Importing necessary Libraries

In [65]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
import plotly.graph_objs as go
from plotly.offline import iplot
```

In [66]:

```
pd.set_option('display.max_rows',None)
```

importing youth progress index.CSV data

In [67]:

```
data = pd.read_csv("C:\\Users\\jesro\\Desktop\\Youth Progress Index.csv",encoding='ISO-8
data.head()
```

Out[67]:

	Country	YPI country code	YPI year	Status	Youth Progress Index	Basic Human Needs	Foundations of Wellbeing	Opportunity	Nutrition and Basic Medical Care	Water and Sanitation	Shelter
0	World	WWW	2020	NaN	65.78	76.30	62.72	58.31	83.60	72.02	82.40
1	World	WWW	2019	NaN	65.53	75.94	62.76	57.90	83.32	71.49	81.91
2	World	WWW	2018	NaN	65.34	75.41	63.01	57.62	83.01	70.71	81.17
3	World	WWW	2017	NaN	64.86	75.07	62.57	56.93	82.60	70.05	80.71
4	World	WWW	2016	NaN	64.19	74.53	61.21	56.83	82.16	69.65	79.80

In [68]:

```
data.dtypes
```

Out[68]:

```
Country                                    object
YPI country code                           object
YPI year                                    int64
Status                                     object
Youth Progress Index                      float64
Basic Human Needs                         float64
Foundations of Wellbeing                  float64
Opportunity                               float64
Nutrition and Basic Medical Care          float64
Water and Sanitation                      float64
Shelter                                   float64
Personal Safety                           float64
Access to Basic Knowledge                 float64
Access to Information and Communications  float64
Health and Wellness                       float64
Environmental Quality                     float64
Personal Rights                           float64
Personal Freedom and Choice               float64
Inclusiveness                             float64
Access to Advanced Education              float64
dtype: object
```

In [69]:

```
data.isnull().sum()
```

Out[69]:

```
Country                                    0
YPI country code                           0
YPI year                                       0
Status                                        10
Youth Progress Index                         478
Basic Human Needs                            474
Foundations of Wellbeing                     283
Opportunity                                  436
Nutrition and Basic Medical Care             177
Water and Sanitation                         283
Shelter                                      293
Personal Safety                              464
Access to Basic Knowledge                    226
Access to Information and Communications     217
Health and Wellness                          210
Environmental Quality                        283
Personal Rights                              240
Personal Freedom and Choice                  410
Inclusiveness                                421
Access to Advanced Education                 211
dtype: int64
```

Creating a copy of dataframe

In [70]:
```
df = data.copy()
df.head()
```

Out[70]:

	Country	YPI country code	YPI year	Status	Youth Progress Index	Basic Human Needs	Foundations of Wellbeing	Opportunity	Nutrition and Basic Medical Care	Water and Sanitation	Shelter
0	World	WWW	2020	NaN	65.78	76.30	62.72	58.31	83.60	72.02	82.40
1	World	WWW	2019	NaN	65.53	75.94	62.76	57.90	83.32	71.49	81.91
2	World	WWW	2018	NaN	65.34	75.41	63.01	57.62	83.01	70.71	81.17
3	World	WWW	2017	NaN	64.86	75.07	62.57	56.93	82.60	70.05	80.71
4	World	WWW	2016	NaN	64.19	74.53	61.21	56.83	82.16	69.65	79.80

In [71]:
```
df.head()
```

Out[71]:

	Country	YPI country code	YPI year	Status	Youth Progress Index	Basic Human Needs	Foundations of Wellbeing	Opportunity	Nutrition and Basic Medical Care	Water and Sanitation	Shelter
0	World	WWW	2020	NaN	65.78	76.30	62.72	58.31	83.60	72.02	82.40
1	World	WWW	2019	NaN	65.53	75.94	62.76	57.90	83.32	71.49	81.91
2	World	WWW	2018	NaN	65.34	75.41	63.01	57.62	83.01	70.71	81.17
3	World	WWW	2017	NaN	64.86	75.07	62.57	56.93	82.60	70.05	80.71
4	World	WWW	2016	NaN	64.19	74.53	61.21	56.83	82.16	69.65	79.80

In [72]:
```
df.drop(df[df['Country'] == 'World'].index, inplace = True)
```

In [73]:
```
df.reset_index(drop = True,inplace = True)
```

In [74]:

```
df.rename(columns = {'Youth Progress Index':'YPI_score'}, inplace = True)
```

In [75]:

```
df.head()
```

Out[75]:

	Country	YPI country code	YPI year	Status	YPI_score	Basic Human Needs	Foundations of Wellbeing	Opportunity	Nutrition and Basic Medical Care	Water and Sanitation	Shelt
0	Afghanistan	AFG	2020	Ranked	31.24	42.19	29.17	22.34	47.21	44.78	62.
1	Afghanistan	AFG	2019	Ranked	29.83	40.41	28.67	20.42	47.10	42.46	61.
2	Afghanistan	AFG	2018	Ranked	29.99	40.79	29.60	19.57	47.20	46.92	62.
3	Afghanistan	AFG	2017	Ranked	30.50	44.11	27.68	19.73	46.73	49.65	50.
4	Afghanistan	AFG	2016	Ranked	29.30	44.04	26.92	16.94	44.97	46.92	54.

In [76]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1960 entries, 0 to 1959
Data columns (total 20 columns):
 #   Column                                     Non-Null Count  Dtype  
---  ------                                     --------------  -----  
 0   Country                                    1960 non-null   object 
 1   YPI country code                           1960 non-null   object 
 2   YPI year                                   1960 non-null   int64  
 3   Status                                     1960 non-null   object 
 4   YPI_score                                  1482 non-null   float64
 5   Basic Human Needs                          1486 non-null   float64
 6   Foundations of Wellbeing                   1677 non-null   float64
 7   Opportunity                                1524 non-null   float64
 8   Nutrition and Basic Medical Care           1783 non-null   float64
 9   Water and Sanitation                       1677 non-null   float64
 10  Shelter                                    1667 non-null   float64
 11  Personal Safety                            1496 non-null   float64
 12  Access to Basic Knowledge                  1734 non-null   float64
 13  Access to Information and Communications   1743 non-null   float64
 14  Health and Wellness                        1750 non-null   float64
 15  Environmental Quality                      1677 non-null   float64
 16  Personal Rights                            1720 non-null   float64
 17  Personal Freedom and Choice                1550 non-null   float64
 18  Inclusiveness                              1539 non-null   float64
 19  Access to Advanced Education               1749 non-null   float64
dtypes: float64(16), int64(1), object(3)
memory usage: 306.4+ KB
```

Drop null values

In [77]:

```
df.dropna(inplace = True)
```

In [78]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1482 entries, 0 to 1482
Data columns (total 20 columns):
 #   Column                                     Non-Null Count  Dtype  
---  ------                                     --------------  -----  
 0   Country                                    1482 non-null   object 
 1   YPI country code                           1482 non-null   object 
 2   YPI year                                   1482 non-null   int64  
 3   Status                                      1482 non-null   object
 4   YPI_score                                   1482 non-null   float64
 5   Basic Human Needs                           1482 non-null   float64
 6   Foundations of Wellbeing                    1482 non-null   float64
 7   Opportunity                                 1482 non-null   float64
 8   Nutrition and Basic Medical Care            1482 non-null   float64
 9   Water and Sanitation                        1482 non-null   float64
 10  Shelter                                     1482 non-null   float64
 11  Personal Safety                             1482 non-null   float64
 12  Access to Basic Knowledge                   1482 non-null   float64
 13  Access to Information and Communications    1482 non-null   float64
 14  Health and Wellness                         1482 non-null   float64
 15  Environmental Quality                       1482 non-null   float64
 16  Personal Rights                             1482 non-null   float64
 17  Personal Freedom and Choice                 1482 non-null   float64
 18  Inclusiveness                               1482 non-null   float64
 19  Access to Advanced Education                1482 non-null   float64
dtypes: float64(16), int64(1), object(3)
memory usage: 243.1+ KB
```

In [79]:

```
df.describe()
```

Out[79]:

	YPI year	YPI_score	Basic Human Needs	Foundations of Wellbeing	Opportunity	Nutrition and Basic Medical Care	Water and Sanitation	Shelter
count	1482.000000	1482.000000	1482.000000	1482.000000	1482.000000	1482.000000	1482.000000	1482.000000
mean	2015.506748	62.392557	70.120526	61.869109	55.188009	77.509190	68.029669	77.001592
std	2.879347	18.023526	20.843847	17.635276	18.538939	23.146816	25.383249	24.917941
min	2011.000000	18.760000	7.830000	18.920000	11.630000	0.000000	0.000000	12.840000
25%	2013.000000	47.882500	54.110000	48.342500	42.465000	58.085000	48.252500	57.930000
50%	2016.000000	63.685000	77.450000	61.610000	52.610000	88.495000	75.175000	91.540000
75%	2018.000000	76.977500	86.860000	75.247500	68.292500	96.767500	89.197500	95.510000
max	2020.000000	96.560000	99.220000	96.960000	96.500000	100.000000	100.000000	100.000000

Displaying top countries with SPI score greater than 90

In [80]:

```
top_country = df[df['YPI_score']>90].sort_values(by = ['YPI_score'], ascending = False)
top_country
```

Out[80]:

	Country	YPI country code	YPI year	Status	YPI_score	Basic Human Needs	Foundations of Wellbeing	Opportunity	Nutrition and Basic Medical Care	Water and Sanitation
1009	Norway	NOR	2018	Ranked	96.56	96.22	96.96	96.50	99.82	99.25
1008	Norway	NOR	2019	Ranked	95.97	95.80	96.54	95.59	99.83	99.06
1010	Norway	NOR	2017	Ranked	95.87	95.50	96.20	95.92	99.81	100.00
1007	Norway	NOR	2020	Ranked	95.80	95.76	96.43	95.20	99.84	98.88
1011	Norway	NOR	2016	Ranked	94.71	94.81	94.83	94.50	99.82	99.24
340	Denmark	DNK	2020	Ranked	94.62	95.51	93.25	95.10	99.62	97.75
422	Finland	FIN	2018	Ranked	94.44	94.94	93.72	94.65	99.80	99.24

341	Denmark	DNK	2019	Ranked	94.14	94.42	93.07	94.93	99.60	98.50
420	Finland	FIN	2020	Ranked	94.03	94.58	93.14	94.35	99.79	99.25
343	Denmark	DNK	2017	Ranked	93.91	94.88	93.52	93.32	99.58	97.36
1254	Sweden	SWE	2015	Ranked	93.89	93.92	92.86	94.89	99.65	98.03
1012	Norway	NOR	2015	Ranked	93.85	93.91	93.93	93.71	99.72	97.35
421	Finland	FIN	2019	Ranked	93.83	94.98	92.41	94.09	99.79	99.25
1252	Sweden	SWE	2017	Ranked	93.69	93.99	93.62	93.47	99.71	98.80
1260	Switzerland	CHE	2019	Ranked	93.68	95.09	95.54	90.41	99.42	98.47
423	Finland	FIN	2017	Ranked	93.67	93.80	93.46	93.74	99.79	97.73
1255	Sweden	SWE	2014	Ranked	93.57	94.07	91.92	94.72	99.60	97.27
342	Denmark	DNK	2018	Ranked	93.56	94.17	93.62	92.88	99.59	96.99
1013	Norway	NOR	2014	Ranked	93.32	93.93	92.84	93.21	99.71	97.53
344	Denmark	DNK	2016	Ranked	93.31	94.31	91.86	93.76	99.61	96.98
1251	Sweden	SWE	2018	Ranked	93.22	92.66	94.00	92.99	99.73	98.81
1253	Sweden	SWE	2016	Ranked	93.21	93.13	92.57	93.93	99.71	98.04
424	Finland	FIN	2016	Ranked	93.20	93.61	92.85	93.15	99.76	96.22
1261	Switzerland	CHE	2018	Ranked	93.20	95.18	95.30	89.12	99.41	98.09
425	Finland	FIN	2015	Ranked	93.15	92.69	92.82	93.94	99.71	96.97
1259	Switzerland	CHE	2020	Ranked	93.14	95.03	93.60	90.78	99.43	95.84
1250	Sweden	SWE	2019	Ranked	92.98	93.05	93.11	92.78	99.75	98.43
1257	Sweden	SWE	2012	Ranked	92.97	93.84	91.89	93.17	99.66	98.76
1256	Sweden	SWE	2013	Ranked	92.94	94.27	91.46	93.08	99.60	96.51
345	Denmark	DNK	2015	Ranked	92.86	94.23	90.99	93.35	99.53	96.60
1249	Sweden	SWE	2020	Ranked	92.81	92.38	92.37	93.67	99.76	97.68
346	Denmark	DNK	2014	Ranked	92.64	94.81	89.78	93.35	99.57	98.86
1014	Norway	NOR	2013	Ranked	92.62	92.80	92.24	92.81	99.76	97.71
1258	Sweden	SWE	2011	Ranked	92.57	93.44	91.87	92.39	99.64	99.51
1262	Switzerland	CHE	2017	Ranked	92.51	94.16	95.36	88.00	99.38	98.84
573	Iceland	ISL	2020	Ranked	92.47	94.43	94.65	88.32	99.96	99.65
574	Iceland	ISL	2019	Ranked	92.40	94.32	94.38	88.49	99.95	99.64
575	Iceland	ISL	2018	Ranked	92.31	94.66	93.70	88.57	99.93	99.63
576	Iceland	ISL	2017	Ranked	92.25	94.73	93.69	88.32	99.85	99.63
426	Finland	FIN	2014	Ranked	92.20	93.67	91.00	91.94	99.76	97.72
1263	Switzerland	CHE	2016	Ranked	92.13	94.73	93.09	88.57	99.30	98.84
243	Canada	CAN	2017	Ranked	92.11	93.65	91.76	90.92	98.98	97.55
242	Canada	CAN	2018	Ranked	92.10	93.43	92.08	90.80	99.00	98.18
959	Netherlands	NLD	2018	Ranked	92.07	93.77	91.23	91.21	99.46	95.50

577	Iceland	ISL	2016	Ranked	91.92	94.58	92.87	88.32	99.83	99.63
1264	Switzerland	CHE	2015	Ranked	91.87	93.41	92.94	89.26	99.31	98.83
969	New Zealand	NZL	2018	Ranked	91.76	90.70	94.33	90.27	98.20	95.78
349	Denmark	DNK	2011	Ranked	91.72	95.19	87.74	92.22	99.57	98.84
967	New Zealand	NZL	2020	Ranked	91.70	90.56	93.95	90.59	98.30	93.17
347	Denmark	DNK	2013	Ranked	91.65	93.88	88.83	92.23	99.61	96.97
962	Netherlands	NLD	2015	Ranked	91.64	93.22	90.76	90.93	99.46	95.86
428	Finland	FIN	2012	Ranked	91.60	92.80	89.78	92.21	99.74	96.94
578	Iceland	ISL	2015	Ranked	91.58	94.20	92.53	88.01	99.82	99.62
1015	Norway	NOR	2012	Ranked	91.53	93.51	91.11	89.98	99.62	97.89
1016	Norway	NOR	2011	Ranked	91.51	93.39	90.53	90.63	99.54	98.07
244	Canada	CAN	2016	Ranked	91.49	93.57	90.72	90.17	98.95	96.92
958	Netherlands	NLD	2019	Ranked	91.46	93.33	89.95	91.11	99.45	95.88
427	Finland	FIN	2013	Ranked	91.46	91.92	90.47	91.98	99.69	94.32
968	New Zealand	NZL	2019	Ranked	91.41	90.21	94.11	89.91	98.25	94.48
67	Australia	AUS	2013	Ranked	91.37	92.37	90.69	91.03	98.55	98.03
66	Australia	AUS	2014	Ranked	91.34	92.48	91.27	90.27	98.69	97.91
240	Canada	CAN	2020	Ranked	91.30	91.97	92.75	89.18	99.06	97.45
465	Germany	DEU	2018	Ranked	91.26	93.90	91.28	88.60	99.92	96.51
68	Australia	AUS	2012	Ranked	91.19	92.11	90.21	91.25	98.49	98.13
963	Netherlands	NLD	2014	Ranked	91.16	92.89	90.50	90.10	99.43	96.61
970	New Zealand	NZL	2017	Ranked	91.15	90.99	93.67	88.79	98.15	94.63
429	Finland	FIN	2011	Ranked	91.09	92.96	89.17	91.13	99.59	96.68
70	Austria	AUT	2020	Ranked	91.08	94.28	91.19	87.78	99.50	95.73
348	Denmark	DNK	2012	Ranked	91.07	93.54	87.74	91.94	99.51	98.47
464	Germany	DEU	2019	Ranked	91.06	93.91	90.02	89.26	99.93	95.39
960	Netherlands	NLD	2017	Ranked	91.01	93.34	89.93	89.75	99.45	97.38
69	Australia	AUS	2011	Ranked	90.99	91.83	90.16	90.97	98.45	98.25
61	Australia	AUS	2019	Ranked	90.97	92.03	90.87	90.01	98.83	95.63
247	Canada	CAN	2013	Ranked	90.93	94.34	89.16	89.30	98.91	96.27
60	Australia	AUS	2020	Ranked	90.90	91.65	90.68	90.38	98.85	95.34
63	Australia	AUS	2017	Ranked	90.90	92.32	91.35	89.04	98.78	96.21
971	New Zealand	NZL	2016	Ranked	90.87	90.66	93.35	88.61	98.10	94.61
1265	Switzerland	CHE	2014	Ranked	90.87	93.29	90.88	88.45	99.38	98.60
245	Canada	CAN	2015	Ranked	90.85	92.96	90.38	89.20	98.94	96.28

241	Canada	CAN	2019	Ranked	90.83	91.29	91.39	89.79	99.03	96.31
248	Canada	CAN	2012	Ranked	90.83	94.28	88.54	89.66	98.92	96.62
972	New Zealand	NZL	2015	Ranked	90.83	90.82	93.15	88.51	98.16	94.59
961	Netherlands	NLD	2016	Ranked	90.82	93.00	90.44	89.01	99.46	95.87
964	Netherlands	NLD	2013	Ranked	90.78	92.16	89.79	90.41	99.40	96.23
467	Germany	DEU	2016	Ranked	90.74	93.29	89.70	89.24	99.92	98.38
71	Austria	AUT	2019	Ranked	90.70	94.60	91.17	86.32	99.49	96.47
62	Australia	AUS	2018	Ranked	90.69	91.71	90.88	89.47	98.81	95.92
466	Germany	DEU	2017	Ranked	90.68	93.31	90.64	88.10	99.91	95.75
957	Netherlands	NLD	2020	Ranked	90.62	92.10	89.07	90.68	99.44	95.88
246	Canada	CAN	2014	Ranked	90.60	92.93	89.82	89.04	98.94	96.28
626	Ireland	IRL	2017	Ranked	90.60	93.64	90.75	87.43	99.54	95.55
625	Ireland	IRL	2018	Ranked	90.59	94.14	89.82	87.81	99.58	97.07
64	Australia	AUS	2016	Ranked	90.57	92.10	90.92	88.68	98.75	96.50
579	Iceland	ISL	2014	Ranked	90.57	93.52	90.64	87.54	99.80	99.61
65	Australia	AUS	2015	Ranked	90.53	92.26	91.02	88.32	98.70	96.79
468	Germany	DEU	2015	Ranked	90.51	93.43	89.21	88.88	99.90	96.87
249	Canada	CAN	2011	Ranked	90.41	94.00	87.81	89.42	98.86	96.60
965	Netherlands	NLD	2012	Ranked	90.34	92.50	89.45	89.07	99.42	96.37
1266	Switzerland	CHE	2013	Ranked	90.30	93.00	90.11	87.78	99.24	98.37
973	New Zealand	NZL	2014	Ranked	90.29	90.76	91.66	88.45	97.92	94.58
463	Germany	DEU	2020	Ranked	90.21	92.71	88.71	89.20	99.94	94.65
470	Germany	DEU	2013	Ranked	90.19	93.75	88.78	88.03	99.85	97.23
580	Iceland	ISL	2013	Ranked	90.19	92.98	90.18	87.42	99.78	99.60

In [81]:

```
top_country[['Country','YPI year','YPI_score']].head(10).reset_index(drop = True)
```

Out[81]:

	Country	YPI year	YPI_score
0	Norway	2018	96.56
1	Norway	2019	95.97
2	Norway	2017	95.87
3	Norway	2020	95.80
4	Norway	2016	94.71
5	Denmark	2020	94.62
6	Finland	2018	94.44
7	Denmark	2019	94.14
8	Finland	2020	94.03

9	Denmark	2017	93.91

Displaying the average, minimum and maximum score and categorizing them accordingly

In [82]:

```
df.columns = df.columns.str.replace(' ','_')
```

In [83]:

```
print('Highest YPI score :',df['YPI_score'].max())
print('Lowest YPI score :',df['YPI_score'].min())
print('Average YPI score :',df['YPI_score'].mean())
```

```
Highest YPI score : 96.56
Lowest YPI score : 18.76
Average YPI score : 62.39255735492578
```

Since the average SPI score is 62.39 and the highest is 96.56 we can consider 88 as the qualifying score

In [84]:

```
df.isnull().sum()
```

Out[84]:

```
Country                                       0
YPI_country_code                              0
YPI_year                                      0
Status                                        0
YPI_score                                     0
Basic_Human_Needs                             0
Foundations_of_Wellbeing                      0
Opportunity                                   0
Nutrition_and_Basic_Medical_Care              0
Water_and_Sanitation                          0
Shelter                                       0
Personal_Safety                               0
Access_to_Basic_Knowledge                     0
Access_to_Information_and_Communications      0
Health_and_Wellness                           0
Environmental_Quality                         0
Personal_Rights                               0
Personal_Freedom_and_Choice                   0
Inclusiveness                                 0
Access_to_Advanced_Education                  0
dtype: int64
```

In [85]:

```
fig = px.scatter(df.query("YPI_score>=88"),
                 x = 'Basic_Human_Needs',
                 y = 'YPI_score',
                 size = 'YPI_score',
                 hover_name = 'Country',
                 color = 'Country',
                 title = 'Countries with better Basic Human Needs',
                 log_x = True, size_max = 40)

fig.show()
```

Singapore , Norway and Austria are the top 3 countries with better basic human needs

In [86]:

```
fig = px.scatter(df.query("YPI_score>=89"),
           x = 'Opportunity',
           y = 'YPI_score',
           size = 'YPI_score',
           hover_name = 'Country',
          color = 'Country',
          title = 'Countries with better Opportunity',
          size_max = 30)

fig.show()
```

### Norway, Denmark and Sweden are the top 3 countries with better Opportunity

In [87]:

```
fig = px.scatter(df.query("YPI_score>=88"),
           x = 'Nutrition_and_Basic_Medical_Care',
           y = 'YPI_score',
           size = 'YPI_score',
           hover_name = 'Country',
          color = 'Country',
          title = 'Countries with better Nutrition and Basic Medical Care',
          size_max = 30)

fig.show()
```

Iceland , Gerrmany and Norway are the top 3 countries with better Nutrition and Basic Medical Care

In [88]:

```
fig = px.scatter(df.query("YPI_score>=88"),
           x = 'Water_and_Sanitation',
           y = 'YPI_score',
           size = 'YPI_score',
           hover_name = 'Country',
          color = 'Country',
          title = 'Countries with better Water and Sanitation',
          size_max = 30)

fig.show()
```

Norway, Iceland and Finland are the top 3 countries with better Water and Sanitation

In [89]:

```
df.shape
```

Out[89]:

```
(1482, 20)
```

In [90]:

```
values = dict(type = 'choropleth',
             locations = df['Country'],
             locationmode = 'country names',
             colorscale = 'Blues',
             z = df['YPI_score'],
             text = df['Country'],
             colorbar = {'title' : 'Youth Progress Index'})
layout = dict(title = 'Youth Progress Index',
             geo = dict(showframe = True,
                       projection = {'type':'natural earth'}))

figure = go.Figure(data = [values], layout = layout)
iplot(figure)
```

A visualization to analyze the overall Youth Progress Index scores globally using a choropleth map:

In [91]:

```
df.head()
```

Out[91]:

	Country	YPI_country_code	YPI_year	Status	YPI_score	Basic_Human_Needs	Foundations_of_Wellbeing	Op
0	Afghanistan	AFG	2020	Ranked	31.24	42.19	29.17	
1	Afghanistan	AFG	2019	Ranked	29.83	40.41	28.67	
2	Afghanistan	AFG	2018	Ranked	29.99	40.79	29.60	
3	Afghanistan	AFG	2017	Ranked	30.50	44.11	27.68	
4	Afghanistan	AFG	2016	Ranked	29.30	44.04	26.92	

In [92]:

```
poor_ypi_score_2020 = df[(df.YPI_score<= 60) & (df.YPI_year == 2020)].sort_values(by = '
poor_ypi_score_2019 = df[(df.YPI_score<= 60) & (df.YPI_year == 2019)].sort_values(by = '
```

In [93]:

```
fig2020 = px.scatter(poor_ypi_score_2020.query("Basic_Human_Needs<=60"),
           x = 'Basic_Human_Needs',
           y = 'Nutrition_and_Basic_Medical_Care',
        color = 'Country',
        size = 'Basic_Human_Needs',
        size_max = 20,
          title = 'Countries with poor Basic Human Needs and Basic Medical Care in the Y

fig2019 = px.scatter(poor_ypi_score_2019.query("Basic_Human_Needs<=60"),
           x = 'Basic_Human_Needs',
           y = 'Nutrition_and_Basic_Medical_Care',
        color = 'Country',
        size = 'Basic_Human_Needs',
        size_max = 20,
          title = 'Countries with poor Basic Human Needs and Basic Medical Care in the Y
```

In [94]:

```
fig2019.show(),fig2020.show()
```

```
Out[94]: (None, None)
```

```
In [95]: df.columns
```

```
Out[95]: Index(['Country', 'YPI_country_code', 'YPI_year', 'Status', 'YPI_score',
                'Basic_Human_Needs', 'Foundations_of_Wellbeing', 'Opportunity',
                'Nutrition_and_Basic_Medical_Care', 'Water_and_Sanitation', 'Shelter',
                'Personal_Safety', 'Access_to_Basic_Knowledge',
                'Access_to_Information_and_Communications', 'Health_and_Wellness',
                'Environmental_Quality', 'Personal_Rights',
                'Personal_Freedom_and_Choice', 'Inclusiveness',
                'Access_to_Advanced_Education'],
               dtype='object')
```

```
In [96]: avg_human_needs = df[['Nutrition_and_Basic_Medical_Care', 'Water_and_Sanitation', 'Shelt
                'Personal_Safety', 'Access_to_Basic_Knowledge',
                'Access_to_Information_and_Communications', 'Health_and_Wellness',
                'Environmental_Quality', 'Personal_Rights',
                'Personal_Freedom_and_Choice', 'Inclusiveness',
                'Access_to_Advanced_Education']].mean()
         avg_human_needs = avg_human_needs.sort_values(ascending = False)
```

Average Human needs based on several columns

```
In [97]: avg_human_needs
```

```
Out[97]: Nutrition_and_Basic_Medical_Care            77.509190
         Shelter                                     77.001592
         Access_to_Basic_Knowledge                   74.551734
         Water_and_Sanitation                        68.029669
         Personal_Rights                             62.525877
         Health_and_Wellness                         62.151073
         Access_to_Information_and_Communications    61.847928
         Inclusiveness                               58.435412
         Personal_Safety                             57.941788
         Personal_Freedom_and_Choice                 50.785304
         Access_to_Advanced_Education                49.004892
         Environmental_Quality                       48.925439
         dtype: float64
```

```
In [98]: fig = px.bar(avg_human_needs,
                      y = avg_human_needs.index,
                      x = avg_human_needs.values,
                     color = avg_human_needs.index,
                     width = 1000, height = 500,
                     orientation = 'h',
                     title = 'World Average Score for the Past 10 Years ')
         fig.update_xaxes(tickangle = -90)
         fig.update_traces(texttemplate = '%{x}')
         fig.show()
```

Creating several dataframes based on countries names and assigning them to its regions name

In [99]:

```
europe_countries = df[df['Country'].isin(['Albania','Andorra','Austria',
                        'Belarus','Belgium','Bosnia and Herzegovina',
                        'Bulgaria','Croatia','Cyprus','Czechia','Czech Republic',
                        'Denmark','Estonia','Finland','France',
                        'Germany','Greece','Gibraltar','Guernsey','Hungary',
                        'Iceland','Ireland','Italy','Isle of Man','Jersey',
                        'Kosovo','Latvia','Liechtenstein','Lithuania',
                        'Luxembourg','Malta','Moldova','Monaco','Montenegro','Macedonia'
                        'Netherlands','Republic of North Macedonia','Norway','Poland',
                        'Portugal','Romania','Russia','San Marino','Serbia',
                         'Serbia and Montenegro','Slovakia','Slovenia','Spain',' Serbia a
                         'Sweden','Switzerland','Turkey','Ukraine','United Kingdom','Vati
```

In [100…

```
europe_countries = europe_countries.reset_index(drop = True)
```

In [101…

```
europe_countries.shape
```

Out[101]:

```
(390, 20)
```

In [102…

```
asian_countries = df[df['Country'].isin(['Afghanistan','Armenia','Azerbaijan','Bahrain',
'Bhutan','British Indian Ocean Territory','Brunei',
'Cambodia','China','Christmas Island','Cocos Islands',
'Hong Kong','India','Indonesia','Iran','Iraq',
'Israel','Japan','Jordan','Kazakhstan','Kuwait','Kyrgyzstan',
'Laos','Lebanon','Macao','Malaysia','Maldives','Mongolia',
'Myanmar','Nepal','North Korea','Oman','Pakistan','Palestinian Territory','Palestine','R
'Philippines','Qatar','Saudi Arabia','United Arab Emirates','Singapore','South Korea','S
'Syria','Taiwan','Tajikistan','Thailand','Turkey','Timor-Leste','Turkmenistan','United A
'Uzbekistan','Vietnam','West Bank and Gaza','Yemen'])]
```

In [103…

```
asian_countries = asian_countries.reset_index(drop = True)
asian_countries.shape
```

```
(393, 20)
```

Out[103]:

In [104…

```
N_america_countries = df[df['Country'].isin(['Anguilla','Antigua and Barbuda','Aruba','B
'Belize','Bermuda','British Virgin Islands','Canada','Cayman Islands','Costa Rica','Cuba
'El Salvador','Greenland','Grenada','Guadeloupe','Georgia','Guatemala','Haiti','Honduras
'Netherlands Antilles','Nicaragua','Panama','Puerto Rico','Saint Barthélemy','Saint Kitt
'Saint Martin','Saint Pierre and Miquelon','Saint Vincent and the Grenadines','Trinidad 
'Turks and Caicos Islands','U.S. Virgin Islands','United States'])]
N_america_countries = N_america_countries.reset_index(drop = True)
N_america_countries.shape
```

Out[104]:

```
(134, 20)
```

In [105…

```
S_america_countries = df[df['Country'].isin(['Argentina','Bolivia','Brazil','Chile','Col
                                            'Ecuador','Guyana','Paraguay','Peru','Surina
S_america_countries = S_america_countries.reset_index(drop = True)
```

In [106…

```
Aus_countries = df[df['Country'].isin(['Australia','Fiji','Kiribati','Marshall Islands',
                   'Micronesia','Nauru','New Zealand','Palau','Papua New Guinea',
                   'Samoa','Solomon Islands','Tonga','Tuvalu','Vanuatu'])]
```

In [107…

```
Aus_countries = Aus_countries.reset_index(drop = True)
```

In [108…

```
africa_countries = df[df['Country'].isin(['Algeria','Angola','Benin','Botswana','Burkina
                                          'Central African Republic','Chad','Comoros','C
                                          "Republic of the Cote d'Ivoire",'Djibouti','Do
                                          'Eritrea','Eswatini','Ethiopia','Gabon','Gambi
                                          'Guinea-Bissau','Kenya','Lesotho','Liberia','L
                                          'Mauritania','Mauritius','Morocco','Mozambique
                                          'Rwanda','Sao Tome and Principe','Senegal','Se
                                          'South Africa','South Sudan','Sudan','Tanzania
                                          'Zimbabwe'])]
```

In [109…

```
africa_countries = africa_countries.reset_index(drop = True)
```

In [110…

```
europe_countries.shape
```

Out[110]:

```
(390, 20)
```

In [111…

```
asian_countries.shape
```

Out[111]:

```
(393, 20)
```

In [112…

```
N_america_countries.shape
```

Out[112]:

```
(134, 20)
```

In [113…

```
S_america_countries.shape
```

Out[113]:

```
(100, 20)
```

In [114…

```
Aus_countries.shape
```

Out[114]:

```
(20, 20)
```

In [115…

```
africa_countries.shape
```

Out[115]:

```
(416, 20)
```

In [116…
```
df = df.replace({"Congo, Democratic Republic of": "Democratic Republic of the Congo",
            "Congo, Republic of":"Republic of the Congo",
            "Côte d'Ivoire":"Republic of the Cote d'Ivoire",
            "Gambia, The":"The Gambia",
            "Korea, Republic of":"South Korea"})
```

In [117…
```
europe_countries.columns
```

Out[117]:
```
Index(['Country', 'YPI_country_code', 'YPI_year', 'Status', 'YPI_score',
       'Basic_Human_Needs', 'Foundations_of_Wellbeing', 'Opportunity',
       'Nutrition_and_Basic_Medical_Care', 'Water_and_Sanitation', 'Shelter',
       'Personal_Safety', 'Access_to_Basic_Knowledge',
       'Access_to_Information_and_Communications', 'Health_and_Wellness',
       'Environmental_Quality', 'Personal_Rights',
       'Personal_Freedom_and_Choice', 'Inclusiveness',
       'Access_to_Advanced_Education'],
      dtype='object')
```

In [118…
```
eu_avg = europe_countries[['Nutrition_and_Basic_Medical_Care', 'Water_and_Sanitation', '
       'Personal_Safety', 'Access_to_Basic_Knowledge',
       'Access_to_Information_and_Communications', 'Health_and_Wellness',
       'Environmental_Quality', 'Personal_Rights',
       'Personal_Freedom_and_Choice', 'Inclusiveness',
       'Access_to_Advanced_Education']].mean()
asia_avg = asian_countries[['Nutrition_and_Basic_Medical_Care', 'Water_and_Sanitation',
       'Personal_Safety', 'Access_to_Basic_Knowledge',
       'Access_to_Information_and_Communications', 'Health_and_Wellness',
       'Environmental_Quality', 'Personal_Rights',
       'Personal_Freedom_and_Choice', 'Inclusiveness',
       'Access_to_Advanced_Education']].mean()
N_am_avg = N_america_countries[['Nutrition_and_Basic_Medical_Care', 'Water_and_Sanitatio
       'Personal_Safety', 'Access_to_Basic_Knowledge',
       'Access_to_Information_and_Communications', 'Health_and_Wellness',
       'Environmental_Quality', 'Personal_Rights',
       'Personal_Freedom_and_Choice', 'Inclusiveness',
       'Access_to_Advanced_Education']].mean()
S_am_avg = S_america_countries[['Nutrition_and_Basic_Medical_Care', 'Water_and_Sanitatio
       'Personal_Safety', 'Access_to_Basic_Knowledge',
       'Access_to_Information_and_Communications', 'Health_and_Wellness',
       'Environmental_Quality', 'Personal_Rights',
       'Personal_Freedom_and_Choice', 'Inclusiveness',
       'Access_to_Advanced_Education']].mean()
aus_avg = Aus_countries[['Nutrition_and_Basic_Medical_Care', 'Water_and_Sanitation', 'Sh
       'Personal_Safety', 'Access_to_Basic_Knowledge',
       'Access_to_Information_and_Communications', 'Health_and_Wellness',
       'Environmental_Quality', 'Personal_Rights',
       'Personal_Freedom_and_Choice', 'Inclusiveness',
       'Access_to_Advanced_Education']].mean()
africa_avg = africa_countries[['Nutrition_and_Basic_Medical_Care', 'Water_and_Sanitation
       'Personal_Safety', 'Access_to_Basic_Knowledge',
       'Access_to_Information_and_Communications', 'Health_and_Wellness',
       'Environmental_Quality', 'Personal_Rights',
       'Personal_Freedom_and_Choice', 'Inclusiveness',
       'Access_to_Advanced_Education']].mean()
```

In [119…
```
eu_avg = eu_avg.sort_values(ascending = True)
asia_avg = asia_avg.sort_values(ascending = True)
N_am_avg = N_am_avg.sort_values(ascending = True)
S_am_avg = S_am_avg.sort_values(ascending = True)
aus_avg = aus_avg.sort_values(ascending = True)
africa_avg = africa_avg.sort_values(ascending = True)
```

In [120…
```
fig = px.bar(eu_avg,
             y = eu_avg.index,
             x = eu_avg.values,
             width = 650, height = 450,
             color = eu_avg,
            title = 'Europe region Average Score for the Past 10 Years')
fig.update_traces(texttemplate = '%{x}')

fig1 = px.bar(asia_avg,
             y = asia_avg.index,
             x = asia_avg.values,
              width = 650, height = 450,
             color = eu_avg,
            title = 'Asia region Average Score for the Past 10 Years')
fig1.update_traces(texttemplate = '%{x}')

fig2 = px.bar(N_am_avg,
             y = N_am_avg.index,
             x = N_am_avg.values,
             width = 650, height = 450,
             color = eu_avg,
            title = 'North American region Average Score for the Past 10 Years')
fig2.update_traces(texttemplate = '%{x}')

fig3 = px.bar(S_am_avg,
             y = S_am_avg.index,
             x = S_am_avg.values,
             width = 650, height = 450,
             color = eu_avg,
            title = 'South American region Average Score for the Past 10 Years')
fig3.update_traces(texttemplate = '%{x}')

fig4 = px.bar(aus_avg,
             y = aus_avg.index,
             x = aus_avg.values,
               width = 650, height = 450,
             color = eu_avg,
            title = 'Oceania region Average Score for the Past 10 Years')
fig4.update_traces(texttemplate = '%{x}')

fig5 = px.bar(africa_avg,
             y = africa_avg.index,
             x = africa_avg.values,
               width = 650, height = 450,
             color = eu_avg,
            title = 'African region Average Score for the Past 10 Years')
fig5.update_traces(texttemplate = '%{x}')

fig5.show(),fig1.show(),fig3.show(),fig2.show(),fig.show(),fig4.show()
```

```
Out[120]: (None, None, None, None, None, None)
```

Average Score based on several factors for the Past 10 years in the African, Asian, North American, South American, Europe and Oceania Region

```
In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```