In [128]:

```
import pandas as pd


excel_data = pd.read_excel('C:/Users/Rishab/Documents/UCL/Masters Thesis/Relevant journa


print(excel_data.head())

print(excel_data.columns)
```

```
        Date  PX_LAST  SpotPriceUSD  \
0 2021-12-31    68.20       40.2380
1 2021-12-30    68.15       40.2085
2 2021-12-29    68.20       40.2380
3 2021-12-28    68.42       40.3678
4 2021-12-27    68.42       40.3678

   Lack of trade awareness and monitoring in the secondary market,  \
0                                               0.19
1                                                NaN
2                                                NaN
3                                                NaN
4                                                NaN

   Credit and counterparty risk  \
0                         0.177
1                           NaN
2                           NaN
3                           NaN
4                           NaN

   Insider trading and information asymmetry, Manipulation of NZU prices
\
0                                              0.161
1                                                NaN
2                                                NaN
3                                                NaN
4                                                NaN

   Money Laundering and Financing of terrorism in the NZU market  \
0                                              0.175
1                                                NaN
2                                                NaN
3                                                NaN
4                                                NaN

   Greenwashing  Technical feasibility risk  Date by Year  ...  \
0         0.166                       0.131        2009.0  ...
1           NaN                         NaN        2010.0  ...
2           NaN                         NaN        2011.0  ...
3           NaN                         NaN        2012.0  ...
4           NaN                         NaN        2013.0  ...

   Risk weights (Carbon market)  Risk Weights (BCS) Unnamed: 15  Unnamed:
16  \
0                         0.190               0.04         NaN          N
aN
1                         0.177               0.20         NaN          N
aN
2                         0.161               0.07         NaN          N
aN
3                         0.175               0.68         NaN          N
aN
4                         0.166               0.01         NaN          N
aN

   Unnamed: 17     (USD)=  0.59 Unnamed: 20    Security  \
0          NaN        NaN   NaN         NaN         NaN
1          Max  40.415000   NaN         NaN  Start Date
2          Min   0.855500   NaN         NaN    End Date
3       Median  10.944500   NaN         NaN      Period
```

```
4      Average  10.712395   NaN         NaN    Currency

                        NZUSSPOT JRDN Index
0                                       NaN
1                       2009-03-09 00:00:00
2                       2023-07-24 00:00:00
3                                         D
4  NZD (converted to USD: 1 NZD = 0.59 USD)

[5 rows x 23 columns]
Index([                                                                  'D
ate',
                                                                      'PX_L
AST',
                                                                   'SpotPrice
USD',
             'Lack of trade awareness and monitoring in the secondary mark
et,',
                                                     'Credit and counterparty r
isk',
       'Insider trading and information asymmetry, Manipulation of NZU pri
ces',
               'Money Laundering and Financing of terrorism in the NZU mar
ket',
                                                                   'Greenwash
ing',
                                                   'Technical feasibility r
isk',
                                                                'Date by Y
ear',
                                                    'Yearly average spot pr
ice',
                                                                 'Unnamed:
11',
                                                       'Risks  (Carbon mark
et)',
                                                  'Risk weights (Carbon mark
et)',
                                                        'Risk Weights (B
CS)',
                                                                 'Unnamed:
15',
                                                                 'Unnamed:
16',
                                                                 'Unnamed:
17',
                                                                      '(US
D)=',

0.59,
                                                                 'Unnamed:
20',
                                                                    'Secur
ity',
                                                          'NZUSSPOT JRDN In
dex'],
      dtype='object')
```

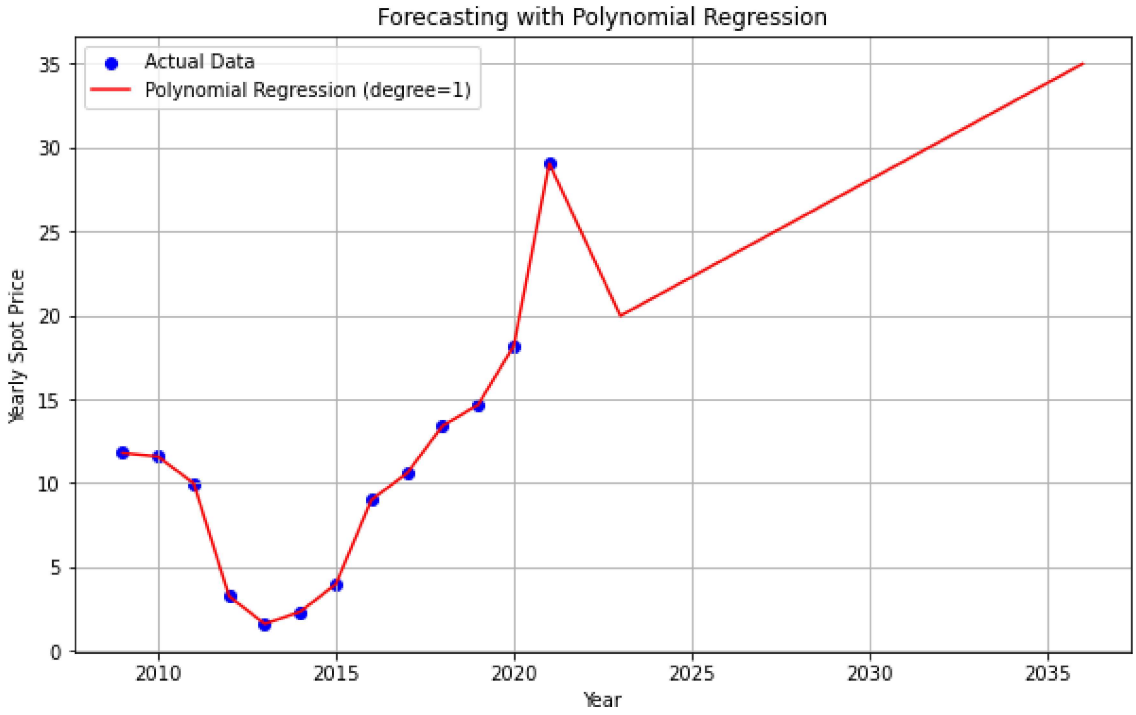
In [96]:

```
import pandas as pd


excel_data = pd.read_excel('C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant


print(excel_data.head())

print(excel_data.columns)
```

```
   Date by Year  Yearly average spot price
0          2009                  11.800000
1          2010                  11.577421
2          2011                   9.993488
3          2012                   3.255738
4          2013                   1.619968
Index(['Date by Year', 'Yearly average spot price'], dtype='object')
```

In [156]:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures

# Yearly data from 2009 to 2021
years = np.array([2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020
values = np.array([11.800000, 11.577421, 9.993488, 3.255738, 1.619968, 2.346143, 3.97978

# Reshaping the data for modeling
years = years.reshape(-1, 1)
values = values.reshape(-1, 1)

# polynomial features
degree = 1  # the degree of the polynomial
poly_features = PolynomialFeatures(degree=degree)
X_poly = poly_features.fit_transform(years)

# Fit for polynomial regression model
model = LinearRegression()
model.fit(X_poly, values)

# Forecast for years 2023 to 2036
forecast_years = np.arange(2023, 2037).reshape(-1, 1)
forecast_poly = poly_features.transform(forecast_years)
forecast_values = model.predict(forecast_poly)

# Plotting
plt.figure(figsize=(10, 6))
plt.scatter(years, values, color='blue', label='Actual Data')
plt.plot(np.concatenate((years, forecast_years)), np.concatenate((values, forecast_value
         color='red', label=f'Polynomial Regression (degree={degree})')
plt.xlabel('Year')
plt.ylabel('Yearly Spot Price')
plt.title('Forecasting with Polynomial Regression')
plt.legend()
plt.grid()
plt.show()

yearly_data = pd.DataFrame({'Year': years.flatten(), 'Value': values.flatten()})
forecasted_yearly_data = pd.DataFrame({'Year': forecast_years.flatten(), 'Forecasted Val

print("Yearly Actual and Forecasted Values:")
print(yearly_data)
print(forecasted_yearly_data)
```



Yearly Actual and Forecasted Values:

	Year	Value
0	2009	11.800000
1	2010	11.577421
2	2011	9.993488
3	2012	3.255738
4	2013	1.619968
5	2014	2.346143
6	2015	3.979787
7	2016	9.009503
8	2017	10.584600
9	2018	13.407377
10	2019	14.665834
11	2020	18.163218
12	2021	29.042287

	Year	Forecasted Value
0	2023	19.962594
1	2024	21.117098
2	2025	22.271601
3	2026	23.426105
4	2027	24.580608
5	2028	25.735112
6	2029	26.889615
7	2030	28.044119
8	2031	29.198622
9	2032	30.353125
10	2033	31.507629
11	2034	32.662132
12	2035	33.816636
13	2036	34.971139

In [214]:

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

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

df = pd.read_excel(file_path)

risk_1 = 0.190
risk_2 = 0.177
risk_3 = 0.161
risk_4 = 0.175
risk_5 = 0.166
risk_6 = 0.131
risk_7 = 0.04
risk_8 = 0.20
risk_9 = 0.07
risk_10 = 0.68
risk_11 = 0.01

total_risk_effect = 0.275


# the expected value for each scenario
df["Expected_Value_3"] = df["New total carbon revenue, after temperature loss. Experimen
df["Expected_Value_4"] = df["New total carbon revenue, after precipitation gain Mangrove
df["Expected_Value_5"] = df["New total carbon revenue, after precipitation loss Seagrass
df["Expected_Value_6"] = df["New total carbon revenue, after Lidar gain Mangroves 50% ma

#  overall expected value for each row
df["Overall_Expected_Value"] = df["Annual Carbon Credits Revenue (USD)"]-df["Expected_Va

# risk adjusted revenue
df["Risk_Number_1"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_1*
df["Risk_Number_2"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_2*
df["Risk_Number_3"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_3*
df["Risk_Number_4"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_4*
df["Risk_Number_5"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_5*
df["Risk_Number_6"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_6*
df["Risk_Number_7"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_7*
df["Risk_Number_8"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_8*
df["Risk_Number_9"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_9*
df["Risk_Number_10"] =df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_10
df["Risk_Number_11"] =df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_11
df["Risk_Number_all"] =df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*total_


# the percentage change from the initial value
initial_value = df["Risk_Number_all"].iloc[0]
df["Percent_Change"] = ((df["Risk_Number_all"] - initial_value) / initial_value) * 100

#  DataFrame with calculated expected values
print(df)


plt.figure(figsize=(10, 6))
plt.plot(df["Risk_Number_1"], label="Risk 1")
plt.plot(df["Risk_Number_2"], label="Risk 2")
plt.plot(df["Risk_Number_3"], label="Risk 3")
```

```
plt.plot(df["Risk_Number_4"], label="Risk 4")
plt.plot(df["Risk_Number_5"], label="Risk 5")
plt.plot(df["Risk_Number_6"], label="Risk 6")
plt.plot(df["Risk_Number_7"], label="Risk 7")
plt.plot(df["Risk_Number_8"], label="Risk 8")
plt.plot(df["Risk_Number_9"], label="Risk 9")
plt.plot(df["Risk_Number_10"], label="Risk 10")
plt.plot(df["Risk_Number_11"], label="Risk 11")
plt.plot(df["Risk_Number_all"], label="All Risk")
plt.title("Risk Adjusted Revenue Forecast")

plt.xlabel("Year")
plt.ylabel("Revenue per year")
plt.legend()
plt.show()
```

07e+05

	Risk_Number_10	Risk_Number_11	Risk_Number_all	Percent_Change
0	21479.368123	2.634723e+04	19154.418068	0.000000
1	321687.091875	3.945910e+05	286867.332852	1397.656216
2	861131.421949	1.056289e+06	767921.624740	3909.109658
3	849973.612340	1.042603e+06	757971.548520	3857.163020
4	843487.717845	1.034647e+06	752187.694265	3826.967092
5	826139.787222	1.013368e+06	736717.522431	3746.201539
6	837333.586049	1.027098e+06	746699.692356	3798.315729
7	822315.029167	1.008676e+06	733306.760327	3728.394879
8	791283.885709	9.706124e+05	705634.461426	3583.925343
9	821552.872078	1.007741e+06	732627.099947	3724.846557
10	791804.387899	9.712508e+05	706098.623895	3586.348608
11	793397.495105	9.732050e+05	707519.291453	3593.765527
12	801764.018425	9.834676e+05	714980.213233	3632.716967
13	788098.729130	9.667054e+05	702794.069643	3569.096431

[14 rows x 60 columns]

In [216]:

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

df = pd.read_excel(file_path)

risk_1 = 0.190
risk_2 = 0.177
risk_3 = 0.161
risk_4 = 0.175
risk_5 = 0.166
risk_6 = 0.131
risk_7 = 0.04
risk_8 = 0.20
risk_9 = 0.07
risk_10 = 0.68
risk_11 = 0.01

total_risk_effect = 0.275

# overall expected value for each year (spot price and alternative data adjusted)
df["Overall_Expected_Value"] = df["Expected Value Revenue_yearly spot price adjusted"]

# risk adjusted revenue
df["Risk_Number_1"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_1*
df["Risk_Number_2"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_2*
df["Risk_Number_3"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_3*
df["Risk_Number_4"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_4*
df["Risk_Number_5"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_5*
df["Risk_Number_6"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_6*
df["Risk_Number_7"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_7*
df["Risk_Number_8"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_8*
df["Risk_Number_9"] = df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_9*
df["Risk_Number_10"] =df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_10
df["Risk_Number_11"] =df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*risk_11
df["Risk_Number_all"] =df["Overall_Expected_Value"]-(df["Overall_Expected_Value"]*total_


# percentage change from the initial value
initial_value = df["Risk_Number_all"].iloc[0]
df["Percent_Change"] = ((df["Risk_Number_all"] - initial_value) / initial_value) * 100


print(df)


plt.figure(figsize=(10, 6))
plt.plot(df["Risk_Number_1"], label="Risk 1")
plt.plot(df["Risk_Number_2"], label="Risk 2")
plt.plot(df["Risk_Number_3"], label="Risk 3")
plt.plot(df["Risk_Number_4"], label="Risk 4")
plt.plot(df["Risk_Number_5"], label="Risk 5")
plt.plot(df["Risk_Number_6"], label="Risk 6")
plt.plot(df["Risk_Number_7"], label="Risk 7")
plt.plot(df["Risk_Number_8"], label="Risk 8")
plt.plot(df["Risk_Number_9"], label="Risk 9")
plt.plot(df["Risk_Number_10"], label="Risk 10")
plt.plot(df["Risk_Number_11"], label="Risk 11")
plt.plot(df["Risk_Number_all"], label="All Risk")
plt.title("Risk and Spot Price Adjusted Revenue Forecast")
```

```
plt.xlabel("Year")
plt.ylabel("Revenue per year")
plt.legend()
plt.show()
```

```
13   1.107686e+06   1.118858e+06   1.147904e+06   1.096834e+06   1.1383
28e+06
```

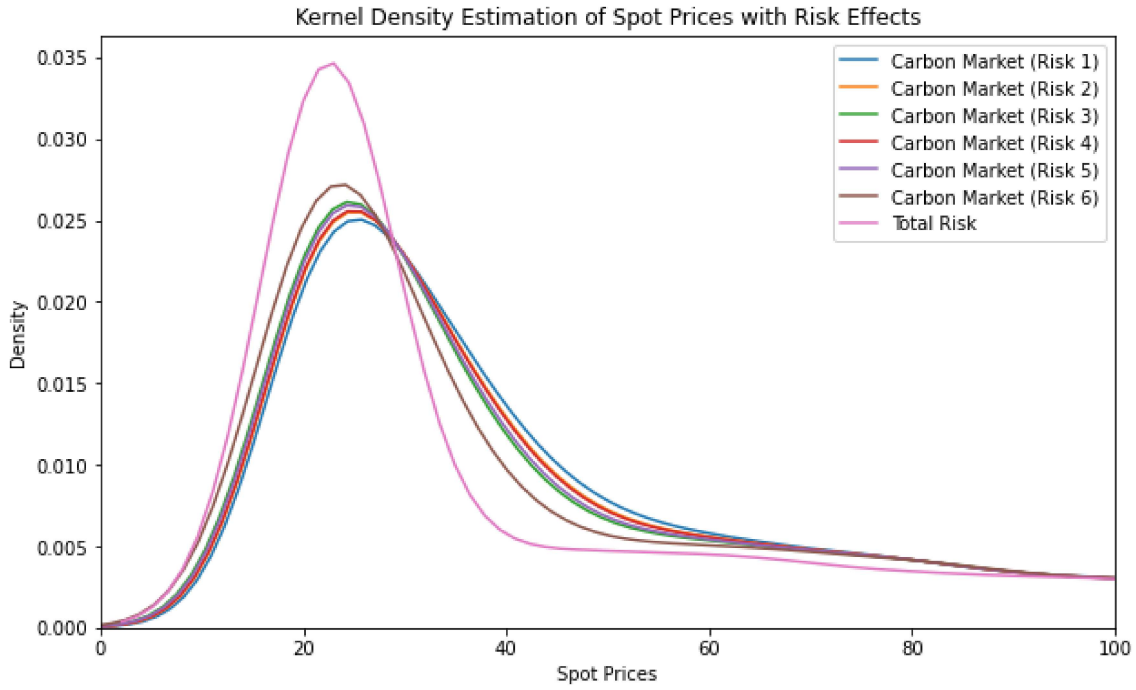
	Risk_Number_10	Risk_Number_11	Risk_Number_all	Percent_Change
0	7219.348123	8.855467e+03	6437.918068	0.000000
1	207176.041875	2.541283e+05	184751.082852	2769.733366
2	603654.321949	7.404604e+05	538314.124740	8261.618136
3	630219.712340	7.730463e+05	562004.048520	8629.593054
4	661131.817845	8.109640e+05	589570.194265	9057.777220
5	681181.887222	8.355580e+05	607450.022431	9335.504087
6	730098.886049	8.955610e+05	651072.192356	10013.086024
7	752478.329167	9.230123e+05	671029.260327	10323.078598
8	759170.385709	9.312210e+05	676996.961426	10415.774731
9	826837.372078	1.014223e+06	737339.599947	11353.075236
10	834812.087899	1.024005e+06	744451.123895	11463.538337
11	873803.195105	1.071833e+06	779221.791453	12003.630137
12	919892.918425	1.128368e+06	820322.713233	12642.049597
13	943625.629130	1.157479e+06	841486.569643	12970.787182

[14 rows x 56 columns]

In [192]:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd


excel_data = pd.read_excel('C:/Users/Rishab/Documents/UCL/Masters Thesis/Relevant journa
historical_spot_prices = excel_data['SpotPriceUSD'][:262].tolist()
risk_weight_values = excel_data[['Lack of trade awareness and monitoring in the secondar

#  risk weight "Total Risk" with a weight of 1
risk_weight_values = np.append(risk_weight_values, 1.0)

# Number of simulations (scenario)
num_simulations = 10000

#  array to store simulated scenarios
simulated_spot_prices = np.zeros((num_simulations, len(historical_spot_prices)))
risk_effects = np.zeros((num_simulations, len(historical_spot_prices), len(risk_weight_v

#spot price affected by the risk weights
risk_effect_percentage = 0.275

# MC Simulation
for i in range(num_simulations):
    risk_adjustments = np.random.normal(size=len(risk_weight_values))

    # Adjusting spot prices and individual risk effects for each historical day
    adjusted_spot_prices = np.array(historical_spot_prices)
    for day in range(len(historical_spot_prices)):
        # Total risk effect for  day
        total_risk_effect = np.dot(risk_weight_values, risk_adjustments) * (day + 1) * r

        # Distributing total risk effect amongindividual risk weights
        for risk_idx in range(len(risk_weight_values)):
            #  the risk effect for this risk weight
            risk_effect = total_risk_effect * risk_weight_values[risk_idx]

            # Subtractrisk effect for all risk weights except "Total Risk"
            if risk_idx != len(risk_weight_values) - 1:
                adjusted_spot_prices[day] -= risk_effect

            risk_effects[i, day, risk_idx] = risk_effect

            #  adjusted spot price doesn't go below 72.5% of the original spot price
            min_spot_price = historical_spot_prices[day] * 0.725
            adjusted_spot_prices[day] = max(adjusted_spot_prices[day], min_spot_price)

        simulated_spot_prices[i, day] = adjusted_spot_prices[day]

# KDE plot to visualize distribution of simulated spot prices for each risk
plt.figure(figsize=(10, 6))
for risk_idx in range(len(risk_weight_values)):
    label = 'Total Risk' if risk_idx == len(risk_weight_values) - 1 else f'Carbon Market
    sns.kdeplot(simulated_spot_prices[:, -1] + risk_effects[:, -1, risk_idx], label=labe

# x-axis limits to match the actual spot price range
plt.xlim(0, 100)
```

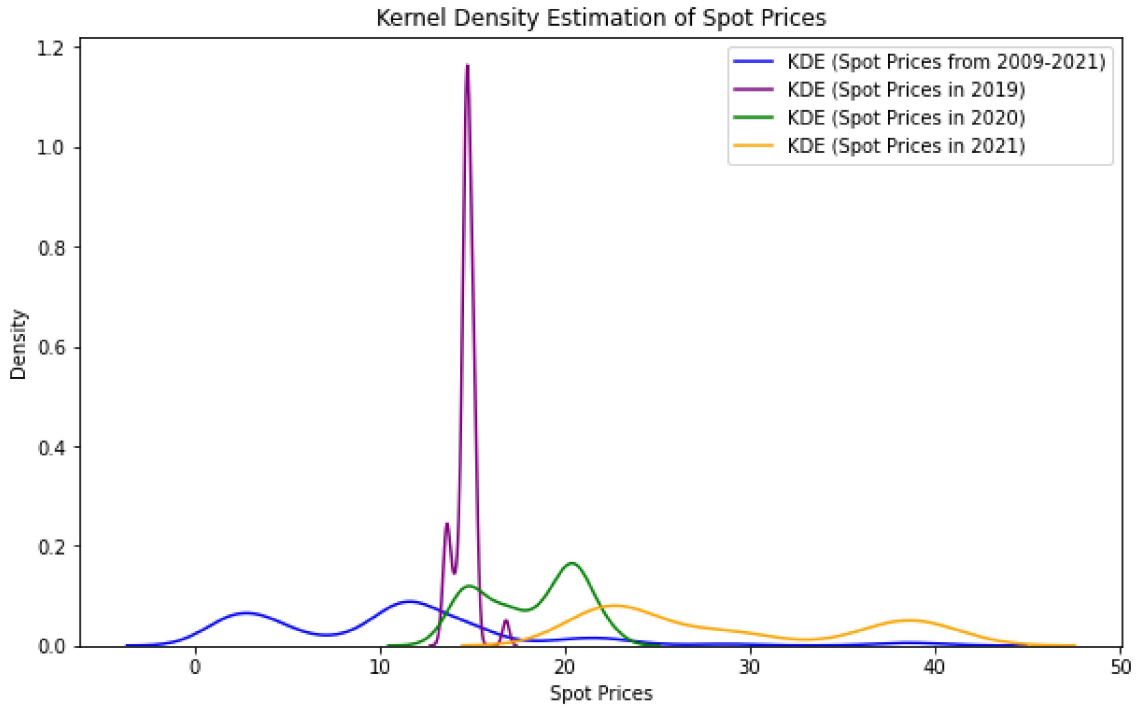
```
plt.xlabel("Spot Prices")
plt.ylabel("Density")
plt.title("Kernel Density Estimation of Spot Prices with Risk Effects")
plt.legend()
plt.show()


#  simulated spot prices with the "Total Risk" weight's effect
total_risk_simulated_prices = simulated_spot_prices[:, -1] + risk_effects[:, -1, -1]

# median of simulated spot prices with "Total Risk" effect
median_total_risk = np.median(total_risk_simulated_prices)
print("Median Total Risk Value:", median_total_risk)

#medianof each risk distribution (excluding "Total Risk")
risk_medians = []
for risk_idx in range(len(risk_weight_values) - 1):
    risk_values = simulated_spot_prices[:, -1] + risk_effects[:, -1, risk_idx]
    risk_medians.append(np.median(risk_values))

#median of all the individual risk medians
median_individual_risks = np.median(risk_medians)
print("Median of Individual Risk Medians:", median_individual_risks)
```




```
Median Total Risk Value: 22.154500000000013
Median of Individual Risk Medians: 34.75177040358797
```

In [177]:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

excel_data = pd.read_excel('C:/Users/Rishab/Documents/UCL/Masters Thesis/Relevant journa
spot_prices_2019 = excel_data['SpotPriceUSD'].tolist()[525:785]
spot_prices_2020 = excel_data['SpotPriceUSD'].tolist()[263:525]
recent_spot_prices = excel_data['SpotPriceUSD'].tolist()[:262]
historical_spot_prices = excel_data['SpotPriceUSD'].tolist()

# KDE plot to visualize the distribution of spot prices
plt.figure(figsize=(10, 6))
sns.kdeplot(historical_spot_prices, color='blue', label='KDE (Spot Prices from 2009-2021
sns.kdeplot(spot_prices_2019, color='purple', label='KDE (Spot Prices in 2019)')
sns.kdeplot(spot_prices_2020, color='green', label='KDE (Spot Prices in 2020)')
sns.kdeplot(recent_spot_prices, color='orange', label='KDE (Spot Prices in 2021)')
plt.xlabel("Spot Prices")
plt.ylabel("Density")
plt.title("Kernel Density Estimation of Spot Prices")
plt.legend()
plt.show()
```

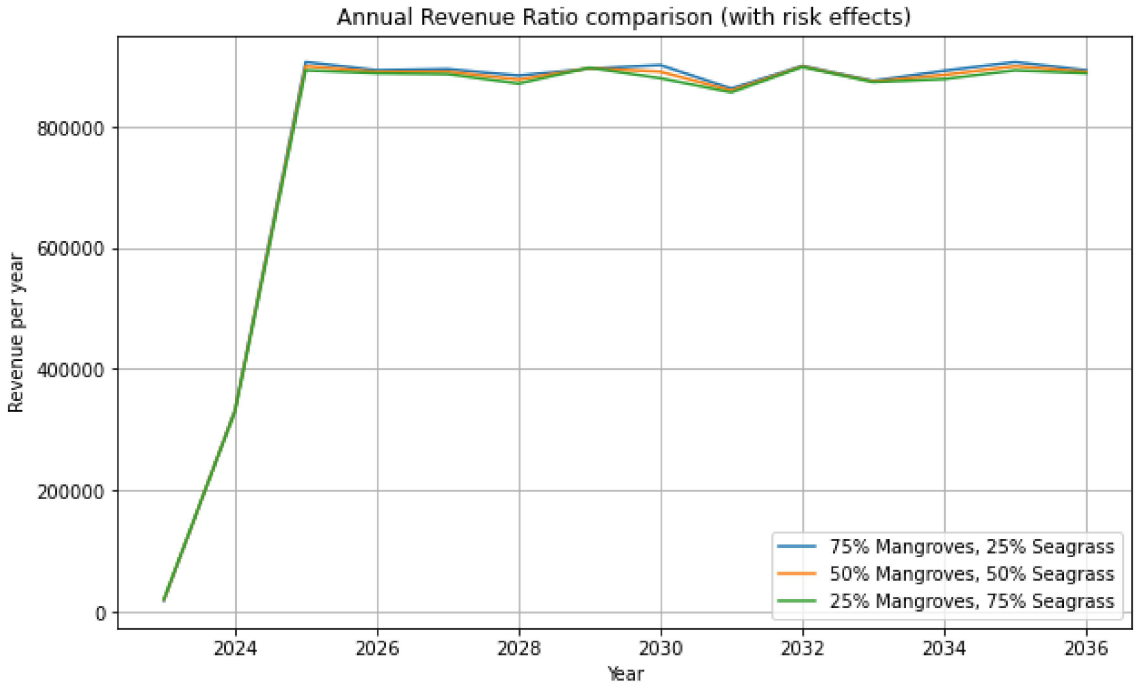


In [217]:

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

# Load your DataFrame
file_path = "C:\\Users\\Rishab\\Documents\\UCL\\Masters Thesis\\Relevant journals\\Alter
df = pd.read_excel(file_path)

# weights for the objectives
weight_1 = 0.75  # Objective 1 weight
weight_2 = 0.25  # Objective 2 weight
weight_3 = 0.50  # Both objectives weight

# Risk weights
risk_7 = 0.04
risk_8 = 0.20
risk_9 = 0.07
risk_10 = 0.68
risk_11 = 0.01

total_risk_effect = 0.275

# Calculate revenue for both objectives 75-25
revenue_1 = (df["Net Total Carbon revenue loss. Experiment 2_MOO Mangrove"]) - \
            (total_risk_effect * (df["Net Total Carbon revenue loss. Experiment 2_MOO Ma

revenue_2 = (df["Net Total Carbon revenue loss. Experiment 2_MOO Seagrass"]) - \
            (total_risk_effect * (risk_7 + risk_10 + risk_11) * (df["Net Total Carbon re

# Calculate the weighted sum of the objectives for each year and scenario
weighted_sum_7525 = (weight_1 * revenue_1) + (weight_2 * revenue_2)
weighted_sum_5050 = (weight_3 * revenue_1) + (weight_3 * revenue_2)
weighted_sum_2575 = (weight_2 * revenue_1) + (weight_1 * revenue_2)


# Dataframe to Excel
result_df = pd.DataFrame({
    "Year": df["Year"],
    "75% Mangroves, 25% Seagrass": weighted_sum_7525,
    "50% Mangroves, 50% Seagrass": weighted_sum_5050,
    "25% Mangroves, 75% Seagrass": weighted_sum_2575
})
output_excel_path = "Scenario_Weighted_Sums.xlsx"
result_df.to_excel(output_excel_path, index=False)
print("Data exported to:", output_excel_path)



#line plot for each scenario
plt.figure(figsize=(10, 6))

plt.plot(df["Year"], weighted_sum_7525, label="75% Mangroves, 25% Seagrass")
plt.plot(df["Year"], weighted_sum_5050, label="50% Mangroves, 50% Seagrass")
plt.plot(df["Year"], weighted_sum_2575, label="25% Mangroves, 75% Seagrass")

plt.title("Annual Revenue Ratio comparison (with risk effects)")
plt.xlabel("Year")
plt.ylabel("Revenue per year")
plt.legend(loc="lower right")
```

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

```
Data exported to: Scenario_Weighted_Sums.xlsx
```



In [ ]: