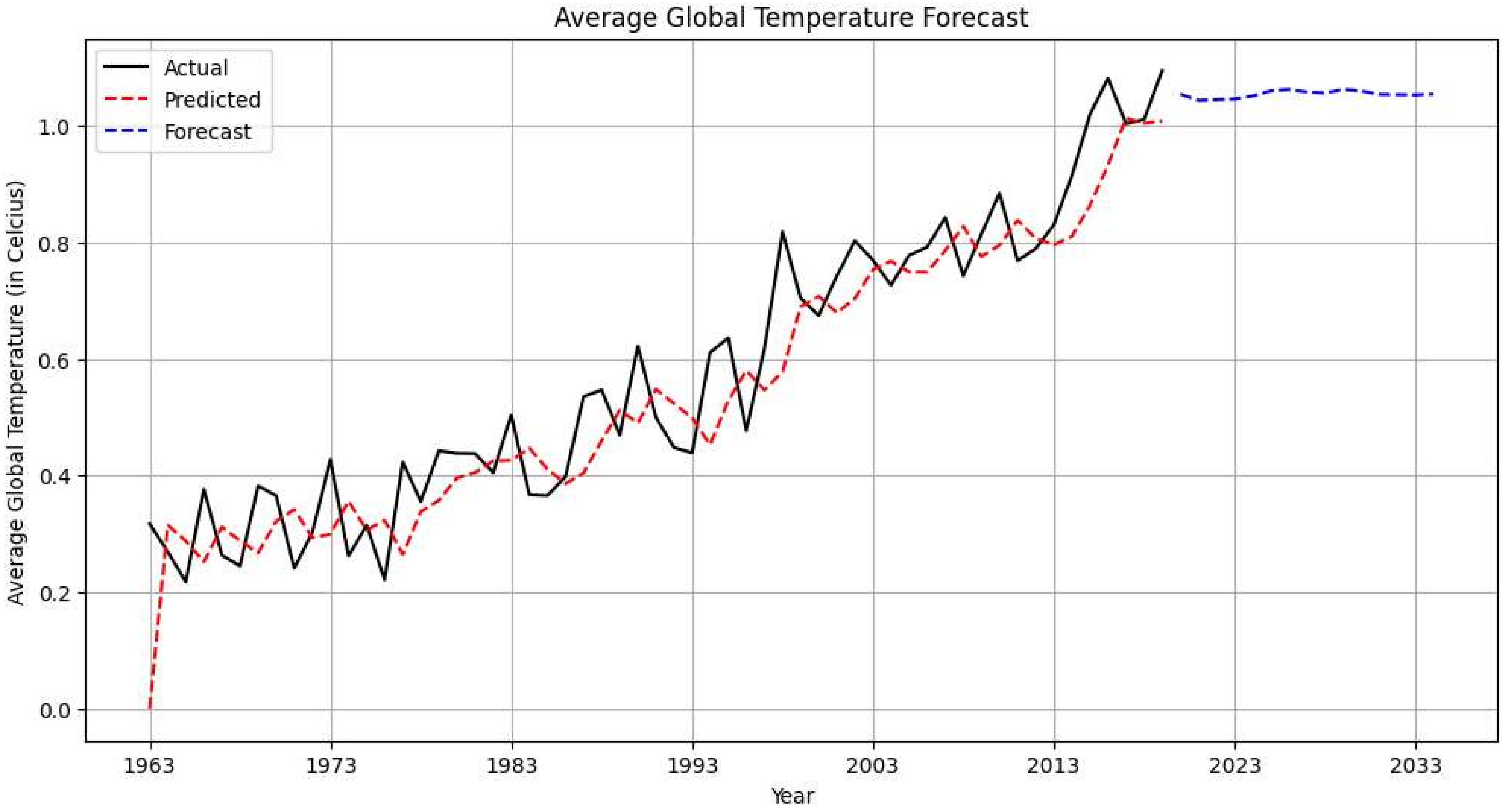
In [10]:

```
future_years = 15

last_year = data.index.max()
forecast_index = list(range(last_year + 1, last_year + future_years + 1))
forecast_values = model_fit.forecast(steps=future_years)
```

In [11]:

```
plt.figure(figsize=(12,6))
plt.plot(data.index, data['Average Temperature'], label="Actual", color="black")
plt.plot(data.index, predictions, label="Predicted", color="red", linestyle="dashed")
plt.plot(forecast_index, forecast_values, label="Forecast", color="blue", linestyle="dashed")

plt.xticks(ticks=range(data.index.min(), last_year + future_years + 1, 10))

plt.xlabel("Year")
plt.ylabel("Average Global Temperature (in Celcius)")
```

```python
plt.legend()
plt.title("Average Global Temperature Forecast")
plt.grid(True)
plt.show()
```



In [12]:

```python
actual = data["Average Temperature"]
predicted = model_fit.predict()

mae = mean_absolute_error(actual, predicted)
print(f"Mean Absolute Error (MAE): {mae:.4f}")
```

```
Mean Absolute Error (MAE): 0.0758
```