In [80]:
```
import pandas as pd

df1=pd.read_csv("Unit04 Global_Population.csv")
col1 = ['Country Name'] + [str(year) for year in range(2001, 2022)]
df1_filt = df1[col1]
df1_filt

df1_filt = df1_filt.dropna()
df1_filt
```

Out[80]:

	Country Name	2001	2002	2003	2004	2005	200
1	Afghanistan	21606992	22600774	23680871	24726689	25654274	2643305
2	Africa Eastern and Southern	408522129	419223717	430246635	441630149	453404076	46558137
3	Africa Western and Central	274433894	281842480	289469530	297353098	305520588	31398547
4	Albania	3060173	3051010	3039616	3026939	3011487	299254
5	Algeria	31451513	31855110	32264159	32692153	33149720	3364100
...	...	...	...	...	...	...	
262	West Bank and Gaza	2997784	3075373	3154969	3236626	3320396	340633
263	World	6193663732	6272724236	6351855732	6431527221	6511724848	659271165
264	Yemen, Rep.	17918369	18443684	18985001	19540096	20107416	2068764
265	Zambia	10692197	10971704	11256740	11550641	11856244	1217351
266	Zimbabwe	11923906	11954293	11982219	12019911	12076697	1215549

266 rows × 22 columns

In [81]:
```
df2=pd.read_csv("Unit04 Global_GDP.csv")
col2 = ['Country Name'] + [str(year) for year in range(2001, 2022)]
df2_filt = df1[col2]
df2_filt

df2_filt = df2_filt.dropna()
df2_filt
```

Out[81]:

	Country Name	2001	2002	2003	2004	2005	200
1	Afghanistan	21606992	22600774	23680871	24726689	25654274	264330
2	Africa Eastern and Southern	408522129	419223717	430246635	441630149	453404076	4655813
3	Africa Western and Central	274433894	281842480	289469530	297353098	305520588	3139854
4	Albania	3060173	3051010	3039616	3026939	3011487	29925
5	Algeria	31451513	31855110	32264159	32692153	33149720	336410
...	...	...	...	...	...	...	
262	West Bank and Gaza	2997784	3075373	3154969	3236626	3320396	34063
263	World	6193663732	6272724236	6351855732	6431527221	6511724848	659271165
264	Yemen, Rep.	17918369	18443684	18985001	19540096	20107416	206876
265	Zambia	10692197	10971704	11256740	11550641	11856244	121735
266	Zimbabwe	11923906	11954293	11982219	12019911	12076697	121554

266 rows × 22 columns

In [82]:
```
df1_filt.set_index('Country Name', inplace=True)
df2_filt.set_index('Country Name', inplace=True)
```

In [83]:
```
df1_filt = df1_filt.apply(pd.to_numeric, errors='coerce')
df2_filt = df2_filt.apply(pd.to_numeric, errors='coerce')
```

In [84]:
```
df1_filt['Mean Population'] = df1_filt.mean(axis=1)
df2_filt['Mean Per Capita GDP'] = df2_filt.mean(axis=1)
```

In [88]:
```
merged_df = pd.merge(df1_filt[['Mean Population']], df2_filt[['Mean Per Capita GDP'
merged_df = merged_df.dropna()
merged_df
```

Out[88]:

Country Name	Mean Population	Mean Per Capita GDP
Afghanistan	3.067118e+07	3.067118e+07
Africa Eastern and Southern	5.398680e+08	5.398680e+08
Africa Western and Central	3.647774e+08	3.647774e+08
Albania	2.930953e+06	2.930953e+06
Algeria	3.722809e+07	3.722809e+07
...	...	...
West Bank and Gaza	3.901664e+06	3.901664e+06
World	7.012228e+09	7.012228e+09
Yemen, Rep.	2.396467e+07	2.396467e+07
Zambia	1.433821e+07	1.433821e+07
Zimbabwe	1.314476e+07	1.314476e+07

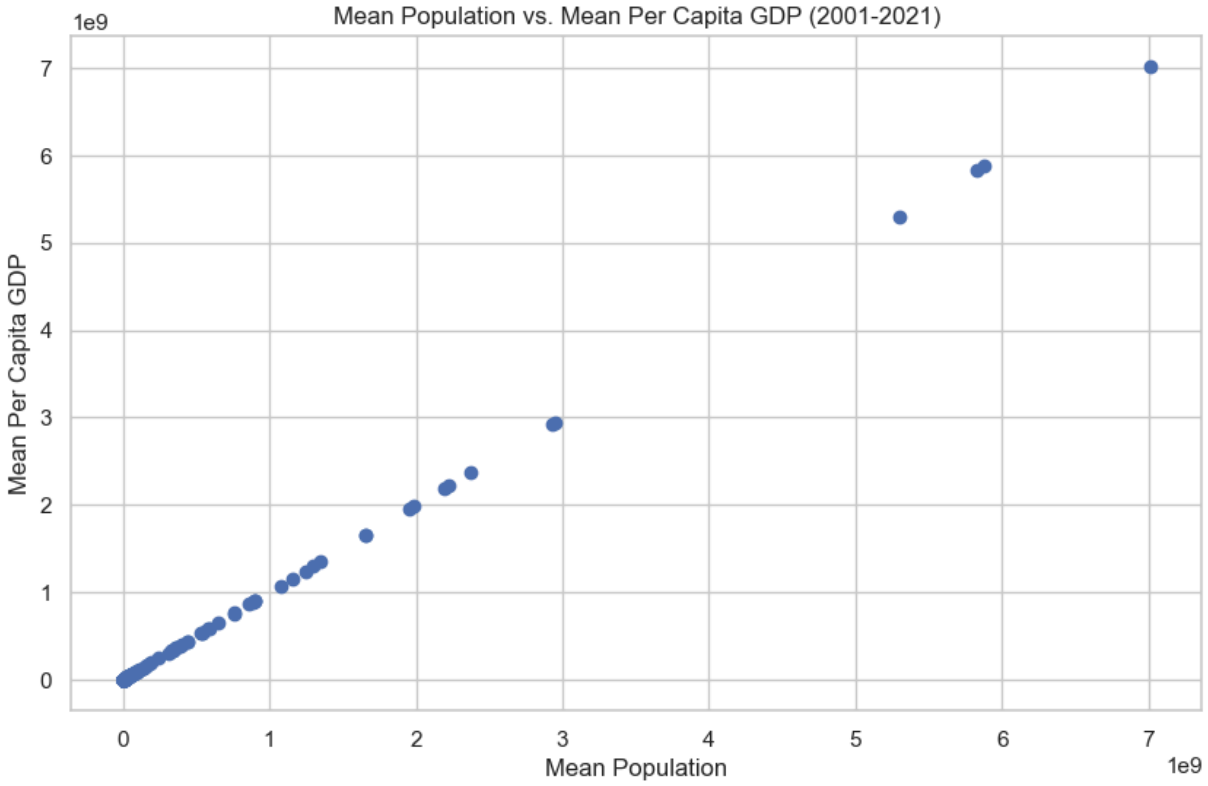
261 rows × 2 columns

In [90]:
```
correlation, p_value = pearsonr(merged_df['Mean Population'], merged_df['Mean Per C
correlation, p_value
```

Out[90]:
```
(1.0, 0.0)
```

In [ ]:
```
The correlation is statistically significant because the p-value is < 0.05.
There is a strong and positive correlation between Mean Population and Mean Per Cap
```

In [91]:
```
plt.figure(figsize=(10, 6))
plt.scatter(merged_df['Mean Population'], merged_df['Mean Per Capita GDP'])
plt.title('Mean Population vs. Mean Per Capita GDP (2001-2021)')
plt.xlabel('Mean Population')
plt.ylabel('Mean Per Capita GDP')
plt.grid(True)
plt.show()
```



In [94]:
```
import statsmodels.api as sm

#Independent variable
X = merged_df['Mean Population']

#Dependent variable
Y = merged_df['Mean Per Capita GDP']
```

In [95]:
```
X = sm.add_constant(X)  #Adds a constant term to the predictors
```

In [96]:
```
#Fit the model
model = sm.OLS(Y, X).fit()

model.summary()
```

Out[96]:

OLS Regression Results

Dep. Variable:	Mean Per Capita GDP	**R-squared:**	1.000
Model:	OLS	**Adj. R-squared:**	1.000
Method:	Least Squares	**F-statistic:**	4.824e+33
Date:	Thu, 24 Oct 2024	**Prob (F-statistic):**	0.00
Time:	14:59:52	**Log-Likelihood:**	3658.2
No. Observations:	261	**AIC:**	-7312.
Df Residuals:	259	**BIC:**	-7305.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-5.215e-08	1.29e-08	-4.055	0.000	-7.75e-08	-2.68e-08
Mean Population	1.0000	1.44e-17	6.95e+16	0.000	1.000	1.000

Omnibus:	362.180	**Durbin-Watson:**	1.297
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	34518.020
Skew:	6.497	**Prob(JB):**	0.00
Kurtosis:	57.820	**Cond. No.**	9.34e+08

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.34e+08. This might indicate that there are strong multicollinearity or other numerical problems.

In [ ]: