Import library

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Import datasets

In [2]:
```
country_code_mapping = pd.read_csv('Metadata_Country.csv')
data1 = pd.read_csv('GDP_PPP_per_capita.csv',skiprows=3) #GDP per capita, PPP
data2 = pd.read_csv('Health_Expenditure_per_capita_PPP.csv',skiprows=3) #Health Expend
data3 = pd.read_csv('Hospital_Beds_per_capita.csv',skiprows=3) #Hospital Beds per capi
data4 = pd.read_csv('Doctors_per_capita.csv')
data5 = pd.read_csv('total-cancer-deaths-by-type.csv')
```

Inspect dataset: Country Code Mapping

In [3]:
```
country_code_mapping.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 265 entries, 0 to 264
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Country Code  265 non-null    object 
 1   Region        217 non-null    object 
 2   IncomeGroup   216 non-null    object 
 3   SpecialNotes  126 non-null    object 
 4   TableName     265 non-null    object 
 5   Unnamed: 5    0 non-null      float64
dtypes: float64(1), object(5)
memory usage: 12.5+ KB
```

In [61]:
```
country_code_mapping.head(20)
```

Out[61]:

	Country_Code	Region	IncomeGroup	Country_Name
0	ABW	Latin America & Caribbean	High income	Aruba
1	AFE	NaN	NaN	Africa Eastern and Southern
2	AFG	South Asia	Low income	Afghanistan
3	AFW	NaN	NaN	Africa Western and Central
4	AGO	Sub-Saharan Africa	Lower middle income	Angola
5	ALB	Europe & Central Asia	Upper middle income	Albania
6	AND	Europe & Central Asia	High income	Andorra
7	ARB	NaN	NaN	Arab World
8	ARE	Middle East & North Africa	High income	United Arab Emirates
9	ARG	Latin America & Caribbean	Upper middle income	Argentina
10	ARM	Europe & Central Asia	Upper middle income	Armenia
11	ASM	East Asia & Pacific	Upper middle income	American Samoa
12	ATG	Latin America & Caribbean	High income	Antigua and Barbuda
13	AUS	East Asia & Pacific	High income	Australia
14	AUT	Europe & Central Asia	High income	Austria
15	AZE	Europe & Central Asia	Upper middle income	Azerbaijan
16	BDI	Sub-Saharan Africa	Low income	Burundi
17	BEL	Europe & Central Asia	High income	Belgium
18	BEN	Sub-Saharan Africa	Lower middle income	Benin
19	BFA	Sub-Saharan Africa	Low income	Burkina Faso

In [5]:
```
country_code_mapping.describe()
```

Out[5]:

	Unnamed: 5
count	0.0
mean	NaN
std	NaN
min	NaN
25%	NaN
50%	NaN
75%	NaN
max	NaN

In [6]:
```
country_code_mapping.duplicated()
```

Out[6]:
```
0      False
1      False
2      False
3      False
4      False
       ...
260    False
261    False
262    False
263    False
264    False
Length: 265, dtype: bool
```

In [7]:
```
country_code_mapping.nunique()
```

Out[7]:
```
Country Code    265
Region            7
IncomeGroup       4
SpecialNotes    111
TableName       265
Unnamed: 5        0
dtype: int64
```

Inspect dataset: GDP_PPP_per_capita

In [8]:
```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 266 entries, 0 to 265
Data columns (total 67 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Country Name    266 non-null    object 
 1   Country Code    266 non-null    object 
 2   Indicator Name  266 non-null    object 
 3   Indicator Code  266 non-null    object 
 4   1960            0 non-null      float64
 5   1961            0 non-null      float64
 6   1962            0 non-null      float64
 7   1963            0 non-null      float64
 8   1964            0 non-null      float64
 9   1965            0 non-null      float64
 10  1966            0 non-null      float64
 11  1967            0 non-null      float64
 12  1968            0 non-null      float64
 13  1969            0 non-null      float64
 14  1970            0 non-null      float64
 15  1971            0 non-null      float64
 16  1972            0 non-null      float64
 17  1973            0 non-null      float64
 18  1974            0 non-null      float64
 19  1975            0 non-null      float64
 20  1976            0 non-null      float64
 21  1977            0 non-null      float64
 22  1978            0 non-null      float64
 23  1979            0 non-null      float64
 24  1980            0 non-null      float64
 25  1981            0 non-null      float64
 26  1982            0 non-null      float64
 27  1983            0 non-null      float64
 28  1984            0 non-null      float64
 29  1985            0 non-null      float64
 30  1986            0 non-null      float64
 31  1987            0 non-null      float64
 32  1988            0 non-null      float64
 33  1989            0 non-null      float64
 34  1990            207 non-null    float64
 35  1991            209 non-null    float64
 36  1992            211 non-null    float64
 37  1993            212 non-null    float64
 38  1994            214 non-null    float64
 39  1995            225 non-null    float64
 40  1996            225 non-null    float64
 41  1997            227 non-null    float64
 42  1998            227 non-null    float64
 43  1999            228 non-null    float64
 44  2000            236 non-null    float64
 45  2001            237 non-null    float64
 46  2002            238 non-null    float64
 47  2003            238 non-null    float64
 48  2004            239 non-null    float64
 49  2005            239 non-null    float64
 50  2006            240 non-null    float64
 51  2007            240 non-null    float64
 52  2008            242 non-null    float64
 53  2009            243 non-null    float64
 54  2010            243 non-null    float64
 55  2011            244 non-null    float64
 56  2012            242 non-null    float64
 57  2013            244 non-null    float64
 58  2014            243 non-null    float64
 59  2015            243 non-null    float64
 60  2016            242 non-null    float64
 61  2017            242 non-null    float64
 62  2018            242 non-null    float64
 63  2019            241 non-null    float64
 64  2020            240 non-null    float64
 65  2021            226 non-null    float64
```

```
 66  Unnamed: 66     0 non-null      float64
dtypes: float64(63), object(4)
memory usage: 139.4+ KB
```

In [9]:
```
data1.head()
```

Out[9]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	
0	Aruba	ABW	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	368
1	Africa Eastern and Southern	AFE	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	33
2	Afghanistan	AFG	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	20
3	Africa Western and Central	AFW	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	40
4	Angola	AGO	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	76

5 rows × 67 columns

In [10]:
```
data1.describe()
```

Out[10]:

	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	...	2013	20
count	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	244.000000	243.0000
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	18788.396174	19218.791
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	21326.505387	21262.829
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	738.474892	720.324
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	4088.679166	4373.5136
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	11546.856197	11980.5663
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	24684.244750	25444.645
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	153563.910960	152856.3410

8 rows × 63 columns

In [11]:
```
data1.duplicated()
```

Out[11]:
```
0      False
1      False
2      False
3      False
4      False
       ...
261    False
262    False
263    False
264    False
265    False
Length: 266, dtype: bool
```

In [12]:
```
data1.nunique()
```

Out[12]:
```
Country Name      266
Country Code      266
Indicator Name      1
Indicator Code      1
1960                0
                 ...
2018              240
2019              240
2020              238
2021              225
Unnamed: 66         0
Length: 67, dtype: int64
```

Inspect dataset: Health_Expenditure_per_capita_PPP

In [13]:
```
data2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 266 entries, 0 to 265
Data columns (total 67 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Country Name    266 non-null    object 
 1   Country Code    266 non-null    object 
 2   Indicator Name  266 non-null    object 
 3   Indicator Code  266 non-null    object 
 4   1960            0 non-null      float64
 5   1961            0 non-null      float64
 6   1962            0 non-null      float64
 7   1963            0 non-null      float64
 8   1964            0 non-null      float64
 9   1965            0 non-null      float64
 10  1966            0 non-null      float64
 11  1967            0 non-null      float64
 12  1968            0 non-null      float64
 13  1969            0 non-null      float64
 14  1970            0 non-null      float64
 15  1971            0 non-null      float64
 16  1972            0 non-null      float64
 17  1973            0 non-null      float64
 18  1974            0 non-null      float64
 19  1975            0 non-null      float64
 20  1976            0 non-null      float64
 21  1977            0 non-null      float64
 22  1978            0 non-null      float64
 23  1979            0 non-null      float64
 24  1980            0 non-null      float64
 25  1981            0 non-null      float64
 26  1982            0 non-null      float64
 27  1983            0 non-null      float64
 28  1984            0 non-null      float64
 29  1985            0 non-null      float64
 30  1986            0 non-null      float64
 31  1987            0 non-null      float64
 32  1988            0 non-null      float64
 33  1989            0 non-null      float64
 34  1990            0 non-null      float64
 35  1991            0 non-null      float64
 36  1992            0 non-null      float64
 37  1993            0 non-null      float64
 38  1994            0 non-null      float64
 39  1995            0 non-null      float64
 40  1996            0 non-null      float64
 41  1997            0 non-null      float64
 42  1998            0 non-null      float64
 43  1999            0 non-null      float64
 44  2000            232 non-null    float64
 45  2001            232 non-null    float64
 46  2002            233 non-null    float64
 47  2003            235 non-null    float64
 48  2004            235 non-null    float64
 49  2005            235 non-null    float64
 50  2006            235 non-null    float64
 51  2007            235 non-null    float64
 52  2008            235 non-null    float64
 53  2009            235 non-null    float64
 54  2010            236 non-null    float64
 55  2011            237 non-null    float64
 56  2012            236 non-null    float64
 57  2013            235 non-null    float64
 58  2014            235 non-null    float64
 59  2015            235 non-null    float64
 60  2016            234 non-null    float64
 61  2017            235 non-null    float64
 62  2018            235 non-null    float64
 63  2019            234 non-null    float64
 64  2020            0 non-null      float64
 65  2021            0 non-null      float64
```

```
 66  Unnamed: 66     0 non-null      float64
dtypes: float64(63), object(4)
memory usage: 139.4+ KB
```

In [14]:
```
data2.head()
```

Out[14]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	
0	Aruba	ABW	Current health expenditure per capita, PPP (cu...	SH.XPD.CHEX.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	
1	Africa Eastern and Southern	AFE	Current health expenditure per capita, PPP (cu...	SH.XPD.CHEX.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	21
2	Afghanistan	AFG	Current health expenditure per capita, PPP (cu...	SH.XPD.CHEX.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	174
3	Africa Western and Central	AFW	Current health expenditure per capita, PPP (cu...	SH.XPD.CHEX.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	14
4	Angola	AGO	Current health expenditure per capita, PPP (cu...	SH.XPD.CHEX.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	20

5 rows × 67 columns

In [15]:
```
data2.describe()
```

Out[15]:

	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	...	2013	2014
count	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	235.000000	235.000000
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	1228.107850	1274.224127
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	1534.373057	1586.987416
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	26.874046	33.809780
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	200.585976	209.538879
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	652.890991	654.405396
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	1466.027405	1551.327820
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	8522.125977	8939.396484

8 rows × 63 columns

In [16]:
```
data2.duplicated()
```

Out[16]:
```
0      False
1      False
2      False
3      False
4      False
       ...
261    False
262    False
263    False
264    False
265    False
Length: 266, dtype: bool
```

In [17]:
```
data2.nunique()
```

Out[17]:
```
Country Name      266
Country Code      266
Indicator Name      1
Indicator Code      1
1960                0
                 ...
2018              232
2019              231
2020                0
2021                0
Unnamed: 66         0
Length: 67, dtype: int64
```

Inspect dataset: Hospital_Beds_per_1000

In [18]:
```
data3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 266 entries, 0 to 265
Data columns (total 67 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Country Name    266 non-null    object
 1   Country Code    266 non-null    object
 2   Indicator Name  266 non-null    object
 3   Indicator Code  266 non-null    object
 4   1960            164 non-null    float64
 5   1961            2 non-null      float64
 6   1962            2 non-null      float64
 7   1963            2 non-null      float64
 8   1964            2 non-null      float64
 9   1965            8 non-null      float64
 10  1966            5 non-null      float64
 11  1967            5 non-null      float64
 12  1968            5 non-null      float64
 13  1969            5 non-null      float64
 14  1970            176 non-null    float64
 15  1971            8 non-null      float64
 16  1972            9 non-null      float64
 17  1973            9 non-null      float64
 18  1974            11 non-null     float64
 19  1975            91 non-null     float64
 20  1976            15 non-null     float64
 21  1977            11 non-null     float64
 22  1978            12 non-null     float64
 23  1979            12 non-null     float64
 24  1980            131 non-null    float64
 25  1981            74 non-null     float64
 26  1982            30 non-null     float64
 27  1983            26 non-null     float64
 28  1984            33 non-null     float64
 29  1985            100 non-null    float64
 30  1986            58 non-null     float64
 31  1987            63 non-null     float64
 32  1988            61 non-null     float64
 33  1989            84 non-null     float64
 34  1990            190 non-null    float64
 35  1991            96 non-null     float64
 36  1992            84 non-null     float64
 37  1993            109 non-null    float64
 38  1994            84 non-null     float64
 39  1995            84 non-null     float64
 40  1996            117 non-null    float64
 41  1997            82 non-null     float64
 42  1998            80 non-null     float64
 43  1999            74 non-null     float64
 44  2000            130 non-null    float64
 45  2001            131 non-null    float64
 46  2002            142 non-null    float64
 47  2003            139 non-null    float64
 48  2004            129 non-null    float64
 49  2005            166 non-null    float64
 50  2006            169 non-null    float64
 51  2007            150 non-null    float64
 52  2008            145 non-null    float64
 53  2009            159 non-null    float64
 54  2010            171 non-null    float64
 55  2011            166 non-null    float64
 56  2012            150 non-null    float64
 57  2013            144 non-null    float64
 58  2014            143 non-null    float64
 59  2015            138 non-null    float64
 60  2016            134 non-null    float64
 61  2017            128 non-null    float64
 62  2018            41 non-null     float64
 63  2019            8 non-null      float64
 64  2020            0 non-null      float64
 65  2021            0 non-null      float64
```

```
 66  Unnamed: 66     0 non-null      float64
dtypes: float64(63), object(4)
memory usage: 139.4+ KB
```

In [19]:
```
data3.head()
```

Out[19]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	201
0	Aruba	ABW	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN	NaN	NaN	...	Nal
1	Africa Eastern and Southern	AFE	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	1.959677	NaN	NaN	NaN	NaN	NaN	...	Nal
2	Afghanistan	AFG	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	0.170627	NaN	NaN	NaN	NaN	NaN	...	0.5
3	Africa Western and Central	AFW	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	0.781043	NaN	NaN	NaN	NaN	NaN	...	Nal
4	Angola	AGO	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	2.061462	NaN	NaN	NaN	NaN	NaN	...	Nal

5 rows × 67 columns

In [20]:
```
data3.describe()
```

Out[20]:

	1960	1961	1962	1963	1964	1965	1966	1967	1
count	164.000000	2.000000	2.000000	2.000000	2.000000	8.000000	5.000000	5.000000	5.000
mean	3.798238	10.150000	10.300000	10.500000	10.650000	5.652500	5.212000	5.240000	5.250
std	4.165419	1.626346	1.555635	1.697056	1.626346	4.511964	5.109708	5.165075	5.237
min	0.115764	9.000000	9.200000	9.300000	9.500000	1.440000	1.520000	1.500000	1.450
25%	1.230003	9.575000	9.750000	9.900000	10.075000	1.440000	1.520000	1.500000	1.450
50%	2.439550	10.150000	10.300000	10.500000	10.650000	5.300000	1.520000	1.500000	1.450
75%	5.184699	10.725000	10.850000	11.100000	11.225000	9.000000	9.700000	9.900000	10.100
max	40.315456	11.300000	11.400000	11.700000	11.800000	11.900000	11.800000	11.800000	11.800

8 rows × 63 columns

In [21]:
```
data3.duplicated()
```

Out[21]:
```
0      False
1      False
2      False
3      False
4      False
       ...
261    False
262    False
263    False
264    False
265    False
Length: 266, dtype: bool
```

In [22]:
```
data3.nunique()
```

Out[22]:
```
Country Name      266
Country Code      266
Indicator Name      1
Indicator Code      1
1960              161
                 ...
2018               40
2019                8
2020                0
2021                0
Unnamed: 66         0
Length: 67, dtype: int64
```

Inspect dataset: Doctors_per_10000

In [23]:
```
data4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 196 entries, 0 to 195
Data columns (total 34 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   SpatialDimValueCode  194 non-null    object 
 1   Location            194 non-null    object 
 2   1990                55 non-null     float64
 3   1991                57 non-null     float64
 4   1992                57 non-null     float64
 5   1993                57 non-null     float64
 6   1994                55 non-null     float64
 7   1995                63 non-null     float64
 8   1996                59 non-null     float64
 9   1997                68 non-null     float64
 10  1998                71 non-null     float64
 11  1999                69 non-null     float64
 12  2000                92 non-null     float64
 13  2001                87 non-null     float64
 14  2002                86 non-null     float64
 15  2003                90 non-null     float64
 16  2004                129 non-null    float64
 17  2005                100 non-null    float64
 18  2006                99 non-null     float64
 19  2007                105 non-null    float64
 20  2008                129 non-null    float64
 21  2009                128 non-null    float64
 22  2010                123 non-null    float64
 23  2011                111 non-null    float64
 24  2012                112 non-null    float64
 25  2013                110 non-null    float64
 26  2014                114 non-null    float64
 27  2015                109 non-null    float64
 28  2016                109 non-null    float64
 29  2017                118 non-null    float64
 30  2018                132 non-null    float64
 31  2019                83 non-null     float64
 32  2020                67 non-null     float64
 33  Unnamed: 33         0 non-null      float64
dtypes: float64(32), object(2)
memory usage: 52.2+ KB
```

In [24]:
```
data4.head()
```

Out[24]:

	SpatialDimValueCode	Location	1990	1991	1992	1993	1994	1995	1996	1997	...	2012	20
0	MAR	Morocco	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	6
1	AFG	Afghanistan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	2.41	2
2	AGO	Angola	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.59	...	NaN	N
3	ALB	Albania	13.74	14.61	16.22	14.13	13.45	13.63	13.81	13.89	...	12.68	12
4	AND	Andorra	NaN	NaN	NaN	NaN	NaN	22.38	NaN	24.57	...	NaN	N

5 rows × 34 columns

In [25]:
```
data4.describe()
```

Out[25]:

	1990	1991	1992	1993	1994	1995	1996	1997	
count	55.000000	57.000000	57.000000	57.000000	55.000000	63.000000	59.000000	68.000000	71.
mean	23.848727	24.177895	25.191579	24.151228	25.459455	23.208254	24.350000	22.441471	22
std	11.111019	11.766469	11.016171	12.669590	11.851134	11.880149	11.424474	12.656483	12.
min	1.270000	1.390000	1.650000	0.580000	1.850000	1.430000	1.140000	0.260000	0.
25%	15.575000	16.080000	16.900000	14.130000	17.165000	12.355000	14.050000	11.990000	12
50%	25.030000	26.220000	26.700000	26.990000	27.820000	23.250000	27.040000	23.820000	23.
75%	32.385000	33.570000	32.900000	33.020000	33.145000	31.920000	31.930000	31.747500	31.
max	46.710000	48.490000	49.890000	54.780000	55.840000	52.200000	54.970000	56.980000	57

8 rows × 32 columns

In [26]:
```
data4.duplicated()
```

Out[26]:
```
0      False
1      False
2      False
3      False
4      False
       ...
191    False
192    False
193    False
194    False
195     True
Length: 196, dtype: bool
```

In [27]:
```
data4.nunique()
```

Out[27]:
```
SpatialDimValueCode    194
Location               194
1990                    55
1991                    57
1992                    57
1993                    57
1994                    55
1995                    63
1996                    57
1997                    68
1998                    71
1999                    69
2000                    92
2001                    87
2002                    86
2003                    89
2004                   126
2005                   100
2006                    97
2007                   104
2008                   124
2009                   125
2010                   119
2011                   111
2012                   110
2013                   110
2014                   113
2015                   105
2016                   106
2017                   114
2018                   127
2019                    81
2020                    66
Unnamed: 33              0
dtype: int64
```

Inspect dataset: Total cancer deaths by type

In [28]:
```
data5.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6840 entries, 0 to 6839
Data columns (total 26 columns):
 #   Column                        Non-Null Count  Dtype 
---  ------                        --------------  ----- 
 0   Entity                        6840 non-null   object
 1   Code                          6150 non-null   object
 2   Year                          6840 non-null   int64 
 3   Liver_cancer                  6840 non-null   int64 
 4   Kidney_cancer                 6840 non-null   int64 
 5   Lip_and_oral_cavity_cancer    6840 non-null   int64 
 6   Lung_cancer                   6840 non-null   int64 
 7   Larynx_cancer                 6840 non-null   int64 
 8   Gallbladder_cancer            6840 non-null   int64 
 9   Malignant_skin                6840 non-null   int64 
 10  Leukemia                      6840 non-null   int64 
 11  Multiple_myeloma              6840 non-null   int64 
 12  Breast_cancer                 6840 non-null   int64 
 13  Prostate_cancer               6840 non-null   int64 
 14  Thyroid_cancer                6840 non-null   int64 
 15  Stomach_cancer                6840 non-null   int64 
 16  Bladder_cancer                6840 non-null   int64 
 17  Uterine_cancer                6840 non-null   int64 
 18  Ovarian_cancer                6840 non-null   int64 
 19  Cervical_cancer               6840 non-null   int64 
 20  Brain_cancer                  6840 non-null   int64 
 21  Non-Hodgkin_lymphoma          6840 non-null   int64 
 22  Pancreatic_cancer             6840 non-null   int64 
 23  Esophageal_cancer             6840 non-null   int64 
 24  Colon_cancer                  6840 non-null   int64 
 25  Non-melanoma_skin_cancer      6840 non-null   int64 
dtypes: int64(24), object(2)
memory usage: 1.4+ MB
```

In [29]:
```
data5.head()
```

Out[29]:

	Entity	Code	Year	Liver_cancer	Kidney_cancer	Lip_and_oral_cavity_cancer	Lung_cancer	Lar
0	Afghanistan	AFG	1990	851	66	89	983	
1	Afghanistan	AFG	1991	866	66	89	982	
2	Afghanistan	AFG	1992	890	68	91	989	
3	Afghanistan	AFG	1993	914	70	93	995	
4	Afghanistan	AFG	1994	933	71	94	996	

5 rows × 26 columns

In [30]:
```
data5.describe()
```

Out[30]:

	Year	Liver_cancer	Kidney_cancer	Lip_and_oral_cavity_cancer	Lung_cancer	Larynx_c
count	6840.000000	6840.00000	6840.000000	6840.000000	6.840000e+03	6840.0
mean	2004.500000	11101.27193	3199.431725	3675.574415	4.104193e+04	2657.4
std	8.656074	49811.25638	12976.577516	14825.600877	1.691944e+05	10193.5
min	1990.000000	0.00000	0.000000	0.000000	0.000000e+00	0.0
25%	1997.000000	39.00000	14.000000	17.000000	1.427500e+02	14.0
50%	2004.500000	222.00000	86.000000	89.000000	8.630000e+02	80.0
75%	2012.000000	985.00000	517.000000	463.000000	5.440750e+03	435.0
max	2019.000000	484577.00000	166438.000000	199398.000000	2.042640e+06	123356.0

8 rows × 24 columns

In [31]:
```
data5.duplicated()
```

Out[31]:
```
0       False
1       False
2       False
3       False
4       False
        ...
6835    False
6836    False
6837    False
6838    False
6839    False
Length: 6840, dtype: bool
```

In [32]:
```
data5.nunique()
```

Out[32]:
```
Entity                          228
Code                            205
Year                             30
Liver_cancer                   2456
Kidney_cancer                  1952
Lip_and_oral_cavity_cancer     1932
Lung_cancer                    3786
Larynx_cancer                  1824
Gallbladder_cancer             1916
Malignant_skin                 1502
Leukemia                       2669
Multiple_myeloma               1732
Breast_cancer                  3223
Prostate_cancer                2939
Thyroid_cancer                 1370
Stomach_cancer                 3318
Bladder_cancer                 2200
Uterine_cancer                 1652
Ovarian_cancer                 2123
Cervical_cancer                2577
Brain_cancer                   2288
Non-Hodgkin_lymphoma           2311
Pancreatic_cancer              2616
Esophageal_cancer              2431
Colon_cancer                   3344
Non-melanoma_skin_cancer       1357
dtype: int64
```

Preparing all datasets for merging

In [33]:
```
country_code_mapping.drop(country_code_mapping.columns[[3,5]], inplace=True, axis=1)
country_code_mapping = country_code_mapping.rename(columns={'TableName': 'Country_Name
country_code_mapping.head(5)
```

Out[33]:

	Country_Code	Region	IncomeGroup	Country_Name
0	ABW	Latin America & Caribbean	High income	Aruba
1	AFE	NaN	NaN	Africa Eastern and Southern
2	AFG	South Asia	Low income	Afghanistan
3	AFW	NaN	NaN	Africa Western and Central
4	AGO	Sub-Saharan Africa	Lower middle income	Angola

In [34]:
```
data1.drop(data1.iloc[:, 2:44], inplace=True, axis=1)
data1.drop(data1.iloc[:, -5:], inplace=True, axis=1)
data1 = data1.rename(columns={'2017': 'gdp_per_capita_2017','Country Name': 'Country_N
data1['gdp_per_capita_2017'] = data1['gdp_per_capita_2017'].fillna(data1.iloc[:, 2:].m
data1.head(5)
```

Out[34]:

	Country_Name	Country_Code	2000	2001	2002	2003	200
0	Aruba	ABW	30149.423396	31421.640537	30907.007372	31205.360781	33774.5272(
1	Africa Eastern and Southern	AFE	2156.354867	2227.889103	2291.273040	2351.059098	2484.6025
2	Afghanistan	AFG	NaN	NaN	876.327643	928.191569	925.7042
3	Africa Western and Central	AFW	2025.095894	2121.942630	2297.086180	2406.429857	2596.9713
4	Angola	AGO	3271.270265	3372.470708	3765.608654	3823.350144	4208.0389

In [35]:

```
data2.drop(data2.iloc[:, 2:44], inplace=True, axis=1)
data2.drop(data2.iloc[:, -5:], inplace=True, axis=1)
data2 = data2.rename(columns={'2017': 'health_exp_per_capita_2017', 'Country Name': 'C
data2['health_exp_per_capita_2017'] = data2['health_exp_per_capita_2017'].fillna(data2
data2.head(5)
```

Out[35]:

	Country_Name	Country_Code	2000	2001	2002	2003	2004	20(
0	Aruba	ABW	NaN	NaN	NaN	NaN	NaN	Na
1	Africa Eastern and Southern	AFE	117.467280	125.101753	123.750953	135.831857	144.733246	153.2938
2	Afghanistan	AFG	NaN	NaN	81.271034	82.457848	89.470055	100.7069
3	Africa Western and Central	AFW	78.675077	82.142511	79.666006	118.302697	119.973695	124.3684
4	Angola	AGO	62.695866	151.747040	126.025124	136.199402	167.632629	137.9809

In [36]:

```
data3.drop(data3.iloc[:, 2:44], inplace=True, axis=1)
data3.drop(data3.iloc[:, -5:], inplace=True, axis=1)
data3 = data3.rename(columns={'2017': 'Beds_per_capita_2017', 'Country Name': 'Country
data3['Beds_per_capita_2017'] = data3['Beds_per_capita_2017'].fillna(data3.iloc[:, 2:]
data3.head(5)
```

Out[36]:

	Country_Name	Country_Code	2000	2001	2002	2003	2004	2005	2006	2007	2008	200
0	Aruba	ABW	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
1	Africa Eastern and Southern	AFE	NaN	NaN	NaN	NaN	NaN	NaN	0.911871	NaN	NaN	Na
2	Afghanistan	AFG	0.3	0.39	0.39	0.39	0.39	0.42	0.420000	0.42	0.42	0.4
3	Africa Western and Central	AFW	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	Angola	AGO	NaN	NaN	NaN	NaN	NaN	0.80	NaN	NaN	NaN	Na

In [37]:

```
data4.drop(data4.iloc[:, 2:12], inplace=True, axis=1)
data4.drop(data4.iloc[:, -4:], inplace=True, axis=1)
data4 = data4.rename(columns={'SpatialDimValueCode': 'Country_Code', 'Location': 'Coun
data4['Dr_per_10000_2017'] = data4['Dr_per_10000_2017'].fillna(data4.iloc[:, 2:].mean(
data4.head(5)
```

Out[37]:

	Country_Code	Country_Name	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	:
0	MAR	Morocco	NaN	NaN	NaN	NaN	5.31	NaN	NaN	5.86	NaN	6.48	
1	AFG	Afghanistan	NaN	1.9	NaN	NaN	NaN	NaN	1.60	1.74	1.74	2.13	
2	AGO	Angola	NaN	NaN	NaN	NaN	0.62	NaN	NaN	NaN	NaN	1.31	
3	ALB	Albania	13.82	13.1	11.7	11.86	11.91	NaN	11.84	11.95	NaN	12.30	'
4	AND	Andorra	25.69	26.0	29.0	33.33	31.32	32.32	30.12	30.11	NaN	31.48	

In [38]:

```
data5 = data5[data5['Year'] == 2017]
```

```
data5 = data5.rename(columns={'Entity': 'Country_Name','Code': 'Country_Code'})
data5
```

Out[38]:

	Country_Name	Country_Code	Year	Liver_cancer	Kidney_cancer	Lip_and_oral_cavity_cancer	Lu
27	Afghanistan	AFG	2017	1305	133	131	
57	African Region (WHO)	NaN	2017	20623	5334	8322	
87	Albania	ALB	2017	279	92	50	
117	Algeria	DZA	2017	643	260	242	
147	American Samoa	ASM	2017	3	0	0	
...	...	...	...	...	...	...	
6717	World Bank Lower Middle Income	NaN	2017	82027	23377	100478	
6747	World Bank Upper Middle Income	NaN	2017	240838	54503	46330	
6777	Yemen	YEM	2017	393	102	103	
6807	Zambia	ZMB	2017	190	114	195	
6837	Zimbabwe	ZWE	2017	982	63	154	

228 rows × 26 columns

Joining datasets

In [39]:

```
#Joining two datasets:
data11 = data1.iloc[:,[0,1,19]]
df1 = pd.merge(country_code_mapping,data11, how='left')
df1
```

Out[39]:

	Country_Code	Region	IncomeGroup	Country_Name	gdp_per_capita_2017
0	ABW	Latin America & Caribbean	High income	Aruba	38893.960556
1	AFE	NaN	NaN	Africa Eastern and Southern	3670.891196
2	AFG	South Asia	Low income	Afghanistan	2058.400221
3	AFW	NaN	NaN	Africa Western and Central	4115.645480
4	AGO	Sub-Saharan Africa	Lower middle income	Angola	7310.896551
...	...	...	...	...	...
260	XKX	Europe & Central Asia	Upper middle income	Kosovo	10436.168846
261	YEM	Middle East & North Africa	Low income	Yemen, Rep.	3525.790032
262	ZAF	Sub-Saharan Africa	Upper middle income	South Africa	13860.270166
263	ZMB	Sub-Saharan Africa	Low income	Zambia	3485.021780
264	ZWE	Sub-Saharan Africa	Lower middle income	Zimbabwe	2416.049969

265 rows × 5 columns

In [40]:

```
#Joining two datasets:
data22 = data2.iloc[:,[0,1,19]]
df2 = pd.merge(df1,data22, how='left')
df2
```

Out[40]:

	Country_Code	Region	IncomeGroup	Country_Name	gdp_per_capita_2017	health_exp_per_capi
0	ABW	Latin America & Caribbean	High income	Aruba	38893.960556	
1	AFE	NaN	NaN	Africa Eastern and Southern	3670.891196	216
2	AFG	South Asia	Low income	Afghanistan	2058.400221	259
3	AFW	NaN	NaN	Africa Western and Central	4115.645480	156
4	AGO	Sub-Saharan Africa	Lower middle income	Angola	7310.896551	204
...	...	...	...	...	...	
260	XKX	Europe & Central Asia	Upper middle income	Kosovo	10436.168846	
261	YEM	Middle East & North Africa	Low income	Yemen, Rep.	3525.790032	18
262	ZAF	Sub-Saharan Africa	Upper middle income	South Africa	13860.270166	110
263	ZMB	Sub-Saharan Africa	Low income	Zambia	3485.021780	152
264	ZWE	Sub-Saharan Africa	Lower middle income	Zimbabwe	2416.049969	226

265 rows × 6 columns

In [41]:

```
#Joining two datasets:
data33 = data3.iloc[:,[0,1,19]]
df3 = pd.merge(df2,data33, how='left')
df3
```

Out[41]:

	Country_Code	Region	IncomeGroup	Country_Name	gdp_per_capita_2017	health_exp_per_capi
0	ABW	Latin America & Caribbean	High income	Aruba	38893.960556	
1	AFE	NaN	NaN	Africa Eastern and Southern	3670.891196	216
2	AFG	South Asia	Low income	Afghanistan	2058.400221	259
3	AFW	NaN	NaN	Africa Western and Central	4115.645480	156
4	AGO	Sub-Saharan Africa	Lower middle income	Angola	7310.896551	204
...	...	...	...	...	...	
260	XKX	Europe & Central Asia	Upper middle income	Kosovo	10436.168846	
261	YEM	Middle East & North Africa	Low income	Yemen, Rep.	3525.790032	18
262	ZAF	Sub-Saharan Africa	Upper middle income	South Africa	13860.270166	110
263	ZMB	Sub-Saharan Africa	Low income	Zambia	3485.021780	152
264	ZWE	Sub-Saharan Africa	Lower middle income	Zimbabwe	2416.049969	226

265 rows × 7 columns

In [42]:

```
#Joining two datasets:
data44 = data4.iloc[:,[0,1,19]]
df4 = pd.merge(df3,data44, how='left')
df4
```

Out[42]:

	Country_Code	Region	IncomeGroup	Country_Name	gdp_per_capita_2017	health_exp_per_capi
0	ABW	Latin America & Caribbean	High income	Aruba	38893.960556	
1	AFE	NaN	NaN	Africa Eastern and Southern	3670.891196	216
2	AFG	South Asia	Low income	Afghanistan	2058.400221	259
3	AFW	NaN	NaN	Africa Western and Central	4115.645480	156
4	AGO	Sub-Saharan Africa	Lower middle income	Angola	7310.896551	204
...	...	...	...	...	...	
260	XKX	Europe & Central Asia	Upper middle income	Kosovo	10436.168846	
261	YEM	Middle East & North Africa	Low income	Yemen, Rep.	3525.790032	18
262	ZAF	Sub-Saharan Africa	Upper middle income	South Africa	13860.270166	110
263	ZMB	Sub-Saharan Africa	Low income	Zambia	3485.021780	152
264	ZWE	Sub-Saharan Africa	Lower middle income	Zimbabwe	2416.049969	226

265 rows × 8 columns

In [43]:

```
#Joining two datasets:
final_df = pd.merge(df4,data5)
```

Inspecting and cleaning the joined datasets

In [44]:

```
# Check for duplicates after merging
final_df.isna().sum()
```

Out[44]:
```
Country_Code                     0
Region                           0
IncomeGroup                      0
Country_Name                     0
gdp_per_capita_2017              7
health_exp_per_capita_2017       7
Beds_per_capita_2017             8
Dr_per_10000_2017               16
Year                             0
Liver_cancer                     0
Kidney_cancer                    0
Lip_and_oral_cavity_cancer       0
Lung_cancer                      0
Larynx_cancer                    0
Gallbladder_cancer               0
Malignant_skin                   0
Leukemia                         0
Multiple_myeloma                 0
Breast_cancer                    0
Prostate_cancer                  0
Thyroid_cancer                   0
Stomach_cancer                   0
Bladder_cancer                   0
Uterine_cancer                   0
Ovarian_cancer                   0
Cervical_cancer                  0
Brain_cancer                     0
Non-Hodgkin_lymphoma             0
Pancreatic_cancer                0
Esophageal_cancer                0
Colon_cancer                     0
Non-melanoma_skin_cancer         0
dtype: int64
```

In [45]:
```
# Deleting column 'Year' and then deleting rows with NaN values
final_df.drop(['Year'],inplace=True, axis=1)
final_df = final_df.dropna()
final_df.reset_index(drop=True, inplace=True)
# final_df.drop(labels=[27, 68, 134],inplace=True) # deleting rows with China, Japan,
```

In [46]:
```
final_df.isna().sum()
```

Out[46]:
```
Country_Code                    0
Region                          0
IncomeGroup                     0
Country_Name                    0
gdp_per_capita_2017             0
health_exp_per_capita_2017      0
Beds_per_capita_2017            0
Dr_per_10000_2017               0
Liver_cancer                    0
Kidney_cancer                   0
Lip_and_oral_cavity_cancer      0
Lung_cancer                     0
Larynx_cancer                   0
Gallbladder_cancer              0
Malignant_skin                  0
Leukemia                        0
Multiple_myeloma                0
Breast_cancer                   0
Prostate_cancer                 0
Thyroid_cancer                  0
Stomach_cancer                  0
Bladder_cancer                  0
Uterine_cancer                  0
Ovarian_cancer                  0
Cervical_cancer                 0
Brain_cancer                    0
Non-Hodgkin_lymphoma            0
Pancreatic_cancer               0
Esophageal_cancer               0
Colon_cancer                    0
Non-melanoma_skin_cancer        0
dtype: int64
```

In [47]:
```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 151 entries, 0 to 150
Data columns (total 31 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Country_Code                  151 non-null    object 
 1   Region                        151 non-null    object 
 2   IncomeGroup                   151 non-null    object 
 3   Country_Name                  151 non-null    object 
 4   gdp_per_capita_2017           151 non-null    float64
 5   health_exp_per_capita_2017    151 non-null    float64
 6   Beds_per_capita_2017          151 non-null    float64
 7   Dr_per_10000_2017             151 non-null    float64
 8   Liver_cancer                  151 non-null    int64  
 9   Kidney_cancer                 151 non-null    int64  
 10  Lip_and_oral_cavity_cancer    151 non-null    int64  
 11  Lung_cancer                   151 non-null    int64  
 12  Larynx_cancer                 151 non-null    int64  
 13  Gallbladder_cancer            151 non-null    int64  
 14  Malignant_skin                151 non-null    int64  
 15  Leukemia                      151 non-null    int64  
 16  Multiple_myeloma              151 non-null    int64  
 17  Breast_cancer                 151 non-null    int64  
 18  Prostate_cancer               151 non-null    int64  
 19  Thyroid_cancer                151 non-null    int64  
 20  Stomach_cancer                151 non-null    int64  
 21  Bladder_cancer                151 non-null    int64  
 22  Uterine_cancer                151 non-null    int64  
 23  Ovarian_cancer                151 non-null    int64  
 24  Cervical_cancer               151 non-null    int64  
 25  Brain_cancer                  151 non-null    int64  
 26  Non-Hodgkin_lymphoma          151 non-null    int64  
 27  Pancreatic_cancer             151 non-null    int64  
 28  Esophageal_cancer             151 non-null    int64  
 29  Colon_cancer                  151 non-null    int64  
 30  Non-melanoma_skin_cancer      151 non-null    int64  
dtypes: float64(4), int64(23), object(4)
memory usage: 36.7+ KB
```

In [48]:

```
final_df.describe()
```

Out[48]:

	gdp_per_capita_2017	health_exp_per_capita_2017	Beds_per_capita_2017	Dr_per_10000_2017	l
count	151.000000	151.000000	151.000000	151.000000	
mean	20301.250152	1446.290694	2.843117	17.810850	
std	20895.387225	1693.170940	2.237102	15.780497	
min	773.572859	57.311356	0.200000	0.208000	
25%	4746.706732	225.507847	1.177273	2.710000	
50%	13429.300622	755.163696	2.210000	13.030000	
75%	28360.785713	2009.294434	3.842500	29.670000	
max	114985.842236	7989.641602	13.050000	61.730000	1

8 rows × 27 columns

In [62]:

```
final_df
```

Out[62]:

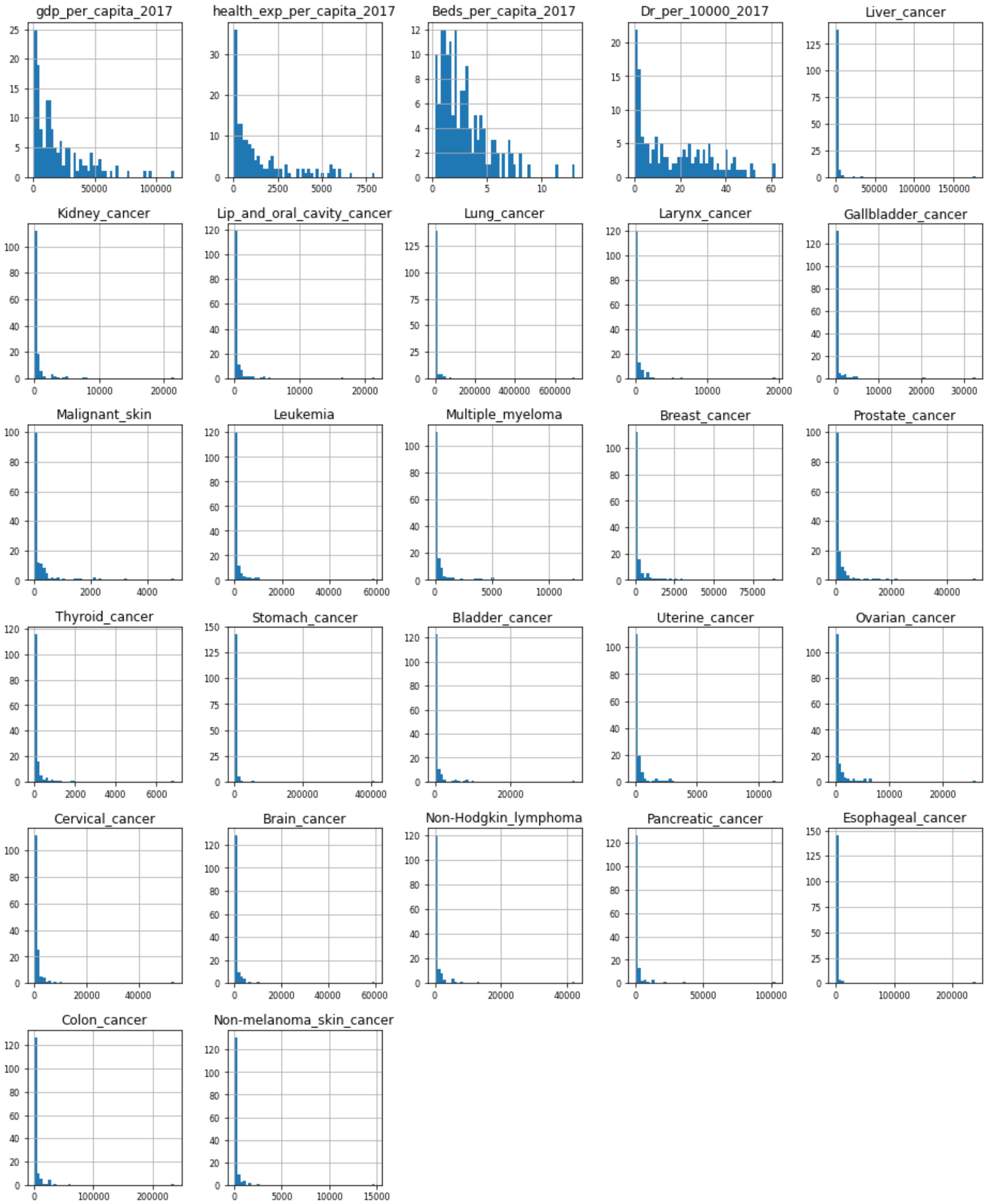
	Country_Code	Region	IncomeGroup	Country_Name	gdp_per_capita_2017	health_exp_per_capi
0	AFG	South Asia	Low income	Afghanistan	2058.400221	259
1	AGO	Sub-Saharan Africa	Lower middle income	Angola	7310.896551	204
2	ALB	Europe & Central Asia	Upper middle income	Albania	12770.964291	639
3	ARE	Middle East & North Africa	High income	United Arab Emirates	67183.605312	2727
4	ARG	Latin America & Caribbean	Upper middle income	Argentina	23597.117753	2470
...	...	...	...	...	...	
146	VUT	East Asia & Pacific	Lower middle income	Vanuatu	3081.461777	86
147	WSM	East Asia & Pacific	Lower middle income	Samoa	6486.108339	361
148	ZAF	Sub-Saharan Africa	Upper middle income	South Africa	13860.270166	1107
149	ZMB	Sub-Saharan Africa	Low income	Zambia	3485.021780	152
150	ZWE	Sub-Saharan Africa	Lower middle income	Zimbabwe	2416.049969	226

148 rows × 31 columns

Histogram for all features

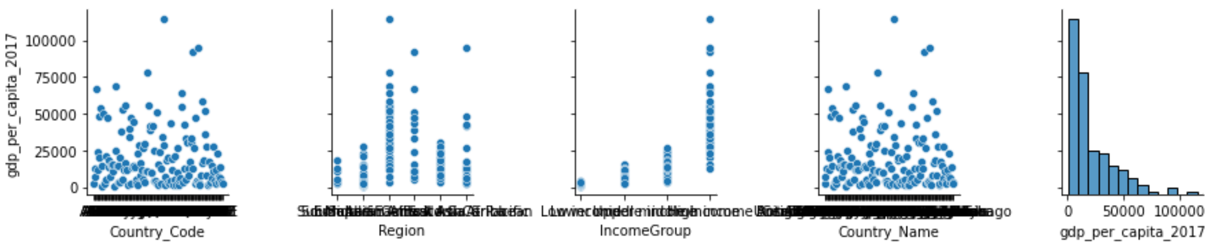
In [49]:

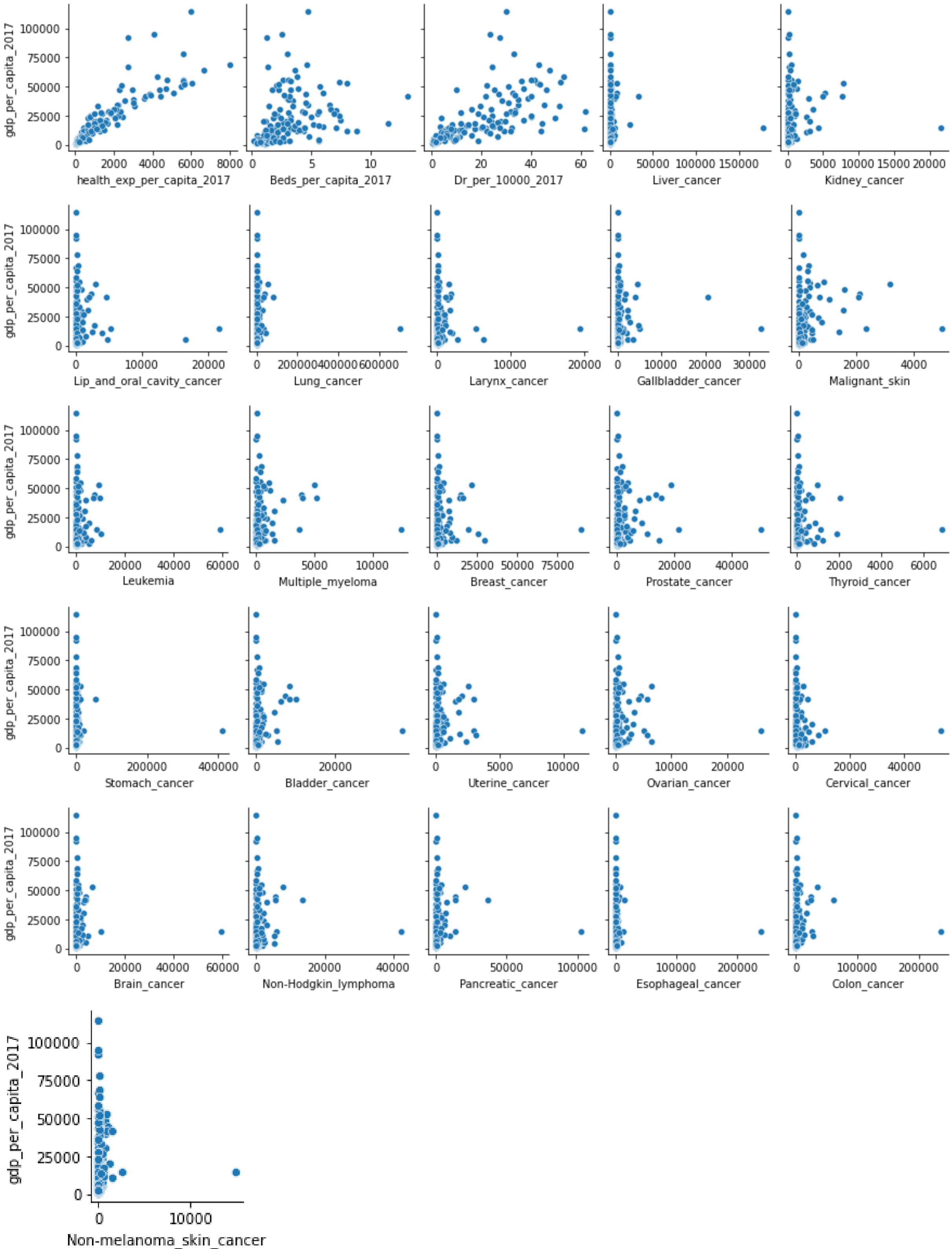
```
final_df.hist(figsize=(16, 20), bins=50, xlabelsize=8, ylabelsize=8);
```



Scatterplots for all features

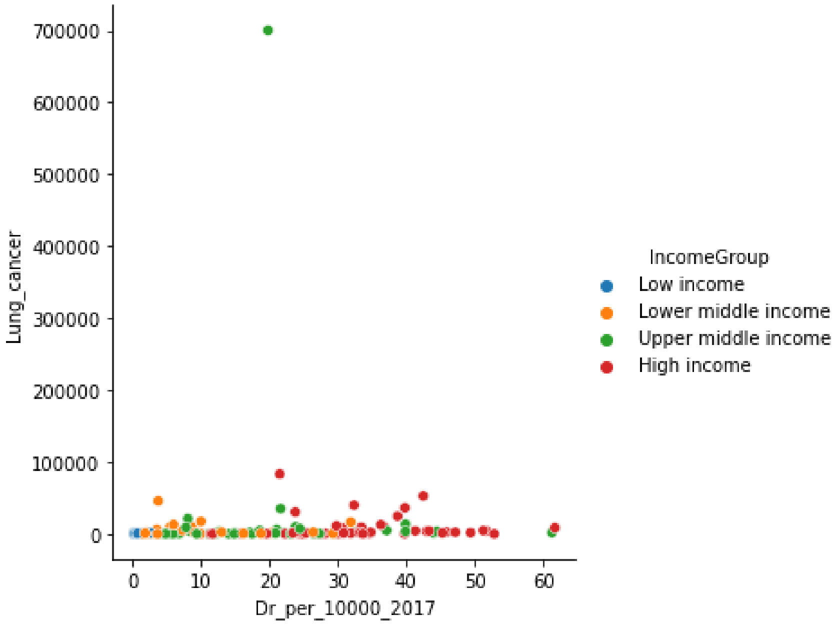
In [50]:
```
for i in range(0, len(final_df.columns), 5):
    sns.pairplot(data=final_df,
                 x_vars=final_df.columns[i:i+5],
                 y_vars=['gdp_per_capita_2017'])
```



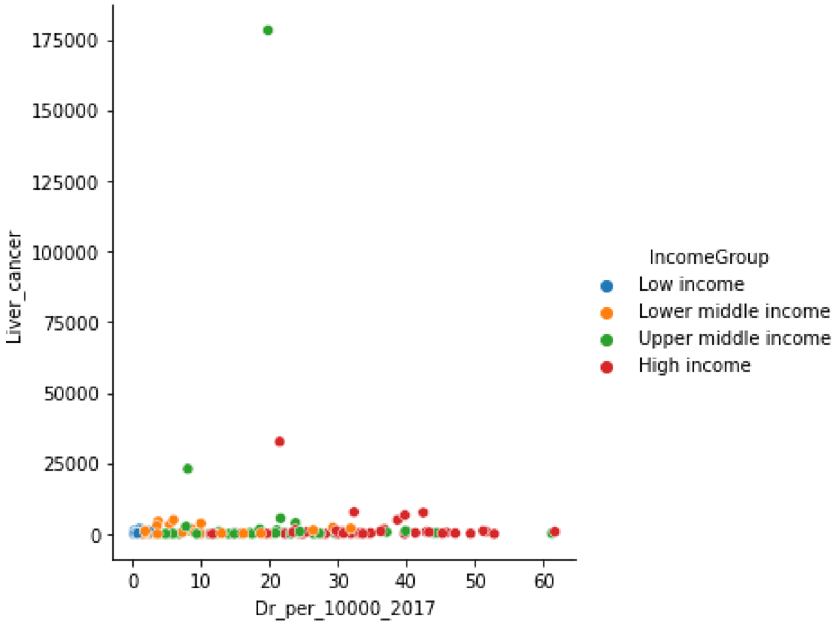


In [51]:
```python
# Scatter plot: Checking outliners
sns.relplot(data=final_df, x="Dr_per_10000_2017", y="Lung_cancer", hue = 'IncomeGroup'
sns.relplot(data=final_df, x="Dr_per_10000_2017", y="Liver_cancer", hue = 'IncomeGroup
sns.relplot(data=final_df, x="gdp_per_capita_2017", y="Lung_cancer", hue = 'IncomeGrou
sns.relplot(data=final_df, x="gdp_per_capita_2017", y="Liver_cancer", hue = 'IncomeGro
```

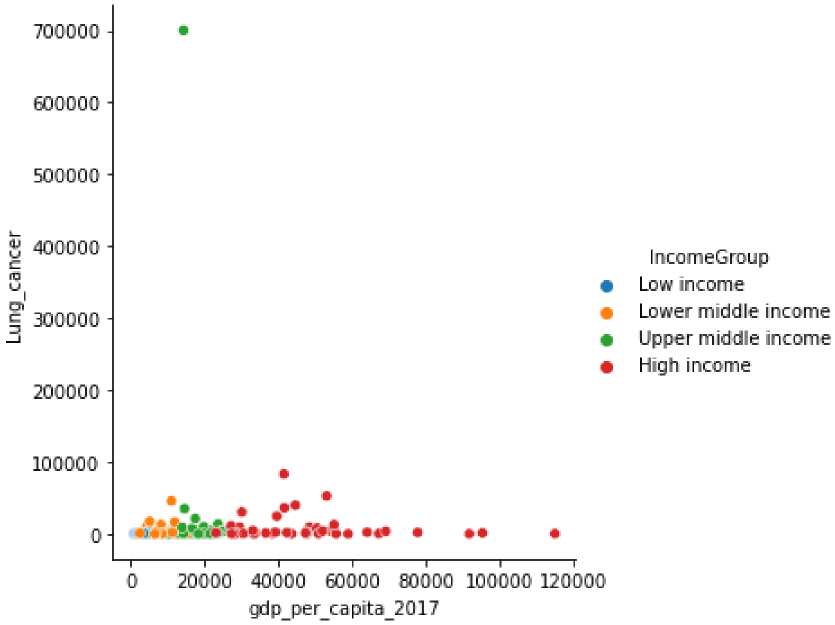
Out[51]: <seaborn.axisgrid.FacetGrid at 0x7fc6eb7fa6a0>



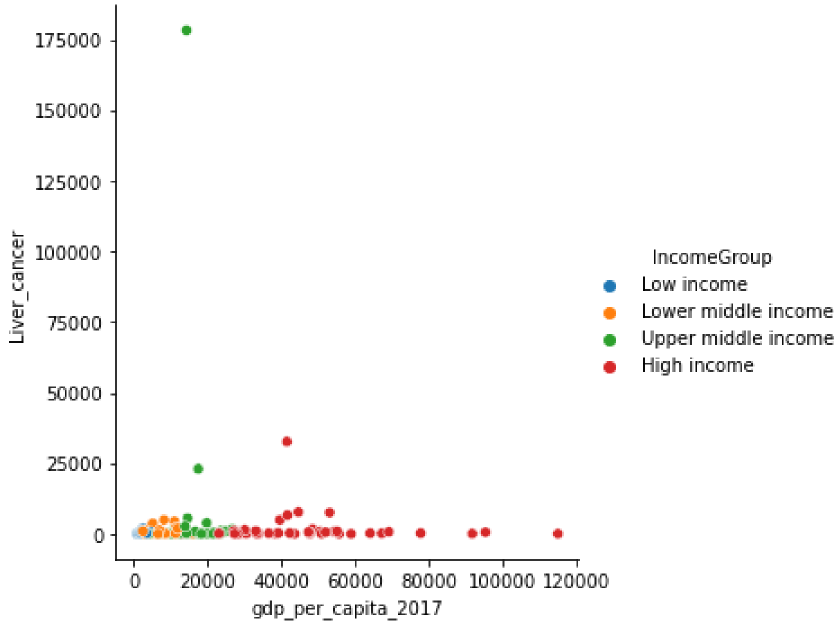
700000
600000
500000
400000
300000
200000
100000
0
Lung_cancer
0
10
20
30
40
50
60
Dr_per_10000_2017
IncomeGroup
Low income
Lower middle income
Upper middle income
High income



175000
150000
125000
100000
75000
50000
25000
0
Liver_cancer
0
10
20
30
40
50
60
Dr_per_10000_2017
IncomeGroup
Low income
Lower middle income
Upper middle income
High income



700000
600000
500000
400000
300000
200000
100000
0
Lung_cancer
0
20000
40000
60000
80000
100000
120000
gdp_per_capita_2017
IncomeGroup
Low income
Lower middle income
Upper middle income
High income



In [52]:
```
# Find outliners
final_df[final_df['Liver_cancer']>20000]
```

Out[52]:

	Country_Code	Region	IncomeGroup	Country_Name	gdp_per_capita_2017	health_exp_per_capita_
27	CHN	East Asia & Pacific	Upper middle income	China	14243.532611	706.36
69	JPN	East Asia & Pacific	High income	Japan	41444.215744	4398.0
135	THA	East Asia & Pacific	Upper middle income	Thailand	17422.952351	669.5

3 rows × 31 columns

In [53]:
```
# Removing outliners
final_df.drop(labels=[27, 68, 134],inplace=True) # deleting rows with China, Japan, &
```

In [54]:
```
# Check impact of outliner removal by describe()
final_df.describe()
```

Out[54]:

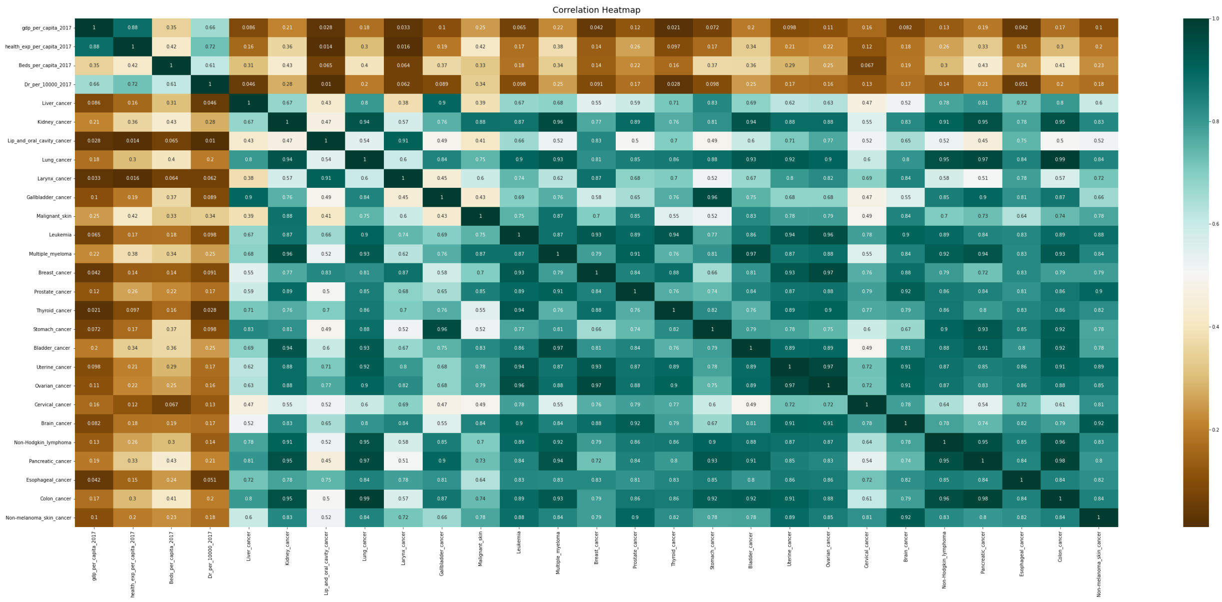
	gdp_per_capita_2017	health_exp_per_capita_2017	Beds_per_capita_2017	Dr_per_10000_2017	
count	148.000000	148.000000	148.000000	148.000000	
mean	20535.333951	1464.556609	2.856289	17.875935	
std	21029.174014	1704.888522	2.247382	15.871256	
min	773.572859	57.311356	0.200000	0.208000	
25%	4820.635048	225.856856	1.188636	2.755000	
50%	13509.504007	756.159393	2.214286	12.968750	
75%	28622.036554	2048.243744	3.833750	29.875000	
max	114985.842236	7989.641602	13.050000	61.730000	3

8 rows × 27 columns

Correlation Matrix: Overall

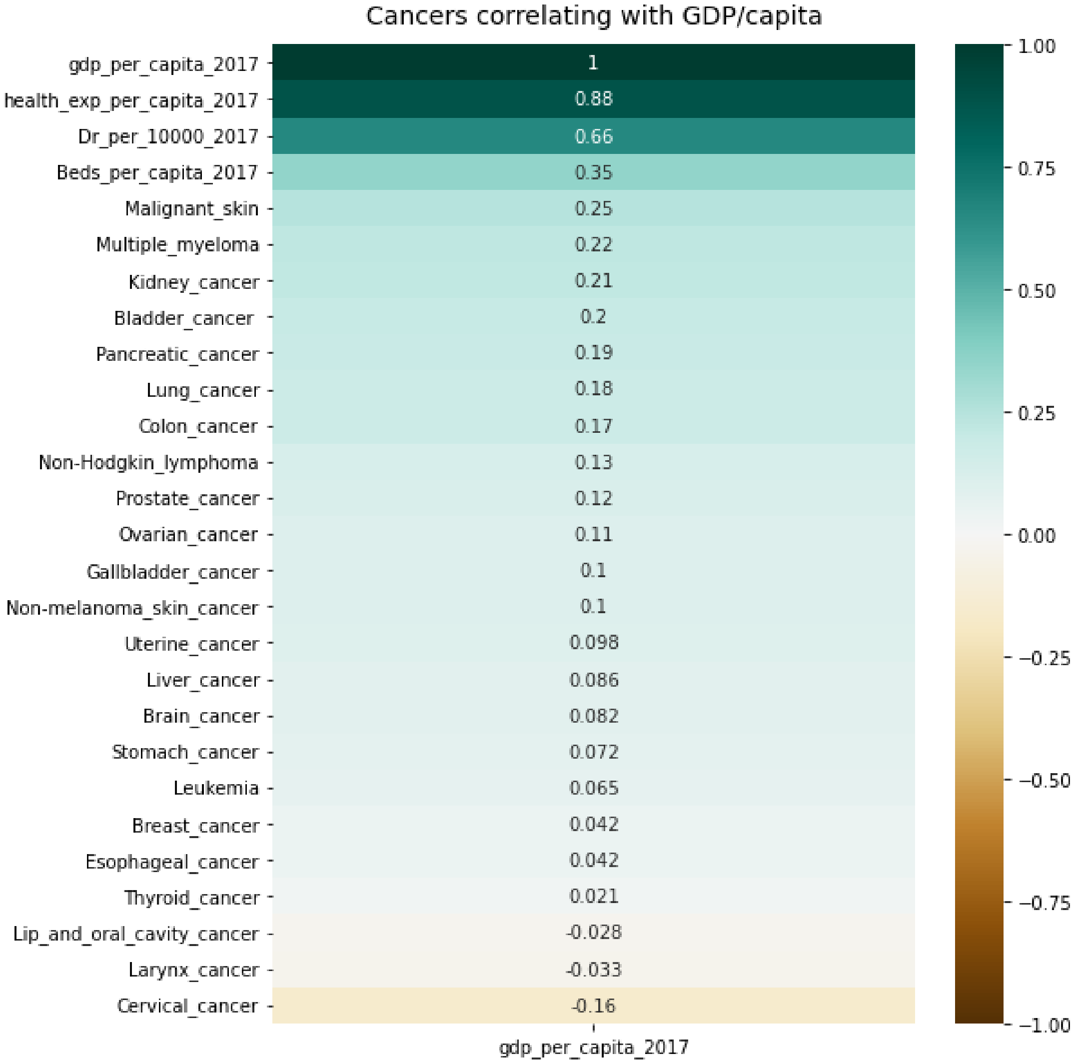
In [55]:
```
# Correlation
plt.figure(figsize=(50, 20))
heatmap = sns.heatmap(final_df.corr().abs(),annot=True, cmap='BrBG')
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':18}, pad=12)
```

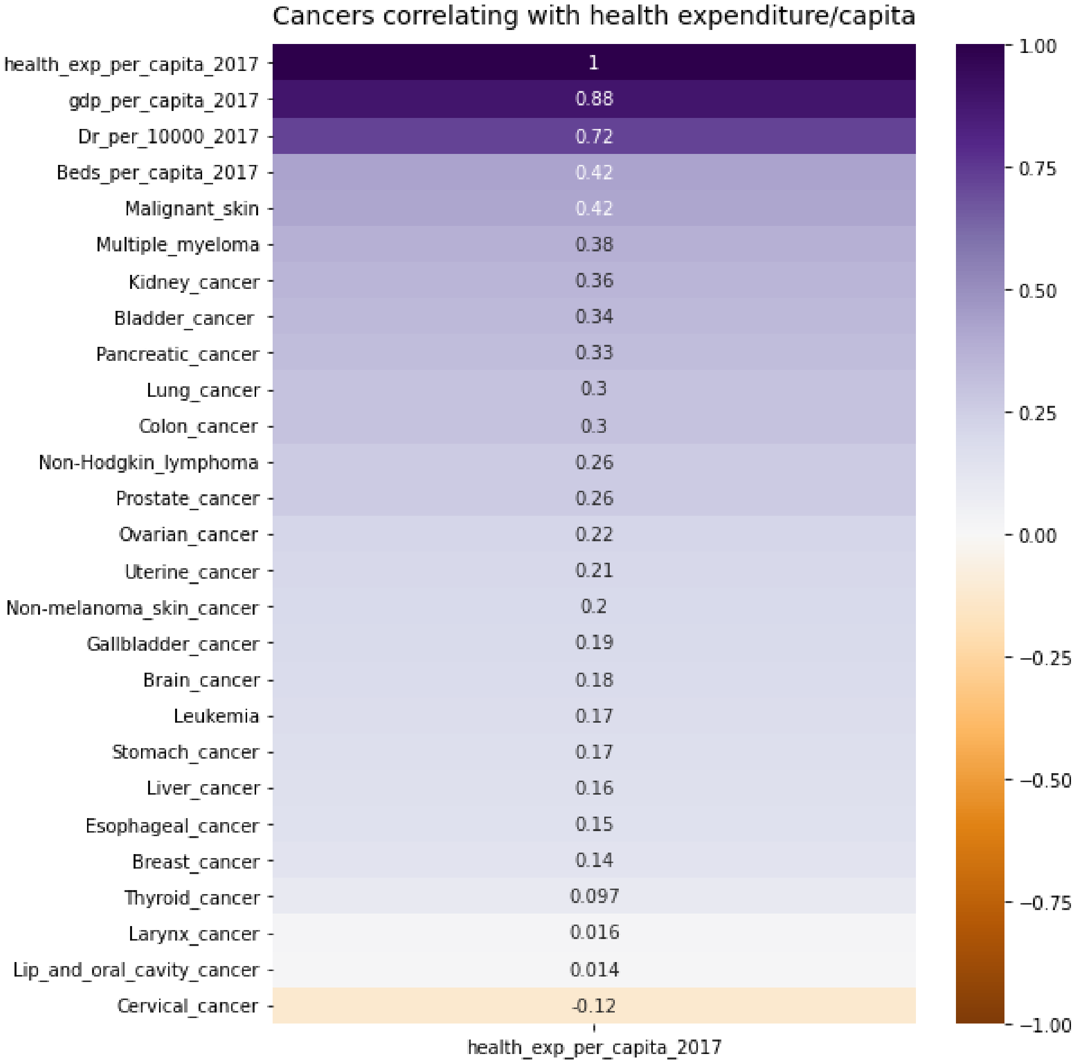
Out[55]:
```
Text(0.5, 1.0, 'Correlation Heatmap')
```

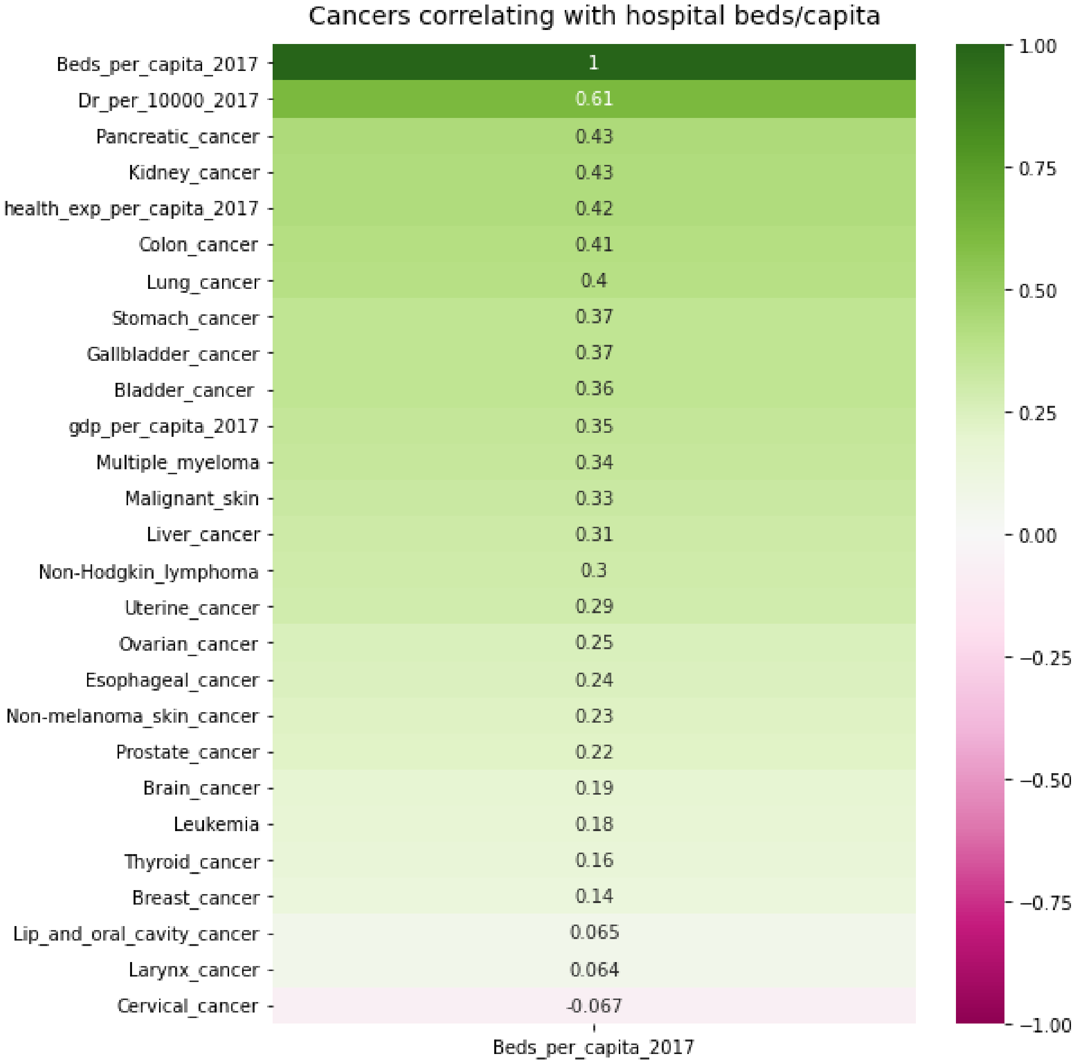


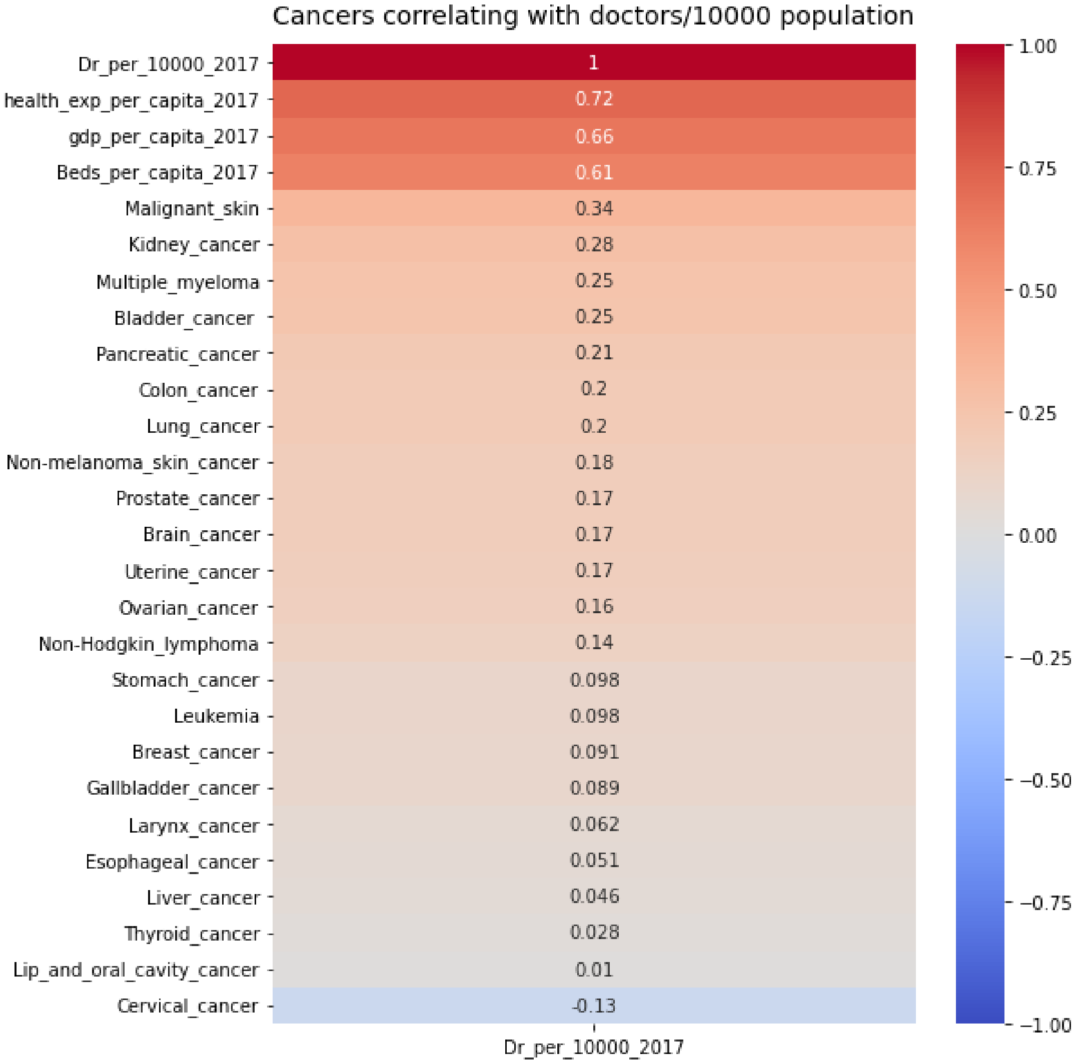
Correlation: With Individual Categorical

In [56]:
```
# Correlation
# final_df.drop(final_df.columns[[0, 1, 2, 3, 4, 5, 6, 7]], inplace=False, axis=1)

plt.figure(figsize=(8, 10))
heatmap = sns.heatmap(final_df.corr()[['gdp_per_capita_2017']].sort_values(by='gdp_per
heatmap.set_title('Cancers correlating with GDP/capita', fontdict={'fontsize':14}, pad

plt.figure(figsize=(8, 10))
heatmap = sns.heatmap(final_df.corr()[['health_exp_per_capita_2017']].sort_values(by='
heatmap.set_title('Cancers correlating with health expenditure/capita', fontdict={'fon

plt.figure(figsize=(8, 10))
heatmap = sns.heatmap(final_df.corr()[['Beds_per_capita_2017']].sort_values(by='Beds_p
heatmap.set_title('Cancers correlating with hospital beds/capita', fontdict={'fontsize

plt.figure(figsize=(8, 10))
heatmap = sns.heatmap(final_df.corr()[['Dr_per_10000_2017']].sort_values(by='Dr_per_10
heatmap.set_title('Cancers correlating with doctors/10000 population', fontdict={'font
```


Cancers correlating with GDP/capita
gdp_per_capita_2017
health_exp_per_capita_2017
Dr_per_10000_2017
Beds_per_capita_2017
Malignant_skin
Multiple_myeloma
Kidney_cancer
Bladder_cancer
Pancreatic_cancer
Lung_cancer
Colon_cancer
Non-Hodgkin_lymphoma
Prostate_cancer
Ovarian_cancer
Gallbladder_cancer
Non-melanoma_skin_cancer
Uterine_cancer
Liver_cancer
Brain_cancer
Stomach_cancer
Leukemia
Breast_cancer
Esophageal_cancer
Thyroid_cancer
Lip_and_oral_cavity_cancer
Larynx_cancer
Cervical_cancer
1
0.88
0.66
0.35
0.25
0.22
0.21
0.2
0.19
0.18
0.17
0.13
0.12
0.11
0.1
0.1
0.098
0.086
0.082
0.072
0.065
0.042
0.042
0.021
-0.028
-0.033
-0.16
gdp_per_capita_2017
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


Cancers correlating with health expenditure/capita
health_exp_per_capita_2017 1
gdp_per_capita_2017 0.88
Dr_per_10000_2017 0.72
Beds_per_capita_2017 0.42
Malignant_skin 0.42
Multiple_myeloma 0.38
Kidney_cancer 0.36
Bladder_cancer 0.34
Pancreatic_cancer 0.33
Lung_cancer 0.3
Colon_cancer 0.3
Non-Hodgkin_lymphoma 0.26
Prostate_cancer 0.26
Ovarian_cancer 0.22
Uterine_cancer 0.21
Non-melanoma_skin_cancer 0.2
Gallbladder_cancer 0.19
Brain_cancer 0.18
Leukemia 0.17
Stomach_cancer 0.17
Liver_cancer 0.16
Esophageal_cancer 0.15
Breast_cancer 0.14
Thyroid_cancer 0.097
Larynx_cancer 0.016
Lip_and_oral_cavity_cancer 0.014
Cervical_cancer -0.12
health_exp_per_capita_2017
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


Cancers correlating with hospital beds/capita
Beds_per_capita_2017 1
Dr_per_10000_2017 0.61
Pancreatic_cancer 0.43
Kidney_cancer 0.43
health_exp_per_capita_2017 0.42
Colon_cancer 0.41
Lung_cancer 0.4
Stomach_cancer 0.37
Gallbladder_cancer 0.37
Bladder_cancer 0.36
gdp_per_capita_2017 0.35
Multiple_myeloma 0.34
Malignant_skin 0.33
Liver_cancer 0.31
Non-Hodgkin_lymphoma 0.3
Uterine_cancer 0.29
Ovarian_cancer 0.25
Esophageal_cancer 0.24
Non-melanoma_skin_cancer 0.23
Prostate_cancer 0.22
Brain_cancer 0.19
Leukemia 0.18
Thyroid_cancer 0.16
Breast_cancer 0.14
Lip_and_oral_cavity_cancer 0.065
Larynx_cancer 0.064
Cervical_cancer -0.067
Beds_per_capita_2017
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

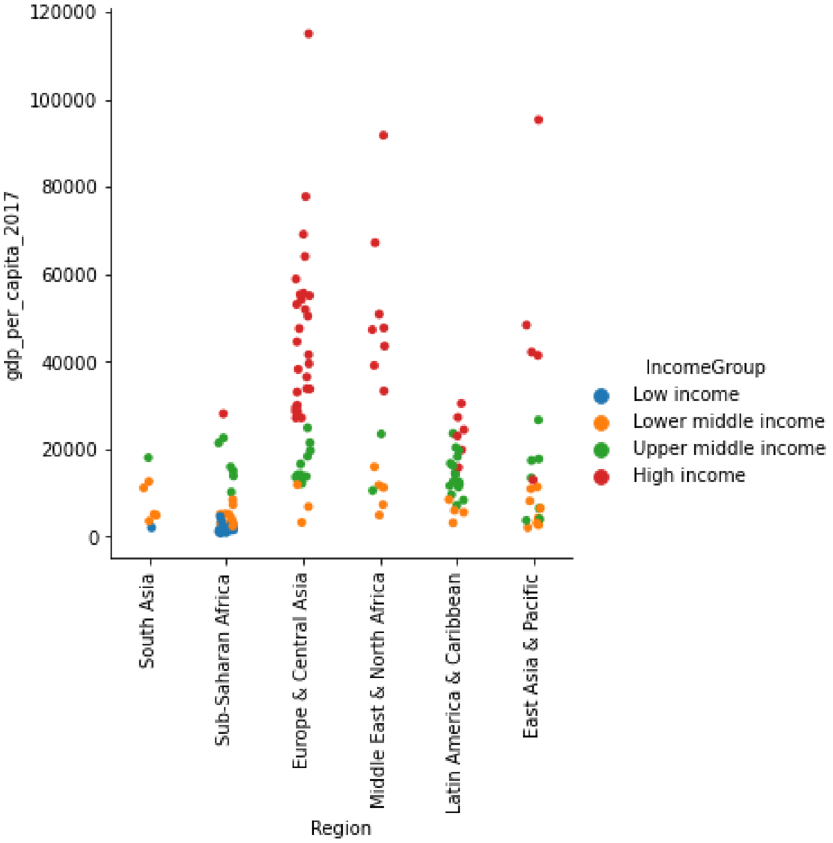


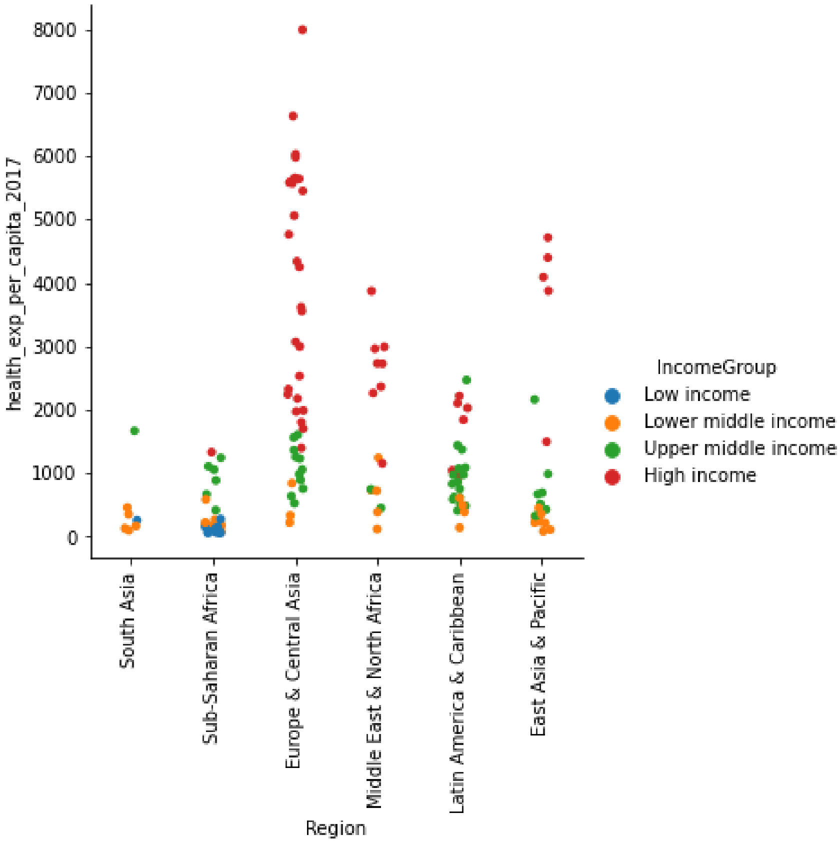
Categorical Scatterplots: Across Region

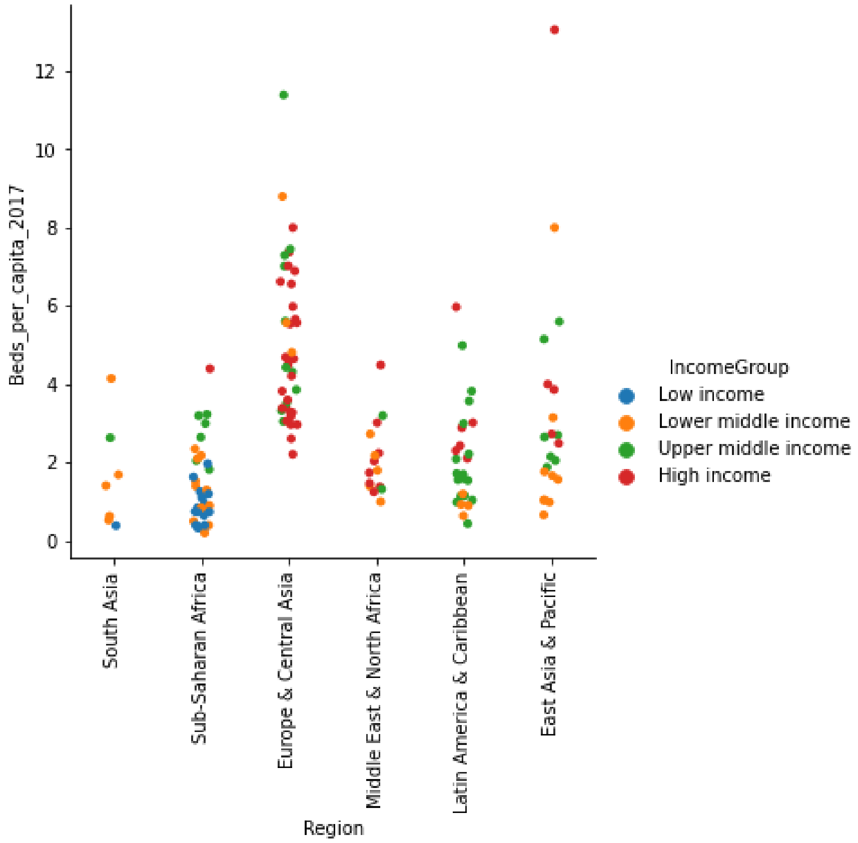
In [57]:
```
# final_df.drop(final_df.columns[[0, 1, 2, 3, 4, 5, 6, 7]], inplace=False, axis=1)
# final_df
```

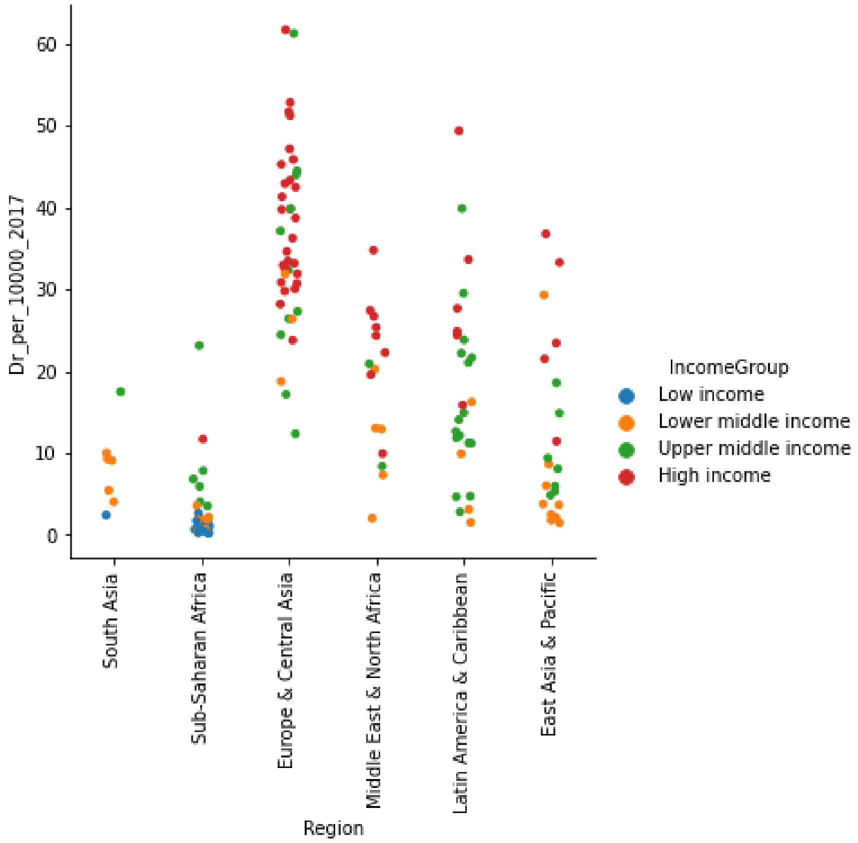
In [58]:
```
chart1 = sns.catplot(data=final_df, x="Region", y="gdp_per_capita_2017", hue="IncomeGr
chart1.set_xticklabels(rotation=90)

chart2 = sns.catplot(data=final_df, x="Region", y="health_exp_per_capita_2017", hue="I
chart2.set_xticklabels(rotation=90)

chart3 = sns.catplot(data=final_df, x="Region", y="Beds_per_capita_2017", hue="IncomeG
chart3.set_xticklabels(rotation=90)

chart4 = sns.catplot(data=final_df, x="Region", y="Dr_per_10000_2017", hue="IncomeGrou
chart4.set_xticklabels(rotation=90)
```

Out[58]: <seaborn.axisgrid.FacetGrid at 0x7fc7187682e0>


gdp_per_capita_2017
120000
100000
80000
60000
40000
20000
0
South Asia
Sub-Saharan Africa
Europe & Central Asia
Middle East & North Africa
Latin America & Caribbean
East Asia & Pacific
Region
IncomeGroup
Low income
Lower middle income
Upper middle income
High income


health_exp_per_capita_2017
8000
7000
6000
5000
4000
3000
2000
1000
0
South Asia
Sub-Saharan Africa
Europe & Central Asia
Middle East & North Africa
Latin America & Caribbean
East Asia & Pacific
Region
IncomeGroup
Low income
Lower middle income
Upper middle income
High income



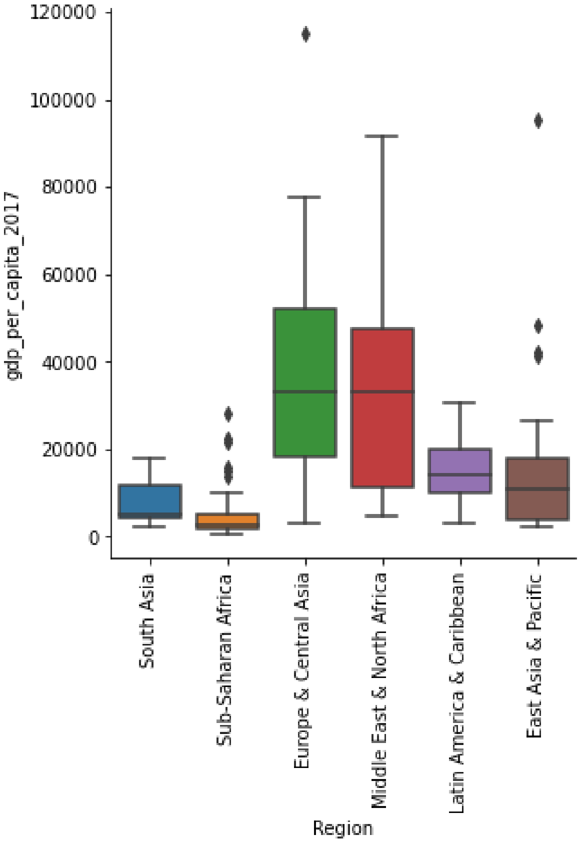


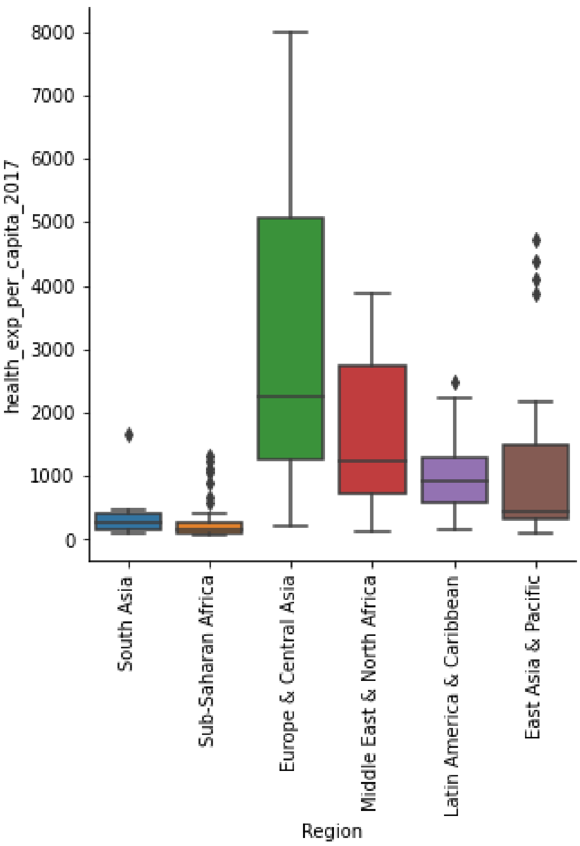
Boxplot Across Region

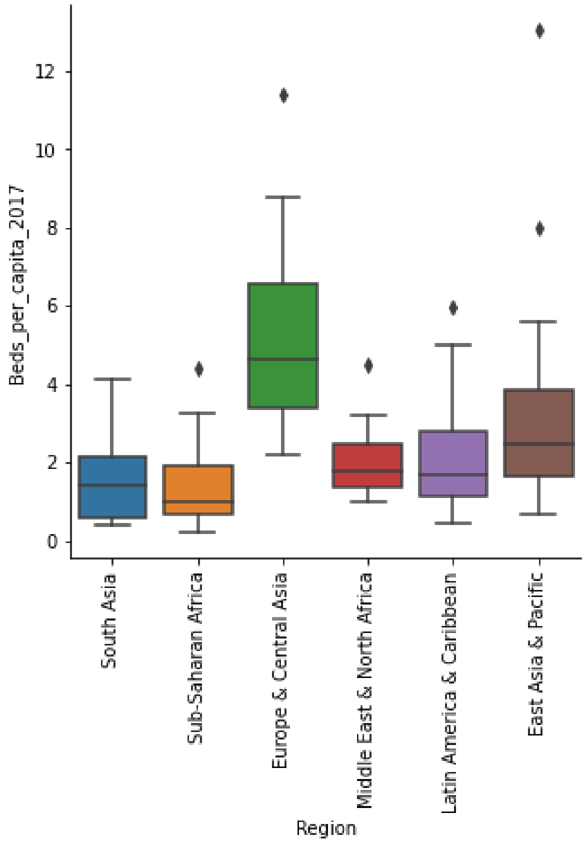
In [59]:
```
chart1 = sns.catplot(data=final_df, x="Region", y="gdp_per_capita_2017", kind="box")
chart1.set_xticklabels(rotation=90)

chart2 = sns.catplot(data=final_df, x="Region", y="health_exp_per_capita_2017", kind="
chart2.set_xticklabels(rotation=90)

chart3 = sns.catplot(data=final_df, x="Region", y="Beds_per_capita_2017", kind="box")
```

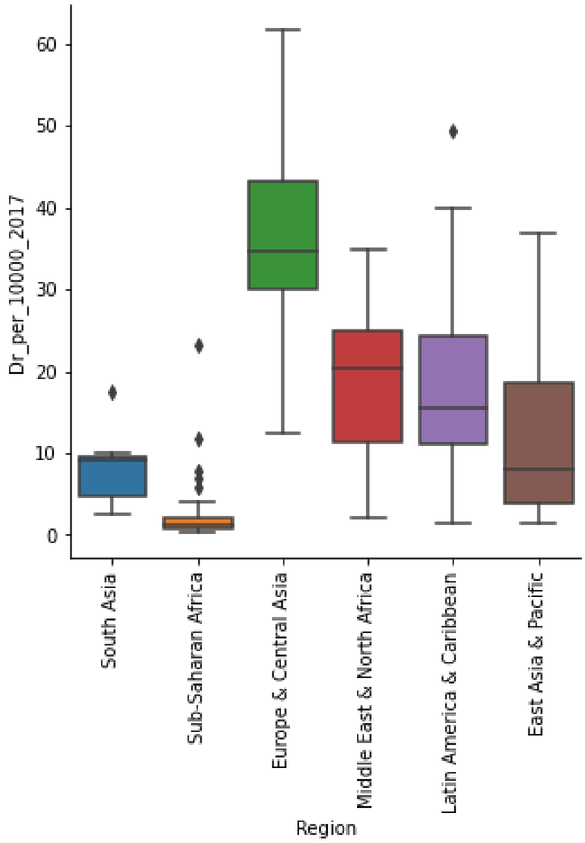
```
chart3.set_xticklabels(rotation=90)

chart4 = sns.catplot(data=final_df, x="Region", y="Dr_per_10000_2017", kind="box")
chart4.set_xticklabels(rotation=90)
```

Out[59]: <seaborn.axisgrid.FacetGrid at 0x7fc7187ab3a0>





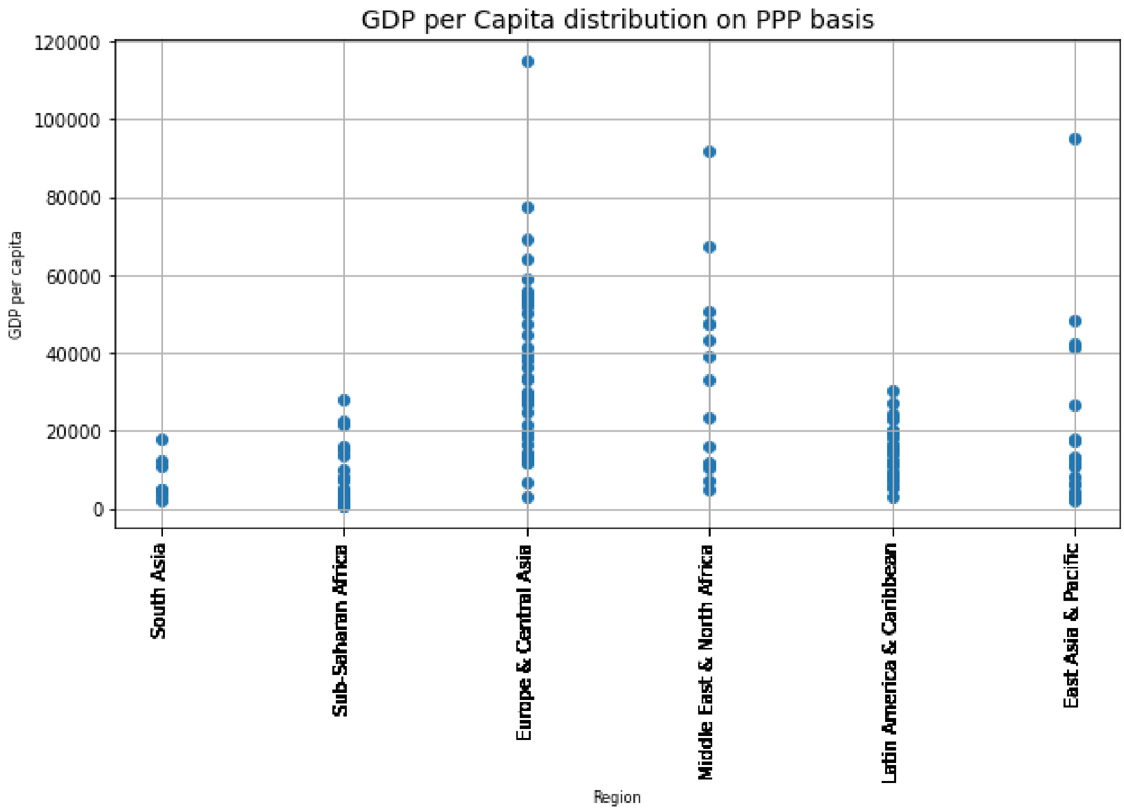


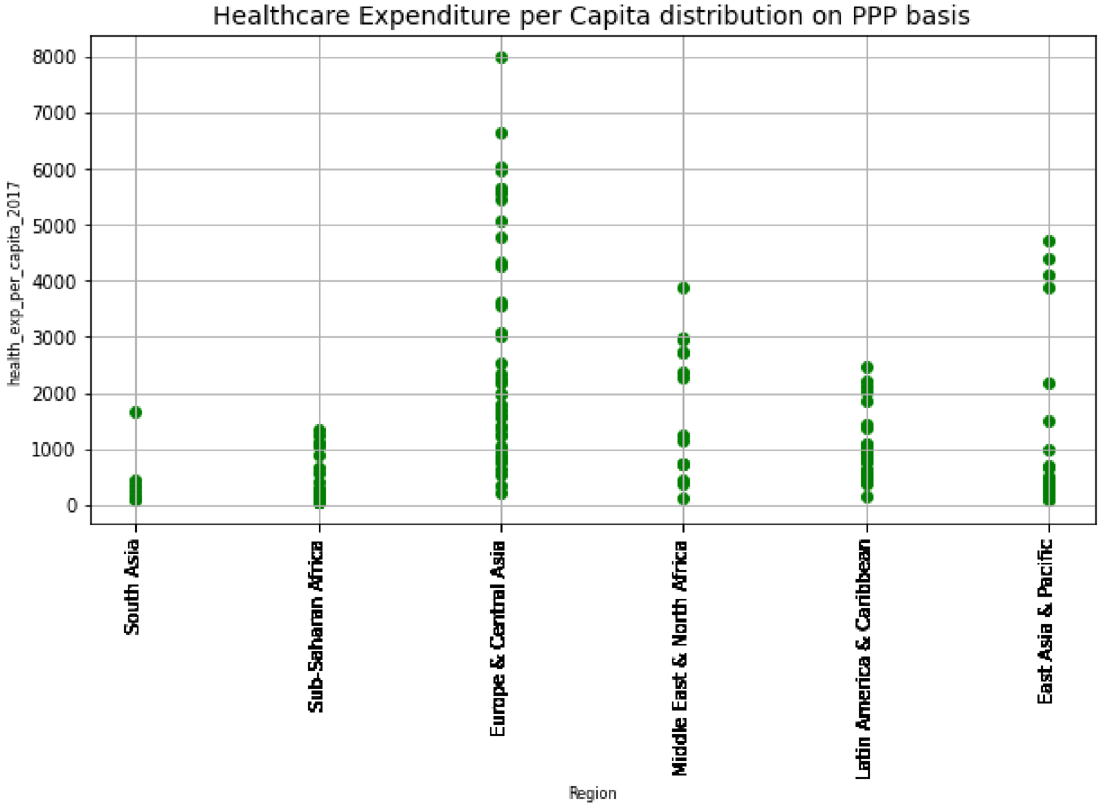


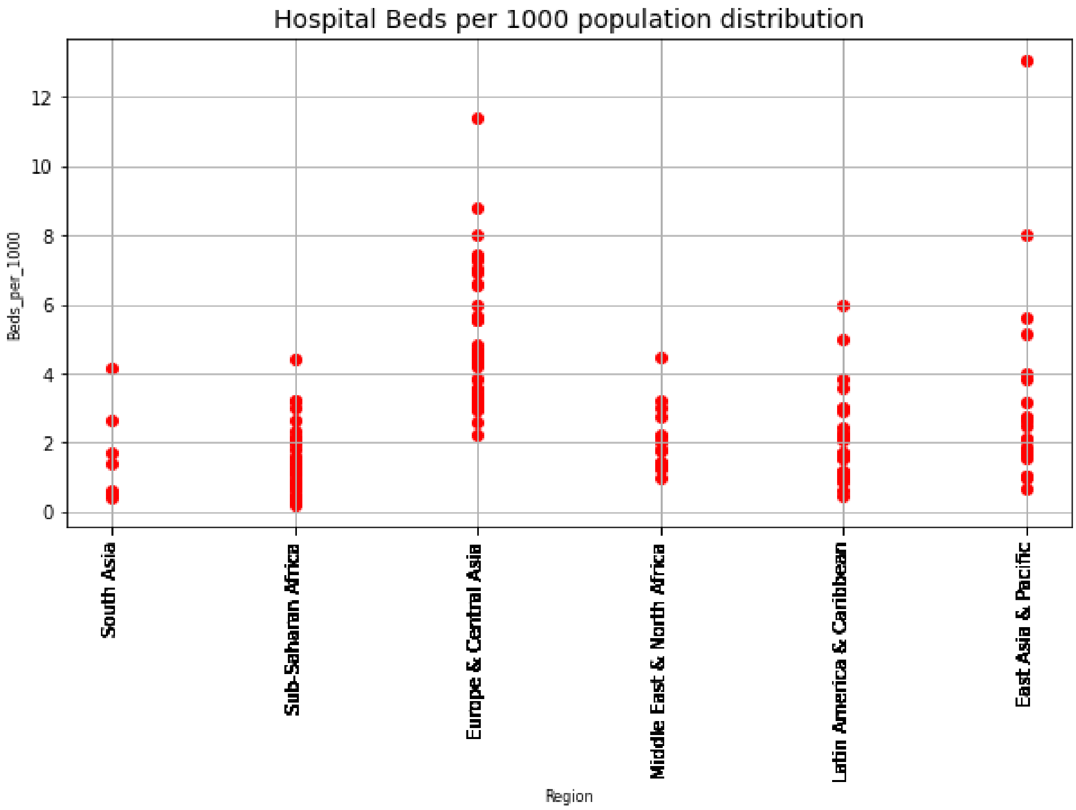
Distribution Across Region

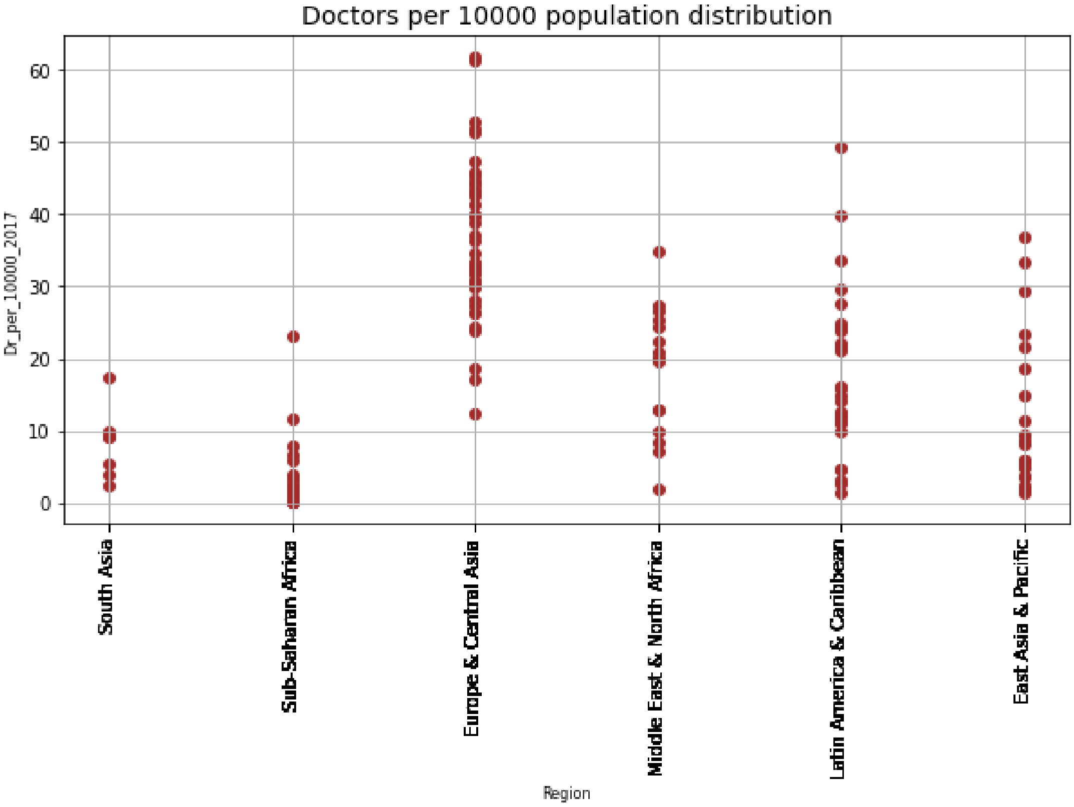
In [60]:
```
plt.figure(figsize=(10,5))
# plt.bar(final_df['Region'],final_df['gdp_per_capita_2017'],color='green') ##bar grap
plt.scatter(final_df['Region'],final_df['gdp_per_capita_2017'])#,color='green', alpha=
plt.xlabel("Region", fontsize=8)
plt.ylabel("GDP per capita", fontsize=8)
plt.title("GDP per Capita distribution on PPP basis", fontsize=14)
plt.grid(True)
```

```
plt.xticks(final_df['Region'], rotation='vertical', size=10)
plt.show();

plt.figure(figsize=(10,5))
# plt.bar(final_df['Region'],final_df['gdp_per_capita_2017'],color='green') ##bar grap
plt.scatter(final_df['Region'],final_df['health_exp_per_capita_2017'], color='green')#
plt.xlabel("Region", fontsize=8)
plt.ylabel("health_exp_per_capita_2017", fontsize=8)
plt.title("Healthcare Expenditure per Capita distribution on PPP basis", fontsize=14)
plt.grid(True)
plt.xticks(final_df['Region'], rotation='vertical', size=10)
plt.show();

plt.figure(figsize=(10,5))
# plt.bar(final_df['Region'],final_df['gdp_per_capita_2017'],color='green') ##bar grap
plt.scatter(final_df['Region'],final_df['Beds_per_capita_2017'], color='red')#, alpha=
plt.xlabel("Region", fontsize=8)
plt.ylabel("Beds_per_1000", fontsize=8)
plt.title("Hospital Beds per 1000 population distribution", fontsize=14)
plt.grid(True)
plt.xticks(final_df['Region'], rotation='vertical', size=10)
plt.show();

plt.figure(figsize=(10,5))
# plt.bar(final_df['Region'],final_df['gdp_per_capita_2017'],color='green') ##bar grap
plt.scatter(final_df['Region'],final_df['Dr_per_10000_2017'], color='brown')#, alpha=0
plt.xlabel("Region", fontsize=8)
plt.ylabel("Dr_per_10000_2017", fontsize=8)
plt.title("Doctors per 10000 population distribution", fontsize=14)
plt.grid(True)
plt.xticks(final_df['Region'], rotation='vertical', size=10)
plt.show();
```




Healthcare Expenditure per Capita distribution on PPP basis
health_exp_per_capita_2017
8000
7000
6000
5000
4000
3000
2000
1000
0
South Asia
Sub-Saharan Africa
Europe & Central Asia
Middle East & North Africa
Latin America & Caribbean
East Asia & Pacific
Region


Hospital Beds per 1000 population distribution
Beds_per_1000
12
10
8
6
4
2
0
South Asia
Sub-Saharan Africa
Europe & Central Asia
Middle East & North Africa
Latin America & Caribbean
East Asia & Pacific
Region


Doctors per 10000 population distribution
Dr_per_10000_2017
0
10
20
30
40
50
60
South Asia
Sub-Saharan Africa
Europe & Central Asia
Middle East & North Africa
Latin America & Caribbean
East Asia & Pacific
Region

In [ ]: