In [4]:

```
import csv


file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter


with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)

    for row in csv_reader:
        print(row)
```

```
['Variable:', 'pr']
['', 'Indonesia', 'Nangroe Aceh Darussalam', 'Bali', 'Bengkulu', 'Daera
h Istimewa Yogyakarta', 'Dki Jakarta', 'Jambi', 'Jawa Tengah', 'Jawa Ti
mur', 'Kalimantan Barat', 'Kalimantan Selatan', 'Kalimantan Tengah', 'K
alimantan Timur', 'Lampung', 'Nusatenggara Barat', 'Nusatenggara Timu
r', 'Sulawesi Tengah', 'Sulawesi Tenggara', 'Sumatera Barat', 'Sumatera
Utara', 'Bangka Belitung', 'Banten', 'Gorontalo', 'Papua Barat', 'Jawa
Barat', 'Kepulauan-riau', 'Maluku', 'Maluku Utara', 'Papua', 'Riau', 'S
ulawesi Barat', 'Sulawesi Selatan', 'Sulawesi Utara', 'Sumatera Selata
n']
['1901', '2707.15', '2350.16', '2485.27', '2756.80', '2217.35', '2571.4
0', '2673.11', '2689.61', '2187.89', '3067.57', '2548.72', '2822.82',
'2629.34', '2557.50', '1903.72', '1905.88', '1810.54', '2167.55', '279
4.03', '2256.87', '2546.37', '3331.59', '1976.33', '2962.31', '2950.6
9', '2434.48', '2513.81', '2278.00', '3335.33', '2252.06', '2736.45',
'2655.85', '2244.10', '2556.20']
['1902', '2556.10', '2278.19', '1970.37', '2569.83', '1697.90', '1715.6
0', '2318.95', '2005.51', '1718.50', '3155.70', '2501.52', '2730.91',
'2693.62', '2028.58', '1722.21', '1595.37', '1797.25', '2167.55', '230
```

In [5]:

```
import csv


file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter


seventeenth_values = []


with open(file_path, 'r') as csv_file:

    csv_reader = csv.reader(csv_file)


    next(csv_reader)
    next(csv_reader)


    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = row[17]
            seventeenth_values.append(seventeenth_value)


print(seventeenth_values)
```

```
['1810.54', '1797.25', '1815.26', '1817.44', '1800.83', '1805.41', '1817.5
9', '1839.00', '1911.99', '1915.94', '1693.12', '1737.10', '1737.45', '159
0.40', '1686.83', '1718.71', '1980.87', '1641.74', '1653.62', '1757.29',
'1806.73', '1696.42', '1704.40', '1877.58', '1689.73', '1673.56', '1673.6
2', '1626.94', '1640.04', '1569.27', '1762.95', '1733.57', '1792.97', '176
2.62', '1761.69', '1890.41', '1706.09', '1782.47', '1722.24', '1680.44',
'1522.65', '1817.85', '1817.85', '1817.85', '1817.85', '1811.07', '1861.1
0', '1795.66', '1813.29', '1799.04', '1725.74', '1961.94', '1809.65', '200
4.35', '2086.91', '1953.79', '1868.31', '1809.54', '1838.43', '1904.34',
'1782.90', '1851.33', '1740.83', '1965.45', '1630.28', '1838.86', '1742.3
1', '1878.51', '1676.67', '1823.83', '2006.61', '1608.22', '1954.17', '189
4.99', '2121.55', '1753.30', '1725.20', '1945.86', '1779.78', '1766.77',
'1838.26', '1671.90', '1728.24', '1907.54', '1831.30', '1867.05', '1779.7
4', '1861.21', '1768.95', '1792.71', '1761.00', '1752.19', '1717.76', '177
3.37', '1819.76', '1844.77', '1720.73', '1773.49', '1782.41', '1858.25',
'1750.23', '1705.13', '1763.76', '1614.43', '1788.83', '1691.37', '1867.2
9', '1942.68', '1726.43', '2214.18', '1795.25', '1837.03', '1855.27', '180
9.21', '1840.21', '1815.53', '1952.53', '2035.02', '1874.03', '2131.90',
'2023.00']
```

In [6]:

```python
import csv

file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter

seventeenth_values = []

with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)

    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = row[17]
            seventeenth_values.append(seventeenth_value)


for value in seventeenth_values:
    print(value)
```

```
1810.54
1797.25
1815.26
1817.44
1800.83
1805.41
1817.59
1839.00
1911.99
1915.94
1693.12
1737.10
1737.45
1590.40
1686.83
1718.71
1980.87
1641.74
1653.62
```

In [7]:

```python
import csv

file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter

seventeenth_values = []

with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)
    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = row[17]
            seventeenth_values.append(seventeenth_value)

print("Rote Island precipitation,Year")

for year, value in zip(range(1901, 2022), seventeenth_values):
    print(f"{value},{year}")
```

```
Rote Island precipitation,Year
1810.54,1901
1797.25,1902
1815.26,1903
1817.44,1904
1800.83,1905
1805.41,1906
1817.59,1907
1839.00,1908
1911.99,1909
1915.94,1910
1693.12,1911
1737.10,1912
1737.45,1913
1590.40,1914
1686.83,1915
1718.71,1916
1980.87,1917
1641.74,1918
```

In [3]:

```
import csv
import pandas as pd

file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter

seventeenth_values = []

with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)
    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = float(row[17]) / 12  # Divide by 12 to convert to mm
            seventeenth_values.append(seventeenth_value)

#  DataFrame
data = {
    "Year": list(range(1901, 2022)),
    "Mean Precipitation (mm)": seventeenth_values
}

df = pd.DataFrame(data)

#  average precipitation
average_precipitation = df["Mean Precipitation (mm)"].mean()


output_file = "precipitation_data.xlsx"
df.to_excel(output_file, index=False)

print(f"Data has been saved to {output_file}")
print("\nAverage precipitation:", round(average_precipitation, 2), "mm")
```

```
Data has been saved to precipitation_data.xlsx

Average precipitation: 150.34 mm
```

In [9]:

```
import csv

file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter

seventeenth_values = []

with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)
    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = float(row[17]) / 12  # Divide by 12 to convert to mm
            seventeenth_values.append(seventeenth_value)

average_precipitation = sum(seventeenth_values) / len(seventeenth_values)
highest_value = max(seventeenth_values)
lowest_value = min(seventeenth_values)

print("Mean precipitation Rote Island region (mm),Year,Mangrove carbon stock precipitati

for year, value in zip(range(1901, 2022), seventeenth_values):
    mangrove_range = max((value - average_precipitation) / (highest_value - average_prec
    seagrass_range = max((average_precipitation - value) / (average_precipitation - lowe
    print("{:.2f},{:d},{:.2f},{:.2f}".format(value, year, mangrove_range, seagrass_range

    Average_precipitation = total_precipitation / len(seventeenth_values)
print("\nAverage precipitation:", round(average_precipitation, 2), "mm")
```

```
Mean precipitation Rote Island region (mm),Year,Mangrove carbon stock p
recipitation gain (%),Seagrass carbon stock precipitation loss (%)
150.88,1901,0.16,0.00
149.77,1902,0.00,0.41
151.27,1903,0.27,0.00
151.45,1904,0.33,0.00
150.07,1905,0.00,0.19
150.45,1906,0.03,0.00
151.47,1907,0.33,0.00
153.25,1908,0.85,0.00
159.33,1909,2.63,0.00
159.66,1910,2.73,0.00
141.09,1911,0.00,6.70
144.76,1912,0.00,4.04
144.79,1913,0.00,4.02
132.53,1914,0.00,12.91
140.57,1915,0.00,7.08
143.23,1916,0.00,5.16
165.07,1917,4.31,0.00
```

In [10]:

```
import csv

file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter

seventeenth_values = []

with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)
    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = float(row[17]) / 12  # Divide by 12 to convert to mm
            seventeenth_values.append(seventeenth_value)

average_precipitation = sum(seventeenth_values) / len(seventeenth_values)
highest_value = max(seventeenth_values)
lowest_value = min(seventeenth_values)

print("Mean precipitation Rote Island region (mm), Year, Mangrove carbon stock precipita

for year, value in zip(range(1901, 2022), seventeenth_values):
    mangrove_range = max((value - average_precipitation) / (highest_value - average_prec
    seagrass_range = max((average_precipitation - value) / (average_precipitation - lowe
    print("{:.2f}, {:d}, {:.2f}, {:.2f}".format(value, year, mangrove_range, seagrass_ra
```

```
Mean precipitation Rote Island region (mm), Year, Mangrove carbon stock
precipitation gain (%), Seagrass carbon stock precipitation loss (%)
150.88, 1901, 0.16, 0.00
149.77, 1902, 0.00, 0.41
151.27, 1903, 0.27, 0.00
151.45, 1904, 0.33, 0.00
150.07, 1905, 0.00, 0.19
150.45, 1906, 0.03, 0.00
151.47, 1907, 0.33, 0.00
153.25, 1908, 0.85, 0.00
159.33, 1909, 2.63, 0.00
159.66, 1910, 2.73, 0.00
141.09, 1911, 0.00, 6.70
144.76, 1912, 0.00, 4.04
144.79, 1913, 0.00, 4.02
132.53, 1914, 0.00, 12.91
140.57, 1915, 0.00, 7.08
143.23, 1916, 0.00, 5.16
165.07, 1917, 4.31, 0.00
```

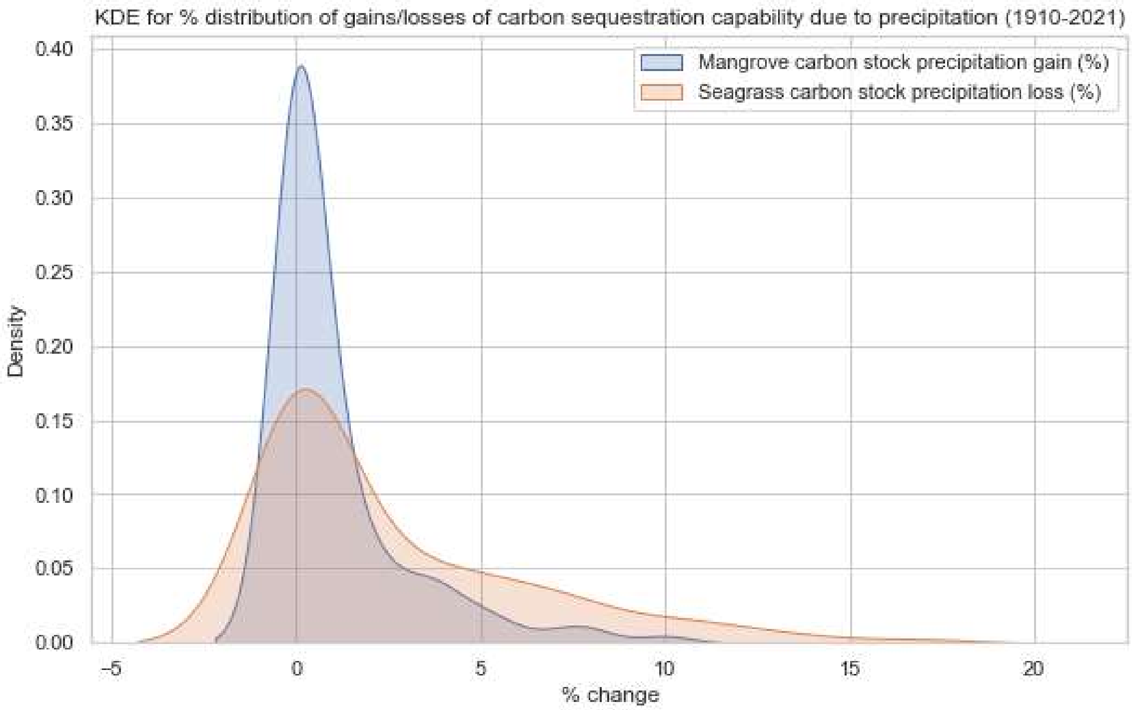
In [11]:

```
import openpyxl

file_path2 = r"C:\Users\Rishab\Documents\UCL\Masters Thesis\Relevant journals\Alternativ

workbook = openpyxl.load_workbook(file_path2)
sheet = workbook.active

for row in sheet.iter_rows(values_only=True):
    print(row)
```

```
('Category', 'Annual Mean')
(1901, 25.4)
(1902, 25.37)
(1903, 25.37)
(1904, 25.36)
(1905, 25.41)
(1906, 25.41)
(1907, 25.33)
(1908, 25.4)
(1909, 25.39)
(1910, 25.37)
(1911, 25.54)
(1912, 25.6)
(1913, 25.56)
(1914, 25.65)
(1915, 25.62)
(1916, 25.46)
(1917, 25.44)
(1918, 25.34)
```

In [12]:

```
import csv
import matplotlib.pyplot as plt
import seaborn as sns
import openpyxl

file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter

seventeenth_values = []

with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)
    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = float(row[17]) / 12  # Divide by 12 to convert to mm
            seventeenth_values.append(seventeenth_value)

average_precipitation = sum(seventeenth_values) / len(seventeenth_values)
highest_value = max(seventeenth_values)
lowest_value = min(seventeenth_values)


file_path2 = r"C:\Users\Rishab\Documents\UCL\Masters Thesis\Relevant journals\Alternativ

workbook = openpyxl.load_workbook(file_path2)
sheet = workbook.active

temperatures = [row[1] for row in sheet.iter_rows(values_only=True) if isinstance(row[0]


years = list(range(1901, 2022))
mangrove_ranges = [min(max((value - average_precipitation) / (highest_value - average_pr
seagrass_ranges = [min(max((average_precipitation - value) / (average_precipitation - lo


sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))

sns.kdeplot(mangrove_ranges, label="Mangrove carbon stock precipitation gain (%)", shade


sns.kdeplot(seagrass_ranges, label="Seagrass carbon stock precipitation loss (%)", shade


plt.xlabel("% change")

plt.ylabel("Density")
plt.title("KDE for % distribution of gains/losses of carbon sequestration capability due

plt.legend()
plt.show()
```


KDE for % distribution of gains/losses of carbon sequestration capability due to precipitation (1910-2021)
Mangrove carbon stock precipitation gain (%)
Seagrass carbon stock precipitation loss (%)
Density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−5
0
5
10
15
20
% change

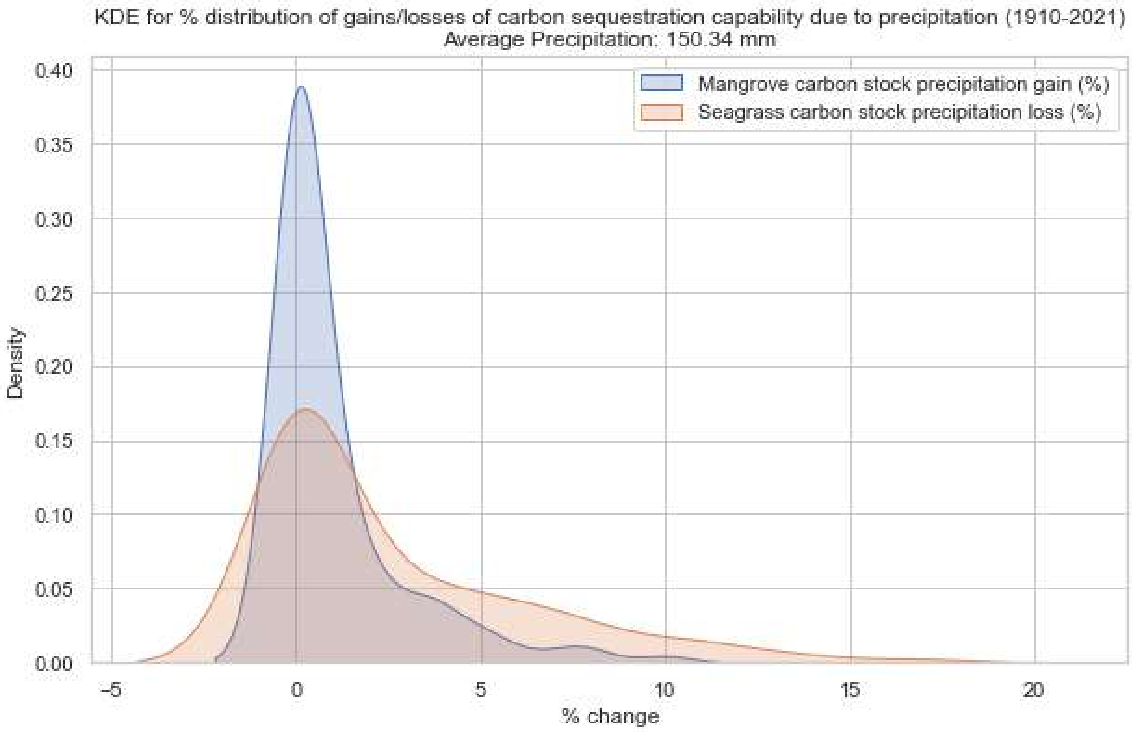
In [13]:

```
import csv
import matplotlib.pyplot as plt
import seaborn as sns
import openpyxl
import pandas as pd

file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter

seventeenth_values = []

with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)
    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = float(row[17]) / 12  # Divide by 12 to convert to mm
            seventeenth_values.append(seventeenth_value)

average_precipitation = sum(seventeenth_values) / len(seventeenth_values)
highest_value = max(seventeenth_values)
lowest_value = min(seventeenth_values)

#temperature new file
file_path2 = r"C:\Users\Rishab\Documents\UCL\Masters Thesis\Relevant journals\Alternativ

workbook = openpyxl.load_workbook(file_path2)
sheet = workbook.active

temperatures = [row[1] for row in sheet.iter_rows(values_only=True) if isinstance(row[0]

# average temperature
average_temperature = sum(temperatures) / len(temperatures)

#  lists to store data for plotting
years = list(range(1901, 2022))
mangrove_ranges = [min(max((value - average_precipitation) / (highest_value - average_pr
seagrass_ranges = [min(max((average_precipitation - value) / (average_precipitation - lo

#  display the data
data = {'Year': years,
        'Precipitation (mm)': seventeenth_values,
        'Temperature (°C)': temperatures,
        'Mangrove Gain (%)': mangrove_ranges,
        'Seagrass Loss (%)': seagrass_ranges}

df = pd.DataFrame(data)


print(df)


print("\nAverage precipitation:", round(average_precipitation, 2), "mm")
print("Average Temperature:", round(average_temperature, 2), "°C")
```

```
     Year  Precipitation (mm)  Temperature (°C)  Mangrove Gain (%)  \
1    1902          149.770833             25.37           0.000000
2    1903          151.271667             25.37           0.273203
3    1904          151.453333             25.36           0.326357
4    1905          150.069167             25.41           0.000000
..    ...                 ...               ...                ...
116  2017          162.710833             26.03           3.620233
117  2018          169.585000             26.04           5.631573
118  2019          156.169167             26.20           1.706181
119  2020          177.658333             26.18           7.993781
120  2021          168.583333             25.99           5.338491

     Seagrass Loss (%)
0             0.000000
1             0.411115
2             0.000000
3             0.000000
4             0.194843
..                 ...
116           0.000000
117           0.000000
118           0.000000
119           0.000000
120           0.000000

[121 rows x 5 columns]

Average precipitation: 150.34 mm
Average Temperature: 25.62 °C
```

In [15]:

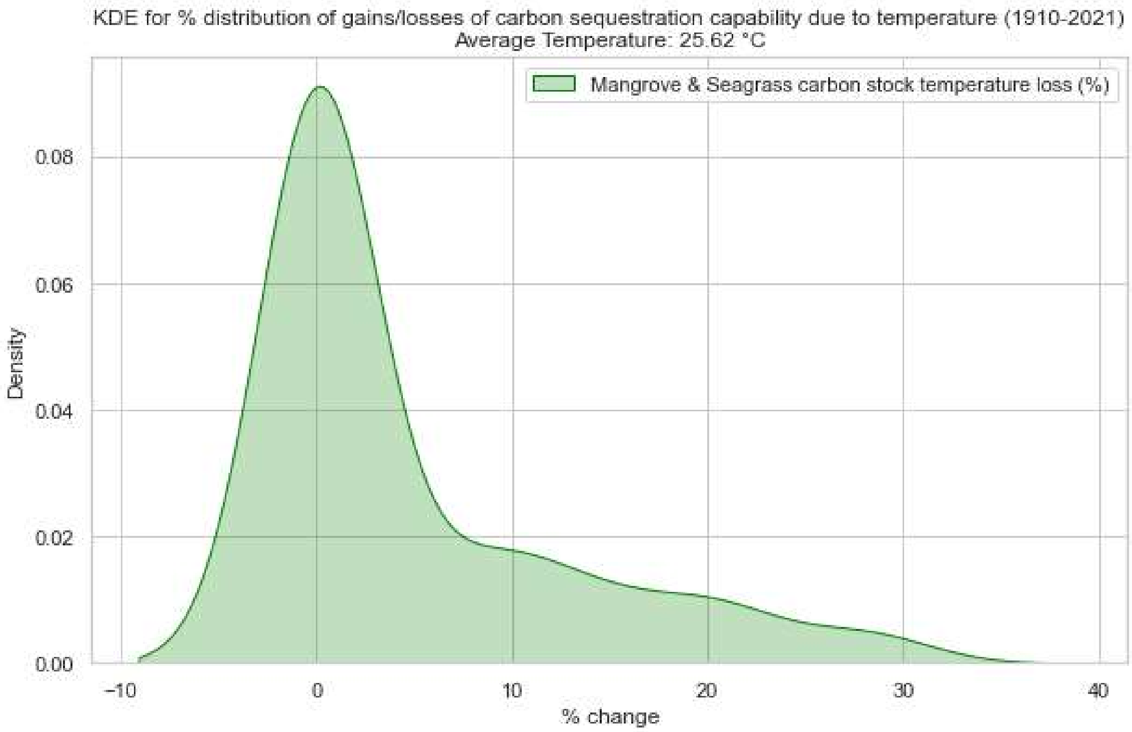
```
import csv
import matplotlib.pyplot as plt
import seaborn as sns

file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter

seventeenth_values = []

with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)
    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = float(row[17]) / 12  # Divide by 12 to convert to mm
            seventeenth_values.append(seventeenth_value)

average_precipitation = sum(seventeenth_values) / len(seventeenth_values)
highest_value = max(seventeenth_values)
lowest_value = min(seventeenth_values)


years = list(range(1901, 2022))
mangrove_ranges = [max((value - average_precipitation) / (highest_value - average_precip
seagrass_ranges = [max((average_precipitation - value) / (average_precipitation - lowest_


mangrove_ranges = [min(max((value - average_precipitation) / (highest_value - average_pr
seagrass_ranges = [min(max((average_precipitation - value) / (average_precipitation - lo

#KDE plot
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))


sns.kdeplot(mangrove_ranges, label="Mangrove carbon stock precipitation gain (%)", shade


sns.kdeplot(seagrass_ranges, label="Seagrass carbon stock precipitation loss (%)", shade

plt.xlabel("% change")
plt.ylabel("Density")

plt.title("KDE for % distribution of gains/losses of carbon sequestration capability due
plt.legend()
plt.show()
```


KDE for % distribution of gains/losses of carbon sequestration capability due to precipitation (1910-2021)
Average Precipitation: 150.34 mm
Mangrove carbon stock precipitation gain (%)
Seagrass carbon stock precipitation loss (%)
Density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−5
0
5
10
15
20
% change

In [16]:

```
import openpyxl
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


file_path2 = r"C:\Users\Rishab\Documents\UCL\Masters Thesis\Relevant journals\Alternativ

workbook = openpyxl.load_workbook(file_path2)
sheet = workbook.active

temperatures = [row[1] for row in sheet.iter_rows(values_only=True) if isinstance(row[0]


average_temperature = sum(temperatures) / len(temperatures)


highest_value = max(temperatures)
temperature_ranges = [min(max((value - average_temperature) / (highest_value - average_t


data = {'Year': list(range(1901, 2022)),
        'Temperature (°C)': temperatures,
        'Temperature Loss (%)': temperature_ranges}

temperature_df = pd.DataFrame(data)


print(temperature_df)


print("Average Temperature:", average_temperature)


sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))

# Modified KDE plot
sns.kdeplot(temperature_ranges, label="Mangrove & Seagrass carbon stock temperature loss


plt.xlabel("% change")
plt.ylabel("Density")
plt.title("KDE for % distribution of gains/losses of carbon sequestration capability due
plt.legend()
plt.show()
```

```
     Year  Temperature (°C)  Temperature Loss (%)
0    1901             25.40              0.000000
1    1902             25.37              0.000000
2    1903             25.37              0.000000
3    1904             25.36              0.000000
4    1905             25.41              0.000000
..    ...               ...                   ...
116  2017             26.03             20.149254
117  2018             26.04             20.641791
118  2019             26.20             28.522388
119  2020             26.18             27.537313
120  2021             25.99             18.179104

[121 rows x 3 columns]
Average Temperature: 25.620909090909088
```



In [17]:

```
import pandas as pd

file_path3 = r"C:\Users\Rishab\Documents\UCL\Masters Thesis\Relevant journals\Alternativ
df = pd.read_excel(file_path3)

print(df)
```

	Year	Month	Mean-Temperature	Mean Precipitation
0	1930	Jan	25.34	288.32
1	1930	Feb	25.36	258.02
2	1930	Mar	25.57	281.38
3	1930	Apr	25.76	259.12
4	1930	May	25.85	231.61
5	1930	Jun	25.30	182.00
6	1930	Jul	24.98	160.35
7	1930	Aug	25.04	164.16
8	1930	Sep	25.36	164.56
9	1930	Oct	25.64	209.07
10	1930	Nov	25.93	245.01
11	1930	Dec	25.55	284.29
12	1960	Jan	25.49	284.78
13	1960	Feb	25.50	257.63
14	1960	Mar	25.65	286.02
15	1960	Apr	25.83	271.09
16	1960	May	25.81	239.75
17	1960	Jun	25.38	187.23
18	1960	Jul	24.98	174.09
19	1960	Aug	25.09	175.89
20	1960	Sep	25.38	173.35
21	1960	Oct	25.72	212.34
22	1960	Nov	25.78	248.42
23	1960	Dec	25.61	284.87
24	1990	Jan	25.36	281.48
25	1990	Feb	25.47	251.89
26	1990	Mar	25.65	281.53
27	1990	Apr	25.89	262.83
28	1990	May	25.86	234.51
29	1990	Jun	25.44	180.34
30	1990	Jul	25.06	167.03
31	1990	Aug	25.15	165.36
32	1990	Sep	25.42	176.18
33	1990	Oct	25.77	207.00
34	1990	Nov	25.80	242.46
35	1990	Dec	25.60	284.16
36	2020	Jan	25.79	281.61
37	2020	Feb	25.88	256.38
38	2020	Mar	26.07	290.57
39	2020	Apr	26.28	278.61
40	2020	May	26.21	227.23
41	2020	Jun	25.86	186.20
42	2020	Jul	25.54	167.26
43	2020	Aug	25.62	162.13
44	2020	Sep	25.81	166.72
45	2020	Oct	26.12	211.66
46	2020	Nov	26.15	256.60
47	2020	Dec	26.02	297.32

In [15]:

```
import csv
import matplotlib.pyplot as plt
import seaborn as sns
import openpyxl
import pandas as pd

file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter

seventeenth_values = []

with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)
    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = float(row[17]) / 12  # Divide by 12 to convert to mm
            seventeenth_values.append(seventeenth_value)

average_precipitation = sum(seventeenth_values) / len(seventeenth_values)
highest_value = max(seventeenth_values)
lowest_value = min(seventeenth_values)

# read temp data
file_path2 = r"C:\Users\Rishab\Documents\UCL\Masters Thesis\Relevant journals\Alternativ

workbook = openpyxl.load_workbook(file_path2)
sheet = workbook.active

temperatures = [row[1] for row in sheet.iter_rows(values_only=True) if isinstance(row[0]


average_temperature = sum(temperatures) / len(temperatures)

# ranges for Mangrove and Seagrass (Temperature Loss in %)
temperature_ranges = [15 if value > average_temperature else 0 for value in temperatures


years = list(range(1901, 2022))
mangrove_ranges = [min(max((value - average_precipitation) / (highest_value - average_pr
seagrass_ranges = [min(max((average_precipitation - value) / (average_precipitation - lo


data = {'Year': years,
        'Precipitation (mm)': seventeenth_values,
        'Temperature (°C)': temperatures,
        'Mangrove Gain (%)': mangrove_ranges,
        'Seagrass Loss (%)': seagrass_ranges,
        'Mangrove and Seagrass Temp Loss (%)': temperature_ranges}

df = pd.DataFrame(data)


print(df)


print("Average Precipitation:", average_precipitation)
print("Average Temperature:", average_temperature)
```

```
0     1901         150.878333          25.40          0.158116
1     1902         149.770833          25.37          0.000000
2     1903         151.271667          25.37          0.273203
3     1904         151.453333          25.36          0.326357
4     1905         150.069167          25.41          0.000000
..     ...                ...            ...               ...
116   2017         162.710833          26.03          3.620233
117   2018         169.585000          26.04          5.631573
118   2019         156.169167          26.20          1.706181
119   2020         177.658333          26.18          7.993781
120   2021         168.583333          25.99          5.338491

     Seagrass Loss (%)  Mangrove and Seagrass Temp Loss (%)
0             0.000000                                    0
1             0.411115                                    0
2             0.000000                                    0
3             0.000000                                    0
4             0.194843                                    0
..                 ...                                  ...
116           0.000000                                   15
117           0.000000                                   15
118           0.000000                                   15
119           0.000000                                   15
120           0.000000                                   15

[121 rows x 6 columns]
Average Precipitation: 150.3379407713499
Average Temperature: 25.620909090909088
```

In [16]:

```python
import csv
import matplotlib.pyplot as plt
import seaborn as sns
import openpyxl
import pandas as pd

file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter

# Read precipitation data
seventeenth_values = []
with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)
    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = float(row[17]) / 12  # Divide by 12 to convert to mm
```

In [51]:

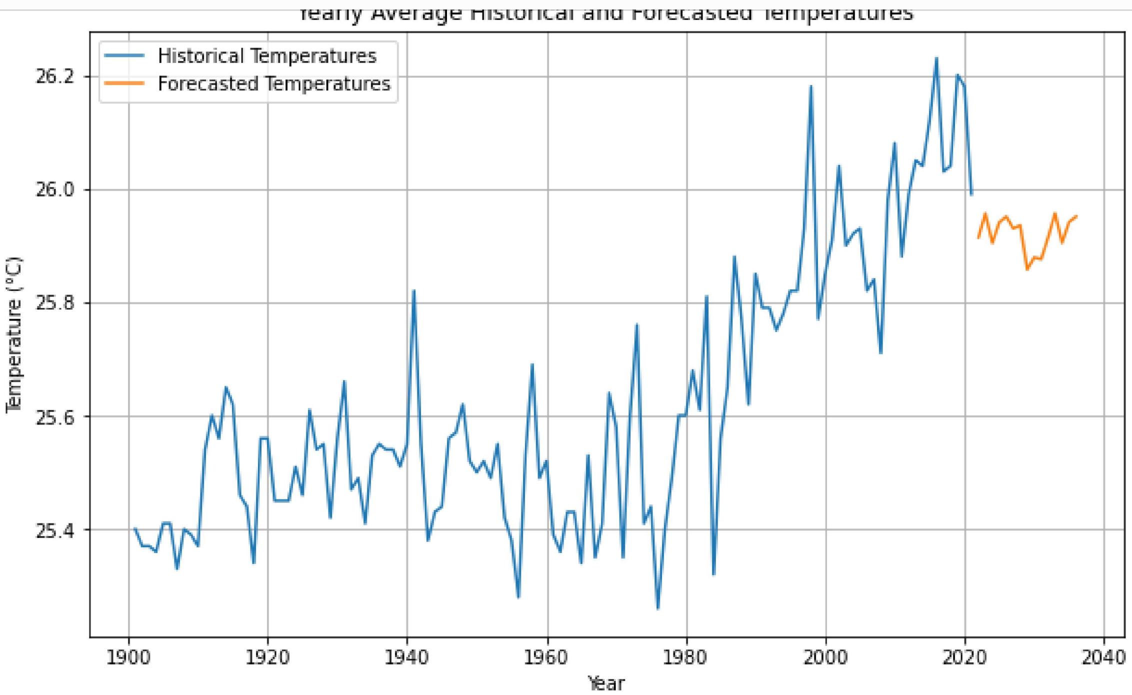
```
     Year  Precipitation (mm)  Temperature (°C)  Mangrove Gain (%)  \
0    1901          150.878333             25.40           0.158116
1    1902          149.770833             25.37           0.000000
2    1903          151.271667             25.37           0.273203
3    1904          151.453333             25.36           0.326357
4    1905          150.069167             25.41           0.000000
..    ...                 ...               ...                ...
116  2017          162.710833             26.03           3.620233
117  2018          169.585000             26.04           5.631573
118  2019          156.169167             26.20           1.706181
119  2020          177.658333             26.18           7.993781
120  2021          168.583333             25.99           5.338491

     Seagrass Loss (%)  Mangrove and Seagrass Temp Loss (%)  \
0             0.000000                             0.000000
1             0.411115                             0.000000
2             0.000000                             0.000000
3             0.000000                             0.000000
4             0.194843                             0.000000
..                 ...                                  ...
116           0.000000                             0.077238
117           0.000000                             0.079126
118           0.000000                             0.109335
119           0.000000                             0.105559
120           0.000000                             0.069686

     Yearly average historical and forecasted mangrove and seagrass temper
ature loss (%)
0                                             0.000000
1                                             0.000000
2                                             0.000000
3                                             0.000000
4                                             0.000000
..                                                 ...
116                                           0.077238
117                                           0.079126
118                                           0.109335
119                                           0.105559
120                                           0.069686

[121 rows x 7 columns]
```

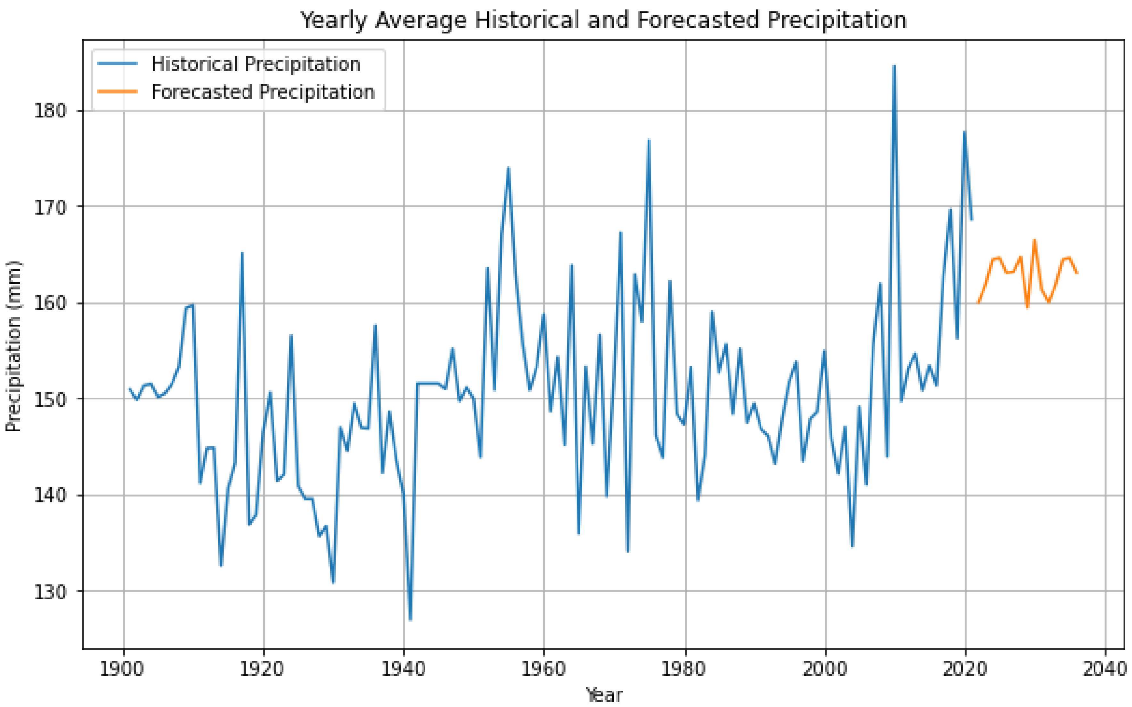
In [24]:

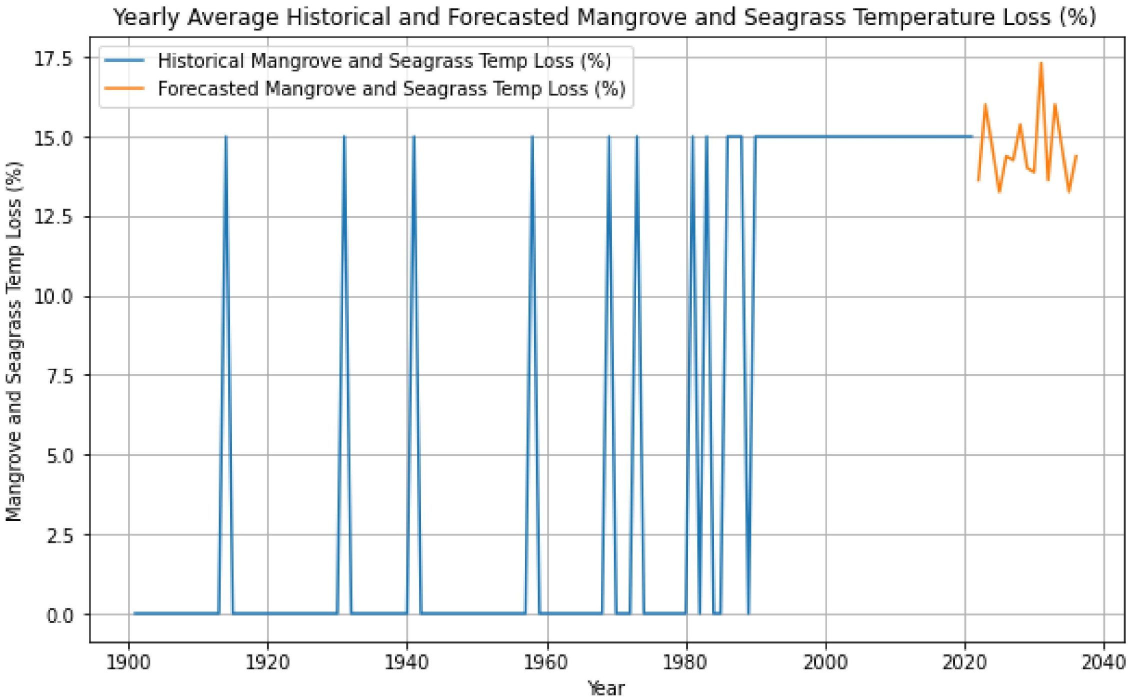
```
import csv
import matplotlib.pyplot as plt
import openpyxl
import pandas as pd
from statsmodels.tsa.holtwinters import ExponentialSmoothing


file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter

seventeenth_values = []

with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)
    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = float(row[17]) / 12  # Divide by 12 to convert to mm
            seventeenth_values.append(seventeenth_value)

average_precipitation = sum(seventeenth_values) / len(seventeenth_values)
highest_value = max(seventeenth_values)
lowest_value = min(seventeenth_values)



#  ranges for Mangrove and Seagrass (Temperature Loss in %)
temperature_ranges = [15 if value > average_temperature else 0 for value in temperatures


years = list(range(1901, 2022))
mangrove_ranges = [min(max((value - average_precipitation) / (highest_value - average_pr
seagrass_ranges = [min(max((value - average_precipitation) / (highest_value - average_pr

data = {'Year': years,
        'Precipitation (mm)': seventeenth_values,
        'Temperature (°C)': temperatures,
        'Mangrove Gain (%)': mangrove_ranges,
        'Seagrass Loss (%)': seagrass_ranges,
        'Mangrove and Seagrass Temp Loss (%)': temperature_ranges}

df = pd.DataFrame(data)


historical_temperatures = df['Temperature (°C)']

historical_precipitation = df['Precipitation (mm)']

historical_mangrove_gain = df['Mangrove Gain (%)']

historical_seagrass_loss = df['Seagrass Loss (%)']

historical_temp_loss = df['Mangrove and Seagrass Temp Loss (%)']

# Split data into training and testing set
train_data = historical_temperatures[:-15]
test_data = historical_temperatures[-15:]

# Exponential Smoothing for temp
```

```
model_temp = ExponentialSmoothing(train_data, seasonal='add', seasonal_periods=10)
model_fit_temp = model_temp.fit()

# Forecasted temp for next 15 yrs
forecasted_temperatures = model_fit_temp.forecast(steps=15)

# DataFrame for  forecasted temperatures Y2-Y15
forecasted_years = list(range(2022, 2037))
forecasted_data_temp = {'Year': forecasted_years, 'Forecasted Temperature (°C)': forecas
forecasted_df_temp = pd.DataFrame(forecasted_data_temp)

# Plotting forecasted temperatures
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], historical_temperatures, label='Historical Temperatures')
plt.plot(forecasted_df_temp['Year'], forecasted_df_temp['Forecasted Temperature (°C)'],
plt.xlabel('Year')
plt.ylabel('Temperature (°C)')
plt.title('Yearly Average Historical and Forecasted Temperatures')
plt.legend()
plt.grid()
plt.show()

# Fit Exponential Smoothing for precip
model_precip = ExponentialSmoothing(historical_precipitation, seasonal='add', seasonal_p
model_fit_precip = model_precip.fit()

# Forecasted precip for  15 years
forecasted_precipitation = model_fit_precip.forecast(steps=15)

# DataFrame for  forecasted precip
forecasted_data_precip = {'Year': forecasted_years, 'Forecasted Precipitation (mm)': for
forecasted_df_precip = pd.DataFrame(forecasted_data_precip)

# Plotting the forecasted precip
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], historical_precipitation, label='Historical Precipitation')
plt.plot(forecasted_df_precip['Year'], forecasted_df_precip['Forecasted Precipitation (m
plt.xlabel('Year')
plt.ylabel('Precipitation (mm)')
plt.title('Yearly Average Historical and Forecasted Precipitation')
plt.legend()
plt.grid()
plt.show()


# Exponential Smoothing for Mangrove and Seagrass Temp Loss (%)
model_temp_loss = ExponentialSmoothing(historical_temp_loss, seasonal='add', seasonal_pe
model_fit_temp_loss = model_temp_loss.fit()

# Forecasted Mangrove and Seagrass Temp Loss (%)
forecasted_temp_loss = model_fit_temp_loss.forecast(steps=15)

# DataFrame for forecasted temperature loss Y2-Y15
forecasted_data_temp_loss = {'Year': forecasted_years, 'Forecasted Mangrove and Seagrass
forecasted_df_temp_loss = pd.DataFrame(forecasted_data_temp_loss)

# Plottingforecasted temperature loss
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], historical_temp_loss, label='Historical Mangrove and Seagrass Temp
plt.plot(forecasted_df_temp_loss['Year'], forecasted_df_temp_loss['Forecasted Mangrove a
plt.xlabel('Year')
```

```
plt.ylabel('Mangrove and Seagrass Temp Loss (%)')
plt.title('Yearly Average Historical and Forecasted Mangrove and Seagrass Temperature Lo
plt.legend()
plt.grid()
plt.show()
```






Yearly Average Historical and Forecasted Mangrove and Seagrass Temperature Loss (%)
Historical Mangrove and Seagrass Temp Loss (%)
Forecasted Mangrove and Seagrass Temp Loss (%)
Mangrove and Seagrass Temp Loss (%)
Year
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
1900
1920
1940
1960
1980
2000
2020
2040

In [25]:

```
import csv
import matplotlib.pyplot as plt
import openpyxl
import pandas as pd
from statsmodels.tsa.holtwinters import ExponentialSmoothing

# Read and preprocess data
file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter

seventeenth_values = []

with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)
    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = float(row[17]) / 12  # Divide by 12 to convert to mm
            seventeenth_values.append(seventeenth_value)

average_precipitation = sum(seventeenth_values) / len(seventeenth_values)
highest_value = max(seventeenth_values)
lowest_value = min(seventeenth_values)




forecasted_temperatures = model_fit_temp.forecast(steps=15)


forecasted_years = list(range(2022, 2037))
forecasted_data_temp = {'Year': forecasted_years, 'Forecasted Temperature (°C)': forecas
forecasted_df_temp = pd.DataFrame(forecasted_data_temp)

#  forecasted temperatures in a table
print("Forecasted Temperatures:")
print(forecasted_df_temp)


model_precip = ExponentialSmoothing(historical_precipitation, seasonal='add', seasonal_p
model_fit_precip = model_precip.fit()


forecasted_precipitation = model_fit_precip.forecast(steps=15)


forecasted_data_precip = {'Year': forecasted_years, 'Forecasted Precipitation (mm)': for
forecasted_df_precip = pd.DataFrame(forecasted_data_precip)

print("\nForecasted Precipitation:")
print(forecasted_df_precip)
```

Forecasted Temperatures:

	Year	Forecasted Temperature (°C)
106	2022	25.913807
107	2023	25.956468
108	2024	25.904194
109	2025	25.940828
110	2026	25.950688
111	2027	25.929520
112	2028	25.935549
113	2029	25.857190
114	2030	25.878596
115	2031	25.875588
116	2032	25.913807
117	2033	25.956468
118	2034	25.904194
119	2035	25.940828
120	2036	25.950688

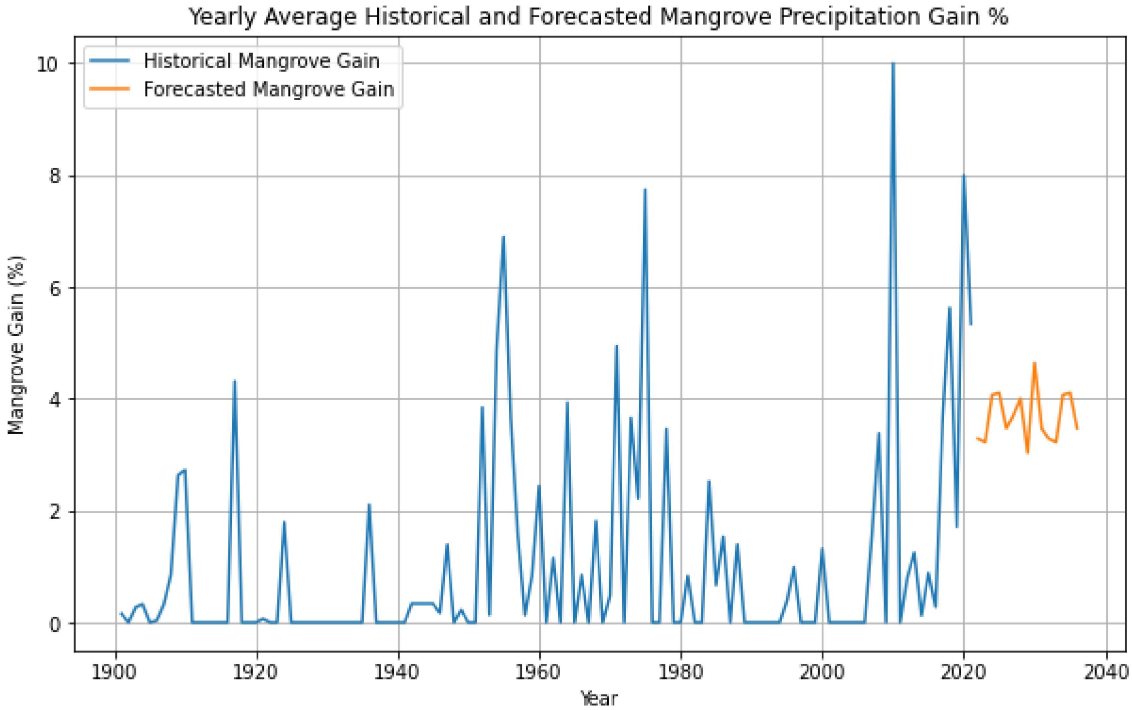
Forecasted Precipitation:

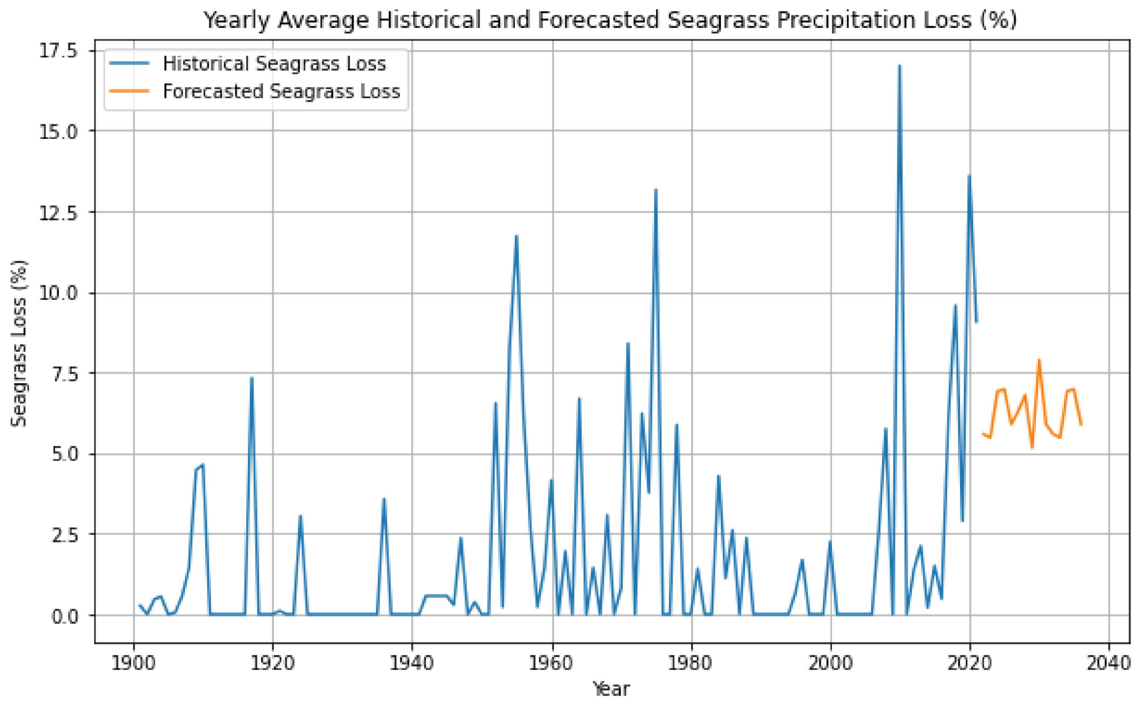
	Year	Forecasted Precipitation (mm)
121	2022	159.959674
122	2023	161.720156
123	2024	164.384597
124	2025	164.567775
125	2026	162.997627
126	2027	163.118891
127	2028	164.664547
128	2029	159.399007
129	2030	166.425217
130	2031	161.265778
131	2032	159.959674
132	2033	161.720156
133	2034	164.384597
134	2035	164.567775
135	2036	162.997627

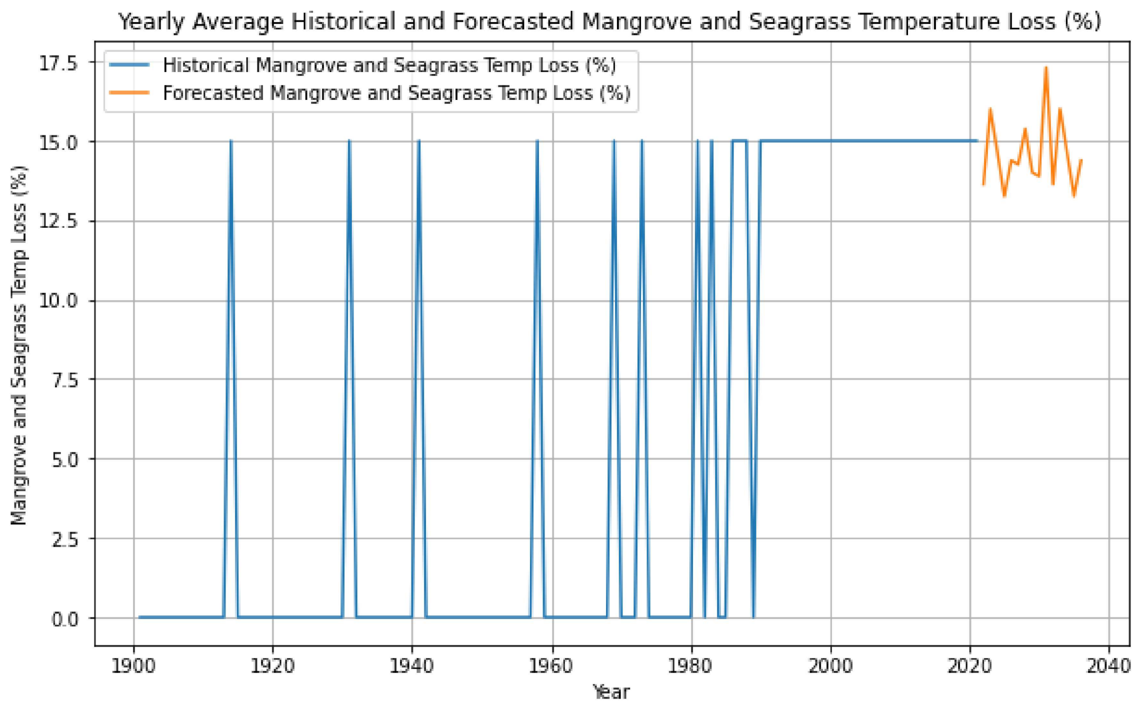
In [26]:

```
import pandas as pd
from statsmodels.tsa.holtwinters import ExponentialSmoothing
import matplotlib.pyplot as plt


#  DataFrame
data = {'Year': years,
        'Precipitation (mm)': seventeenth_values,
        'Temperature (°C)': temperatures,
        'Mangrove Gain (%)': mangrove_ranges,
        'Seagrass Loss (%)': seagrass_ranges,
        'Mangrove and Seagrass Temp Loss (%)': temperature_ranges}

df = pd.DataFrame(data)

 # ranges for Mangrove and Seagrass (Temperature Loss in %)
temperature_ranges = [15 if value > average_temperature else 0 for value in temperatures

# Exponential Smoothing for mangrove gain
model_mangrove_gain = ExponentialSmoothing(historical_mangrove_gain, seasonal='add', sea
model_fit_mangrove_gain = model_mangrove_gain.fit()
# Forecasted mangrove gain for next 15 years
forecasted_mangrove_gain = model_fit_mangrove_gain.forecast(steps=15)
# Exponential Smoothing for seagrass loss
model_seagrass_loss = ExponentialSmoothing(historical_seagrass_loss, seasonal='add', sea
model_fit_seagrass_loss = model_seagrass_loss.fit()
# Forecasted seagrass loss for next 15 years
forecasted_seagrass_loss = model_fit_seagrass_loss.forecast(steps=15)
# Exponential Smoothing for Mangrove and Seagrass Temp Loss (%)
model_temp_loss = ExponentialSmoothing(historical_temp_loss, seasonal='add', seasonal_pe
model_fit_temp_loss = model_temp_loss.fit()
# Forecasted Mangrove and Seagrass Temp Loss (%) for next 15 years
forecasted_temp_loss = model_fit_temp_loss.forecast(steps=15)


#  forecasted DataFrames
forecasted_data_mangrove_gain = {'Year': forecasted_years, 'Forecasted Mangrove Gain (%)
forecasted_df_mangrove_gain = pd.DataFrame(forecasted_data_mangrove_gain)

forecasted_data_seagrass_loss = {'Year': forecasted_years, 'Forecasted Seagrass Loss (%)
forecasted_df_seagrass_loss = pd.DataFrame(forecasted_data_seagrass_loss)

forecasted_data_temp_loss = {'Year': forecasted_years, 'Forecasted Mangrove and Seagrass
forecasted_df_temp_loss = pd.DataFrame(forecasted_data_temp_loss)

# historical and forecasted mangrove gain
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], historical_mangrove_gain, label='Historical Mangrove Gain')
plt.plot(forecasted_df_mangrove_gain['Year'], forecasted_df_mangrove_gain['Forecasted Ma
plt.xlabel('Year')
plt.ylabel('Mangrove Gain (%)')
plt.title('Yearly Average Historical and Forecasted Mangrove Precipitation Gain %')
plt.legend()
plt.grid()
plt.show()

#  historical and forecasted seagrass loss
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], historical_seagrass_loss, label='Historical Seagrass Loss')
```

```python
plt.plot(forecasted_df_seagrass_loss['Year'], forecasted_df_seagrass_loss['Forecasted Se
plt.xlabel('Year')
plt.ylabel('Seagrass Loss (%)')
plt.title('Yearly Average Historical and Forecasted Seagrass Precipitation Loss (%)')
plt.legend()
plt.grid()
plt.show()

#  historical and forecasted mangrove and seagrass temp loss
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], historical_temp_loss, label='Historical Mangrove and Seagrass Temp
plt.plot(forecasted_df_temp_loss['Year'], forecasted_df_temp_loss['Forecasted Mangrove a
plt.xlabel('Year')
plt.ylabel('Mangrove and Seagrass Temp Loss (%)')
plt.title('Yearly Average Historical and Forecasted Mangrove and Seagrass Temperature Lo
plt.legend()
plt.grid()
plt.show()
```




Yearly Average Historical and Forecasted Seagrass Precipitation Loss (%)
Historical Seagrass Loss
Forecasted Seagrass Loss
Seagrass Loss (%)
Year

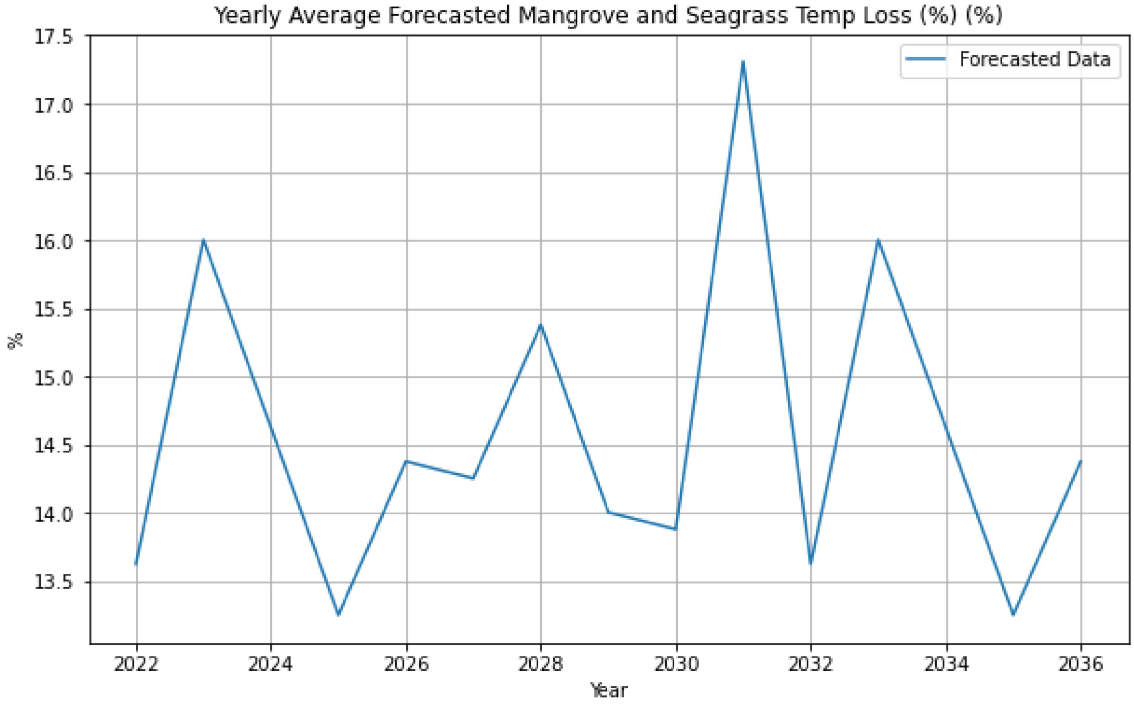

Yearly Average Historical and Forecasted Mangrove and Seagrass Temperature Loss (%)
Historical Mangrove and Seagrass Temp Loss (%)
Forecasted Mangrove and Seagrass Temp Loss (%)
Mangrove and Seagrass Temp Loss (%)
Year

In [35]:

```
import csv
import openpyxl
import pandas as pd
from statsmodels.tsa.holtwinters import ExponentialSmoothing
import matplotlib.pyplot as plt

# Read and preprocess data
file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter
seventeenth_values = []

with open(file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    next(csv_reader)
    next(csv_reader)
    for row in csv_reader:
        if len(row) > 17:
            seventeenth_value = float(row[17]) / 12  # Divide by 12 to convert to mm
            seventeenth_values.append(seventeenth_value)


#  ranges for Mangrove and Seagrass (Temperature Loss in %)
temperature_ranges = [15 if value > average_temperature else 0 for value in temperatures

#

#  DataFrame
data = {'Year': years,
        'Precipitation (mm)': seventeenth_values,
        'Temperature (°C)': temperatures,
        'Mangrove Gain (%)': mangrove_ranges,
        'Seagrass Loss (%)': seagrass_ranges,
        'Mangrove and Seagrass Temp Loss (%)': temperature_ranges}

df = pd.DataFrame(data)

# Exponential Smoothing for Mangrove and Seagrass Temp Loss (%)
model_temp_loss = ExponentialSmoothing(historical_temp_loss, seasonal='add', seasonal_pe
model_fit_temp_loss = model_temp_loss.fit()
# Forecasted Mangrove and Seagrass Temp Loss (%) for next 15 years
forecasted_temp_loss = model_fit_temp_loss.forecast(steps=15)
# Replace forecasted_years with the appropriate range of years
forecasted_years = list(range(2022, 2037))
# Create forecasted DataFrame for temperature loss
forecasted_data_temp_loss = {'Year': forecasted_years, 'Forecasted Mangrove and Seagrass
forecasted_df_temp_loss = pd.DataFrame(forecasted_data_temp_loss)

# Merge all forecasted DataFrames on Year
forecasted_df_combined = forecasted_df_temp_loss

# Display DataFrame as Excellike table
print("Combined Forecasted Data:")
print(forecasted_df_combined.to_string(index=False))

# forecasted temperature loss
for column in forecasted_df_combined.columns[1:]:
    plt.figure(figsize=(10, 6))
    plt.plot(forecasted_df_combined['Year'], forecasted_df_combined[column], label='Fore
    plt.xlabel('Year')
    plt.ylabel('%')
```

```python
    plt.title(f'Yearly Average {column} (%)')
    plt.legend()
    plt.grid()
    plt.show()
```

```
Combined Forecasted Data:
 Year  Forecasted Mangrove and Seagrass Temp Loss (%)
 2022                                       13.626783
 2023                                       16.002217
 2024                                       14.627695
 2025                                       13.253326
 2026                                       14.378796
 2027                                       14.254364
 2028                                       15.380049
 2029                                       14.005388
 2030                                       13.881279
 2031                                       17.308695
 2032                                       13.626783
 2033                                       16.002217
 2034                                       14.627695
 2035                                       13.253326
 2036                                       14.378796
```

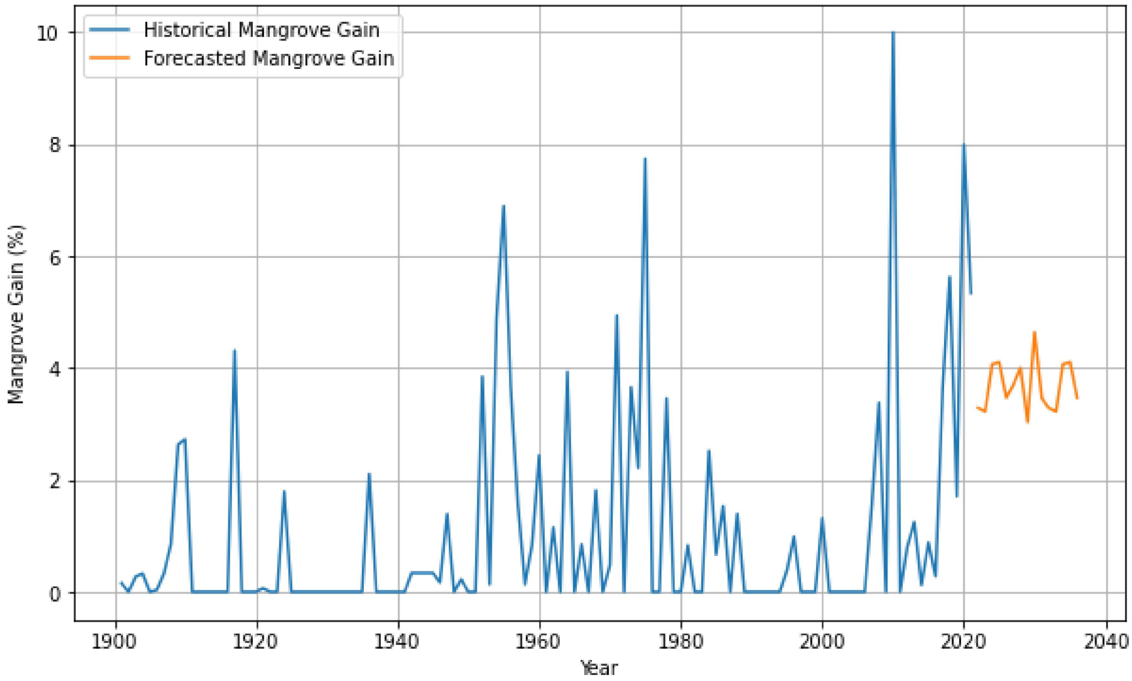


In [31]:

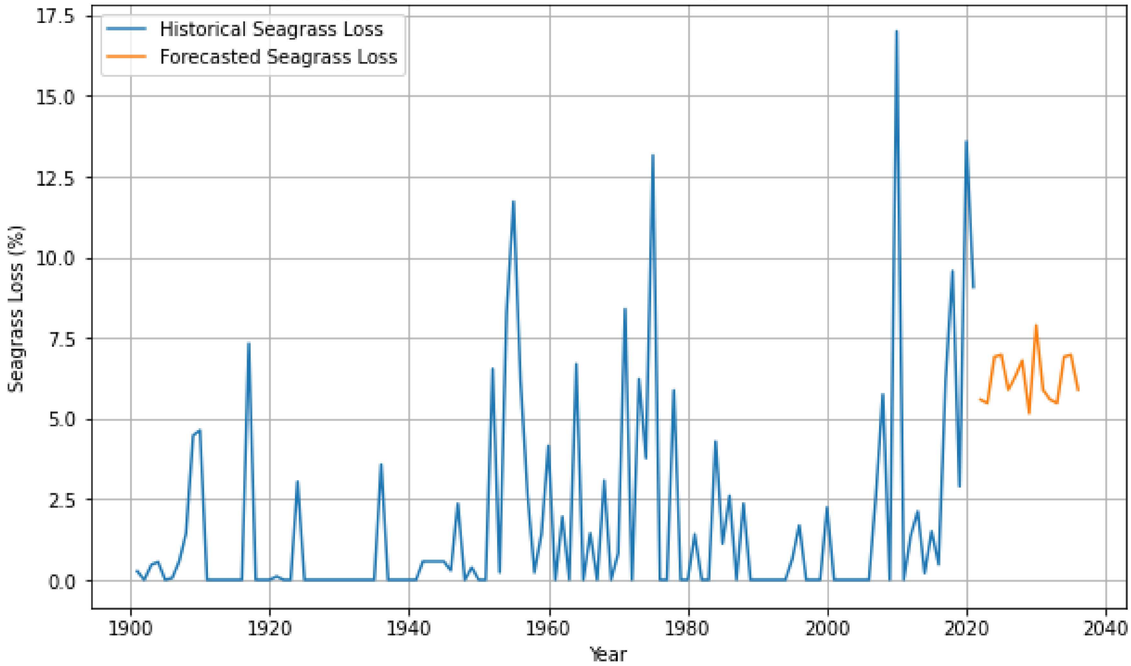
```
model_mangrove_gain = ExponentialSmoothing(historical_mangrove_gain, seasonal='add', sea
model_fit_mangrove_gain = model_mangrove_gain.fit()


forecasted_mangrove_gain = model_fit_mangrove_gain.forecast(steps=15)


forecasted_data_mangrove_gain = {'Year': forecasted_years, 'Forecasted Mangrove Gain (%)
forecasted_df_mangrove_gain = pd.DataFrame(forecasted_data_mangrove_gain)

plt.figure(figsize=(10, 6))
plt.plot(df['Year'], historical_mangrove_gain, label='Historical Mangrove Gain')
plt.plot(forecasted_df_mangrove_gain['Year'], forecasted_df_mangrove_gain['Forecasted Ma
plt.xlabel('Year')
plt.ylabel('Mangrove Gain (%)')
plt.title('Yearly Average Historical and Forecasted Mangrove Precipitation Gain %')
plt.legend()
plt.grid()
plt.show()

# Exponential Smoothing for seagrass loss
model_seagrass_loss = ExponentialSmoothing(historical_seagrass_loss, seasonal='add', sea
model_fit_seagrass_loss = model_seagrass_loss.fit()

# Forecasted seagrass loss for next 15 years
forecasted_seagrass_loss = model_fit_seagrass_loss.forecast(steps=15)

# forecasted seagrass loss Y2-Y15
forecasted_data_seagrass_loss = {'Year': forecasted_years, 'Forecasted Seagrass Loss (%)
forecasted_df_seagrass_loss = pd.DataFrame(forecasted_data_seagrass_loss)

plt.figure(figsize=(10, 6))
plt.plot(df['Year'], historical_seagrass_loss, label='Historical Seagrass Loss')
plt.plot(forecasted_df_seagrass_loss['Year'], forecasted_df_seagrass_loss['Forecasted Se
plt.xlabel('Year')
plt.ylabel('Seagrass Loss (%)')
plt.title('Yearly Average Historical and Forecasted Seagrass Precipitation Loss (%)')
plt.legend()
plt.grid()
plt.show()

#Exponential Smoothing for Mangrove and Seagrass Temp Loss (%)
model_temp_loss = ExponentialSmoothing(historical_temp_loss, seasonal='add', seasonal_pe
model_fit_temp_loss = model_temp_loss.fit()

# Forecasted Mangrove and Seagrass Temp Loss (%)
forecasted_temp_loss = model_fit_temp_loss.forecast(steps=15)

# forecasted Mangrove and Seagrass Temp Loss (%)
forecasted_data_temp_loss = {'Year': forecasted_years, 'Forecasted Mangrove and Seagrass
forecasted_df_temp_loss = pd.DataFrame(forecasted_data_temp_loss)

plt.figure(figsize=(10, 6))
plt.plot(df['Year'], historical_temp_loss, label='Historical Mangrove and Seagrass Temp
plt.plot(forecasted_df_temp_loss['Year'], forecasted_df_temp_loss['Forecasted Mangrove a
plt.xlabel('Year')
plt.ylabel('Mangrove and Seagrass Temp Loss (%)')
plt.title('Yearly Average Historical and Forecasted Mangrove and Seagrass Temperature Lo
plt.legend()
```

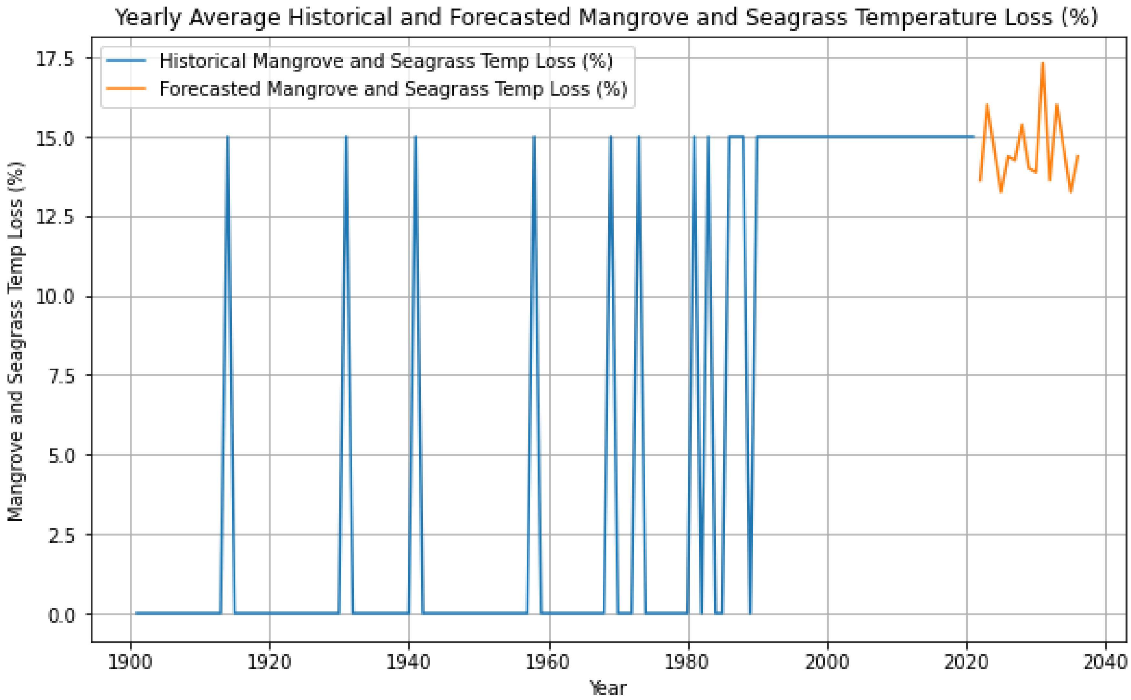
```
plt.grid()
plt.show()
```

Yearly Average Historical and Forecasted Mangrove Precipitation Gain %



Yearly Average Historical and Forecasted Seagrass Precipitation Loss (%)




Yearly Average Historical and Forecasted Mangrove and Seagrass Temperature Loss (%)
Historical Mangrove and Seagrass Temp Loss (%)
Forecasted Mangrove and Seagrass Temp Loss (%)
Mangrove and Seagrass Temp Loss (%)
Year
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
1900
1920
1940
1960
1980
2000
2020
2040

In [42]:

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from tabulate import tabulate

# Your data and model fitting code goes here...

# Forecasted mangrove gain for next 15 years
forecasted_mangrove_gain = model_fit_mangrove_gain.forecast(steps=15)
forecasted_data_mangrove_gain = {'Year': forecasted_years, 'Forecasted Mangrove Gain (%)
forecasted_df_mangrove_gain = pd.DataFrame(forecasted_data_mangrove_gain)

# Forecasted seagrass loss for next 15 yers
forecasted_seagrass_loss = model_fit_seagrass_loss.forecast(steps=15)
forecasted_data_seagrass_loss = {'Year': forecasted_years, 'Forecasted Seagrass Loss (%)
forecasted_df_seagrass_loss = pd.DataFrame(forecasted_data_seagrass_loss)

# Forecasted Mangrove and Seagrass Temp Loss (%)
forecasted_temp_loss = model_fit_temp_loss.forecast(steps=15)
forecasted_data_temp_loss = {'Year': forecasted_years, 'Forecasted Mangrove and Seagrass
forecasted_df_temp_loss = pd.DataFrame(forecasted_data_temp_loss)


forecasted_table_mangrove_gain = tabulate(forecasted_df_mangrove_gain, headers='keys', t
forecasted_table_seagrass_loss = tabulate(forecasted_df_seagrass_loss, headers='keys', t
forecasted_table_temp_loss = tabulate(forecasted_df_temp_loss, headers='keys', tablefmt=


for df, table_title in [(forecasted_df_mangrove_gain, 'Forecasted Mangrove Gain'), (fore
    print(table_title + ":")
    print(tabulate(df, headers='keys', tablefmt='grid'))

    plt.figure(figsize=(10, 6))
    plt.plot(df['Year'], df.iloc[:, 1], label='Forecasted Data')
    plt.xlabel('Year')
    plt.ylabel('%')
    plt.title(f'Yearly Average {table_title} (%)')
    plt.legend()
    plt.grid()
    plt.show()




import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from tabulate import tabulate

.


forecasted_temp_loss = model_fit_temp_loss.forecast(steps=15)


forecasted_years = list(range(2022, 2037))


forecasted_data_temp_loss = {'Year': forecasted_years, 'Forecasted Mangrove and Seagrass
```

```
forecasted_df_temp_loss = pd.DataFrame(forecasted_data_temp_loss)


forecasted_table_temp_loss = tabulate(forecasted_df_temp_loss, headers='keys', tablefmt=


print("Combined Forecasted Data:")
print(forecasted_table_temp_loss)

# Plotting forecasted temperature loss
for column in forecasted_df_temp_loss.columns[1:]:
    plt.figure(figsize=(10, 6))
    plt.plot(forecasted_df_temp_loss['Year'], forecasted_df_temp_loss[column], label='Fo
    plt.xlabel('Year')
    plt.ylabel('%')
    plt.title(f'Yearly Average {column} (%)')
    plt.legend()
    plt.grid()
    plt.show()



import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from tabulate import tabulate



forecasted_seagrass_loss = model_fit_seagrass_loss.forecast(steps=15)
forecasted_data_seagrass_loss = {'Year': forecasted_years, 'Forecasted Seagrass Loss (%)
forecasted_df_seagrass_loss = pd.DataFrame(forecasted_data_seagrass_loss)


forecasted_mangrove_gain = model_fit_mangrove_gain.forecast(steps=15)
forecasted_data_mangrove_gain = {'Year': forecasted_years, 'Forecasted Mangrove Gain (%)
forecasted_df_mangrove_gain = pd.DataFrame(forecasted_data_mangrove_gain)

forecasted_table_seagrass_loss = tabulate(forecasted_df_seagrass_loss, headers='keys', t


print("Forecasted Seagrass Loss Data:")
print(forecasted_table_seagrass_loss)

forecasted_table_mangrove_gain = tabulate(forecasted_df_mangrove_gain, headers='keys', t


print("Forecasted Mangrove Gain Data:")
print(forecasted_table_mangrove_gain)

#  table for forecasted temperature loss
forecasted_table_temp_loss = tabulate(forecasted_df_temp_loss, headers='keys', tablefmt=


print("Forecasted Temperature Loss Data:")
print(forecasted_table_temp_loss)


for column in forecasted_df_seagrass_loss.columns[1:]:
    plt.figure(figsize=(10, 6))
    plt.plot(forecasted_df_seagrass_loss['Year'], forecasted_df_seagrass_loss[column], l
```

```python
    plt.xlabel('Year')
    plt.ylabel('%')
    plt.title(f'Yearly Average {column} (%)')
    plt.legend()
    plt.grid()
    plt.show()

for column in forecasted_df_mangrove_gain.columns[1:]:
    plt.figure(figsize=(10, 6))
    plt.plot(forecasted_df_mangrove_gain['Year'], forecasted_df_mangrove_gain[column], l
    plt.xlabel('Year')
    plt.ylabel('%')
    plt.title(f'Yearly Average {column} (%)')
    plt.legend()
    plt.grid()
    plt.show()


for column in forecasted_df_temp_loss.columns[1:]:
    plt.figure(figsize=(10, 6))
    plt.plot(forecasted_df_temp_loss['Year'], forecasted_df_temp_loss[column], label='Fo
    plt.xlabel('Year')
    plt.ylabel('%')
    plt.title(f'Yearly Average {column} (%)')
    plt.legend()
    plt.grid()
    plt.show()
```

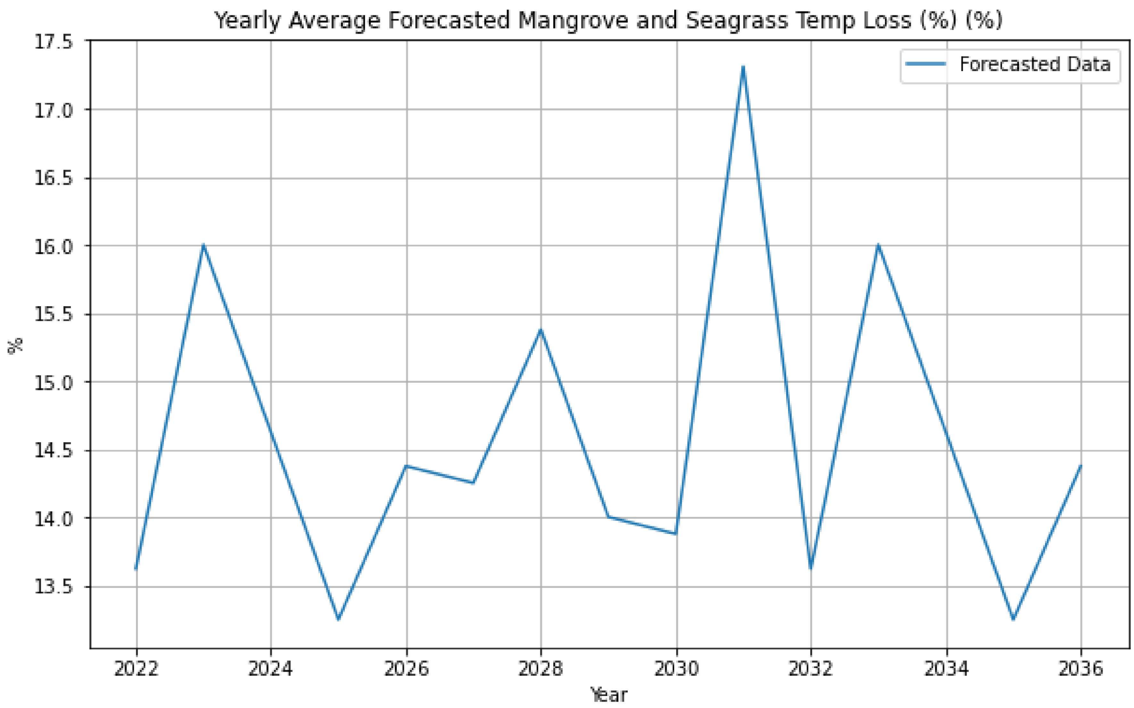
Forecasted Mangrove Gain:

	Year	Forecasted Mangrove Gain (%)
121	2022	3.2881
122	2023	3.21999
123	2024	4.06676
124	2025	4.10368
125	2026	3.46939
126	2027	3.7044
127	2028	4.00261
128	2029	3.03715

In [43]:

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from tabulate import tabulate



# Forecasted Mangrove and Seagrass Temp Loss (%) for next 15 years
forecasted_temp_loss = model_fit_temp_loss.forecast(steps=15)

# Replace forecasted_years with the appropriate range of years
forecasted_years = list(range(2022, 2037))

# Create forecasted DataFrame for temperature loss
forecasted_data_temp_loss = {'Year': forecasted_years, 'Forecasted Mangrove and Seagrass
forecasted_df_temp_loss = pd.DataFrame(forecasted_data_temp_loss)

# Extract and Save Forecasted Temperature Loss Data to Excel
forecasted_df_temp_loss.to_excel("forecasted_temperature_loss.xlsx", index=False)

# Display the combined DataFrame as an Excel-like table
print("Combined Forecasted Data:")
print(tabulate(forecasted_df_temp_loss, headers='keys', tablefmt='pretty'))

# Plotting forecasted temperature loss
for column in forecasted_df_temp_loss.columns[1:]:
    plt.figure(figsize=(10, 6))
    plt.plot(forecasted_df_temp_loss['Year'], forecasted_df_temp_loss[column], label='Fo
    plt.xlabel('Year')
    plt.ylabel('%')
    plt.title(f'Yearly Average {column} (%)')
    plt.legend()
    plt.grid()
    plt.show()
```

```
Combined Forecasted Data:
+-----+--------+-----------------------------------------------+
|     |  Year  | Forecasted Mangrove and Seagrass Temp Loss (%) |
+-----+--------+-----------------------------------------------+
| 121 | 2022.0 |               13.626782569783401               |
| 122 | 2023.0 |               16.002216884447726               |
| 123 | 2024.0 |               14.627695497710176               |
| 124 | 2025.0 |               13.253326050938778               |
| 125 | 2026.0 |               14.37879606894401                |
| 126 | 2027.0 |               14.254364495337047               |
| 127 | 2028.0 |               15.380049063018497               |
| 128 | 2029.0 |               14.005388249747098               |
| 129 | 2030.0 |               13.881279443685298               |
| 130 | 2031.0 |               17.308695016602957               |
| 131 | 2032.0 |               13.626782569783401               |
| 132 | 2033.0 |               16.002216884447726               |
| 133 | 2034.0 |               14.627695497710176               |
| 134 | 2035.0 |               13.253326050938778               |
| 135 | 2036.0 |               14.37879606894401                |
+-----+--------+-----------------------------------------------+
```


Yearly Average Forecasted Mangrove and Seagrass Temp Loss (%) (%)
Forecasted Data
17.5
17.0
16.5
16.0
15.5
15.0
14.5
14.0
13.5
%
2022
2024
2026
2028
2030
2032
2034
2036
Year

In [44]:

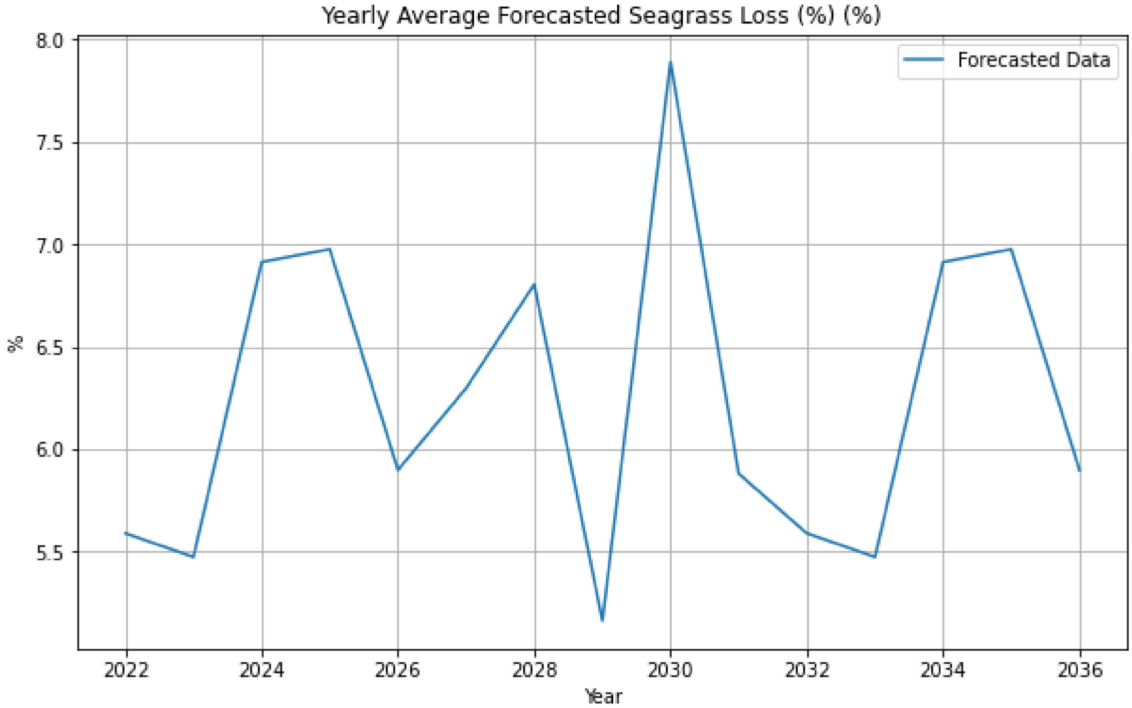
```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from tabulate import tabulate

# Forecasted mangrove gain for next 15 years
forecasted_mangrove_gain = model_fit_mangrove_gain.forecast(steps=15)
forecasted_data_mangrove_gain = {'Year': forecasted_years, 'Forecasted Mangrove Gain (%)
forecasted_df_mangrove_gain = pd.DataFrame(forecasted_data_mangrove_gain)

# Forecasted seagrass loss for next 15 years
forecasted_seagrass_loss = model_fit_seagrass_loss.forecast(steps=15)
forecasted_data_seagrass_loss = {'Year': forecasted_years, 'Forecasted Seagrass Loss (%)
forecasted_df_seagrass_loss = pd.DataFrame(forecasted_data_seagrass_loss)

# Extract and Save Forecasted Mangrove Gain Data to Excel
forecasted_df_mangrove_gain.to_excel("forecasted_mangrove_gain.xlsx", index=False)

# Display the forecasted mangrove gain table
print("Forecasted Mangrove Gain Data:")
print(tabulate(forecasted_df_mangrove_gain, headers='keys', tablefmt='pretty'))
# Extract and Save Forecasted Seagrass Loss Data to Excel
forecasted_df_seagrass_loss.to_excel("forecasted_seagrass_loss.xlsx", index=False)
# Display the forecasted seagrass loss table
print("Forecasted Seagrass Loss Data:")
print(tabulate(forecasted_df_seagrass_loss, headers='keys', tablefmt='pretty'))
# Plotting forecasted seagrass loss
for column in forecasted_df_seagrass_loss.columns[1:]:
    plt.figure(figsize=(10, 6))
    plt.plot(forecasted_df_seagrass_loss['Year'], forecasted_df_seagrass_loss[column], l
    plt.xlabel('Year')
    plt.ylabel('%')
    plt.title(f'Yearly Average {column} (%)')
    plt.legend()
    plt.grid()
    plt.show()
# Plotting forecasted mangrove gain
for column in forecasted_df_mangrove_gain.columns[1:]:
    plt.figure(figsize=(10, 6))
    plt.plot(forecasted_df_mangrove_gain['Year'], forecasted_df_mangrove_gain[column], l
    plt.xlabel('Year')
    plt.ylabel('%')
    plt.title(f'Yearly Average {column} (%)')
    plt.legend()
    plt.grid()
    plt.show()
```

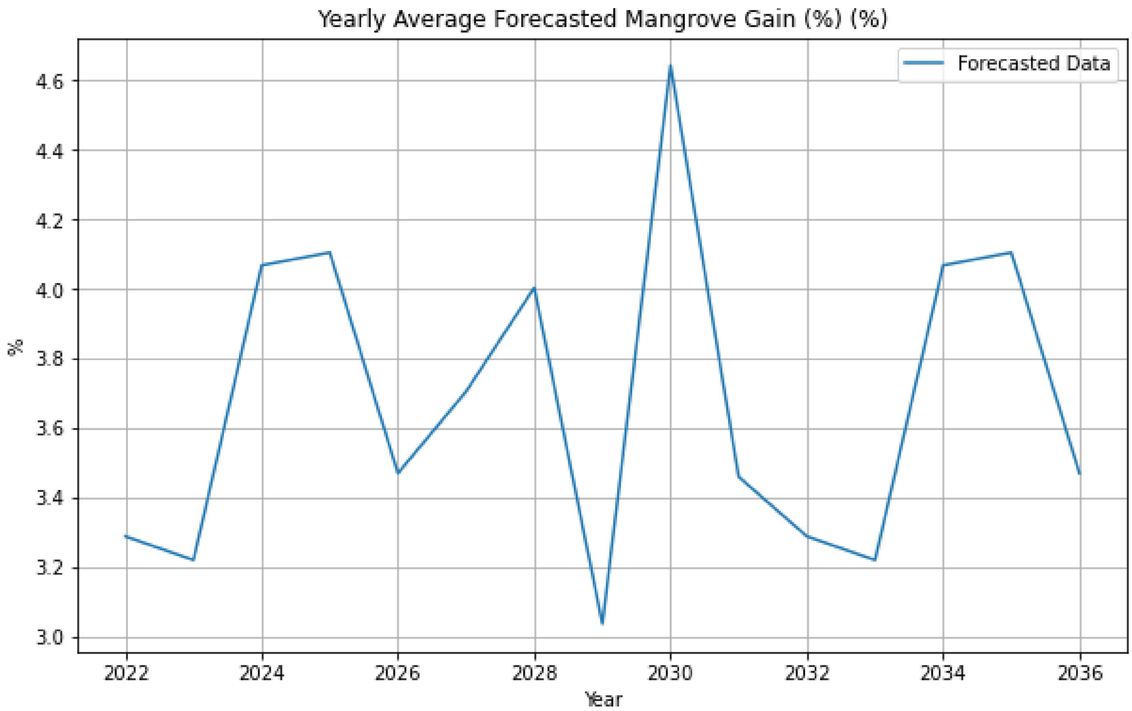
Forecasted Mangrove Gain Data:

	Year	Forecasted Mangrove Gain (%)
121	2022.0	3.288095220965223
122	2023.0	3.219993396789673
123	2024.0	4.066760969500409
124	2025.0	4.103684047110404
125	2026.0	3.4693864445338245
126	2027.0	3.704402651183429
127	2028.0	4.002605019536132
128	2029.0	3.037151553585807
129	2030.0	4.640949461751387
130	2031.0	3.4591415957662606
131	2032.0	3.288095220965223
132	2033.0	3.219993396789673
133	2034.0	4.066760969500409
134	2035.0	4.103684047110404
135	2036.0	3.4693864445338245

Forecasted Seagrass Loss Data:

	Year	Forecasted Seagrass Loss (%)
121	2022.0	5.588709038510648
122	2023.0	5.473714436249247
123	2024.0	6.913443161188546
124	2025.0	6.975623485618811
125	2026.0	5.897713299406815
126	2027.0	6.297175670891295
127	2028.0	6.804269335413683
128	2029.0	5.1628465925325795
129	2030.0	7.889298365817967
130	2031.0	5.8801424429431295
131	2032.0	5.588709038510648
132	2033.0	5.473714436249247
133	2034.0	6.913443161188546
134	2035.0	6.975623485618811
135	2036.0	5.897713299406815


Yearly Average Forecasted Seagrass Loss (%) (%)
Forecasted Data
8.0
7.5
7.0
6.5
6.0
5.5
%
2022
2024
2026
2028
2030
2032
2034
2036
Year


Yearly Average Forecasted Mangrove Gain (%) (%)
Forecasted Data
4.6
4.4
4.2
4.0
3.8
3.6
3.4
3.2
3.0
%
2022
2024
2026
2028
2030
2032
2034
2036
Year

In [45]:

```
import csv
import matplotlib.pyplot as plt
import openpyxl
import pandas as pd
from statsmodels.tsa.holtwinters import ExponentialSmoothing

.

# Forecasted temperature
forecasted_temperatures = model_fit_temp.forecast(steps=15)

# DataFrame forecasted temperatures Y2-Y15
forecasted_years = list(range(2022, 2037))
forecasted_data_temp = {'Year': forecasted_years, 'Forecasted Temperature (°C)': forecas
forecasted_df_temp = pd.DataFrame(forecasted_data_temp)

# Extract and Save Forecasted Temperature on excel
forecasted_df_temp.to_excel("forecasted_temperature.xlsx", index=False)

# Display forecast temperatures in a table
print("Forecasted Temperatures:")
print(forecasted_df_temp)
# Fit Exponential Smoothing for precipitation
model_precip = ExponentialSmoothing(historical_precipitation, seasonal='add', seasonal_p
model_fit_precip = model_precip.fit()
# Forecasted precipitation for next 15 years
forecasted_precipitation = model_fit_precip.forecast(steps=15)
# DataFrame for forecasted precipitation Y2-Y15
forecasted_data_precip = {'Year': forecasted_years, 'Forecasted Precipitation (mm)': for
forecasted_df_precip = pd.DataFrame(forecasted_data_precip)
# Extract and Save Forecasted Precipitation Data to Excel
forecasted_df_precip.to_excel("forecasted_precipitation.xlsx", index=False)
print("\nForecasted Precipitation:")
print(forecasted_df_precip)
```

```
Forecasted Temperatures:
     Year  Forecasted Temperature (°C)
106  2022                    25.913807
107  2023                    25.956468
108  2024                    25.904194
109  2025                    25.940828
110  2026                    25.950688
111  2027                    25.929520
112  2028                    25.935549
113  2029                    25.857190
114  2030                    25.878596
115  2031                    25.875588
116  2032                    25.913807
117  2033                    25.956468
118  2034                    25.904194
119  2035                    25.940828
120  2036                    25.950688

Forecasted Precipitation:
     Year  Forecasted Precipitation (mm)
121  2022                     159.959674
122  2023                     161.720156
123  2024                     164.384597
124  2025                     164.567775
125  2026                     162.997627
126  2027                     163.118891
127  2028                     164.664547
128  2029                     159.399007
129  2030                     166.425217
130  2031                     161.265778
131  2032                     159.959674
132  2033                     161.720156
133  2034                     164.384597
134  2035                     164.567775
135  2036                     162.997627
```

In [ ]:

```

```