Countries Rating

This Jupiter notebook constitutes part of a research project aimed at addressing the inquiry, "Which country is the most suitable for an individual?" utilizing data collected by the Organization for Economic Co-operation and Development (OECD). The project's full details are available in the webpage at ___ or in the PowerPoint presentation at _.

This notebook covers the following steps:

1. Data Collection and Exploration: The OECD data will be obtained, transformed into a Pandas dataframe, investigated, cleansed, profiled, and extraneous data will be eliminated.
2. Data Wrangling: The missing values will be imputed using various linear regression models and the k-nearest neighbors method from Sklearn. In order to facilitate inter-country comparisons, new parameters will be generated by combining normalized specific characteristics. Following that, the dataframe will be restructured into a suitable format.
3. Data Visualization and Analysis: The dataset will be visualized and examined using interactive Plotly charts in order to extract insights and create an informative dashboard.

Data Collection

Import the required libraries.

In [1]:
```
import pandas as pd
import numpy as np
import plotly.express as px
from sklearn import linear_model
from sklearn.preprocessing import MinMaxScaler
from sklearn.impute import KNNImputer
```

The dataset in question is available on the OECD website at the URL: https://stats.oecd.org/Index.aspx?DataSetCode=BLI.
To facilitate subsequent work, we shall transform the dataset, which is stored in a .csv file, into a Pandas dataframe.

In [2]:
```
df = pd.read_csv('BLI_11042023094335807.csv')
```

Data Exploration

Firstly, we will print the dataset's shape and the top 5 rows to acquaint ourselves with the data. The dataset comprises 2369 rows and 17 columns.

In [3]:
```
print(df.shape)
df.head(3)
```

```
(2369, 17)
```

Out[3]:

	LOCATION	Country	INDICATOR	Indicator	MEASURE	Measure	INEQUALITY	Inequality	Unit Code	Unit	F
0	AUS	Australia	JE_LMIS	Labour	L	Value	TOT	Total	PC	Percentage	

				market insecurity						
1	AUT	Austria	JE_LMIS	Labour market insecurity	L	Value	TOT	Total	PC	Percentage
2	BEL	Belgium	JE_LMIS	Labour market insecurity	L	Value	TOT	Total	PC	Percentage

The dataset contains a substantial amount of information, while we are only interested in specific data. To achieve our aim, we shall employ the .pivot function to generate a dataframe from the data we require, with the country names as rows and the feature names as columns.

In [4]:

```
Rating = df[df['INEQUALITY']=='TOT'] #to obtain general information, dataset contains se
Rating = Rating.pivot(index='Country', columns='Indicator', values='Value')
pd.options.display.max_columns = None #to see all columns
Rating
```

Out[4]:

Indicator	**Air pollution**	**Dwellings without basic facilities**	**Educational attainment**	**Employees working very long hours**	**Employment rate**	**Feeling safe walking alone at night**	**Homicide rate**	**Household net adjusted disposable income**	**Househ net we**
Country									
Australia	6.7	NaN	84.0	12.5	73.0	67.0	0.9	37433.0	5287
Austria	12.2	0.8	86.0	5.3	72.0	86.0	0.5	37001.0	3096
Belgium	12.8	0.7	80.0	4.3	65.0	56.0	1.1	34884.0	4476
Brazil	11.7	6.7	57.0	5.6	57.0	45.0	19.0	NaN	N
Canada	7.1	0.2	92.0	3.3	70.0	78.0	1.2	34421.0	4782
Chile	23.4	9.4	67.0	7.7	56.0	41.0	2.4	NaN	1357
Colombia	22.6	12.3	59.0	23.7	58.0	50.0	23.1	NaN	N
Costa Rica	17.5	2.3	43.0	22.0	55.0	47.0	10.0	16517.0	N
Czech Republic	17.0	0.5	94.0	4.5	74.0	77.0	0.7	26664.0	N
Denmark	10.0	0.5	82.0	1.1	74.0	85.0	0.5	33774.0	1498
Estonia	5.9	5.7	91.0	2.2	74.0	79.0	1.9	23784.0	1886
Finland	5.5	0.4	91.0	3.6	72.0	88.0	1.2	33471.0	2300
France	11.4	0.5	81.0	7.7	65.0	74.0	0.4	34375.0	2986
Germany	12.0	0.1	86.0	3.9	77.0	76.0	0.4	38971.0	3043
Greece	14.5	0.4	76.0	4.5	56.0	69.0	1.0	20791.0	1483
Hungary	16.7	3.5	86.0	1.5	70.0	74.0	0.9	21026.0	1502
Iceland	6.4	0.0	76.0	11.7	78.0	85.0	0.3	NaN	N
Ireland	7.8	0.2	85.0	4.7	68.0	76.0	0.5	29488.0	3703
Israel	19.7	NaN	88.0	14.1	67.0	80.0	1.5	NaN	N

Italy	15.9	0.6	63.0	3.3	58.0	73.0	0.5	29431.0	2950.
Japan	13.7	6.4	NaN	NaN	77.0	77.0	0.2	28872.0	2947.
Korea	27.3	2.5	89.0	NaN	66.0	82.0	0.8	24590.0	3623
Latvia	12.7	11.2	89.0	1.6	72.0	72.0	3.7	19783.0	792
Lithuania	10.5	11.8	94.0	1.0	72.0	62.0	2.5	26976.0	1820.
Luxembourg	10.0	0.1	74.0	2.8	67.0	87.0	0.2	44773.0	9411
Mexico	20.3	25.9	42.0	27.0	59.0	42.0	26.8	16269.0	N
Netherlands	12.2	0.1	81.0	0.3	78.0	83.0	0.6	34984.0	2485
New Zealand	6.0	NaN	81.0	14.0	77.0	66.0	1.3	39024.0	5141
Norway	6.7	0.0	82.0	1.4	75.0	93.0	0.6	39144.0	2683.
OECD - Total	14.0	3.0	79.0	10.2	66.0	74.0	2.6	30490.0	3239
Poland	22.8	2.3	93.0	4.2	69.0	71.0	0.5	23675.0	2332.
Portugal	8.3	0.9	55.0	5.6	69.0	83.0	0.7	24877.0	2553
Russia	11.8	13.8	95.0	0.1	70.0	64.0	4.8	19546.0	N
Slovak Republic	18.5	1.5	92.0	4.2	68.0	76.0	0.8	21149.0	1714.
Slovenia	17.0	0.2	90.0	5.6	71.0	91.0	0.4	25250.0	2332
South Africa	28.5	35.9	48.0	15.4	39.0	40.0	13.7	9338.0	N
Spain	10.0	0.3	63.0	2.5	62.0	80.0	0.7	27155.0	3665.
Sweden	5.8	0.0	84.0	0.9	75.0	79.0	1.1	33730.0	N
Switzerland	10.1	0.0	89.0	0.4	80.0	86.0	0.3	39697.0	N
Türkiye	27.1	4.9	42.0	25.0	48.0	59.0	1.0	NaN	N
United Kingdom	10.1	0.5	82.0	10.8	75.0	78.0	0.2	33049.0	5244.
United States	7.7	0.1	92.0	10.4	67.0	78.0	6.0	51147.0	6845

We shall examine the data types to check for any issues that may require attention.

In [5]:
```
Rating.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 42 entries, Australia to United States
Data columns (total 24 columns):
 #   Column                                     Non-Null Count  Dtype  
---  ------                                     --------------  -----  
 0   Air pollution                              42 non-null     float64
 1   Dwellings without basic facilities         39 non-null     float64
 2   Educational attainment                     41 non-null     float64
 3   Employees working very long hours          40 non-null     float64
 4   Employment rate                            42 non-null     float64
 5   Feeling safe walking alone at night        42 non-null     float64
 6   Homicide rate                              42 non-null     float64
 7   Household net adjusted disposable income   36 non-null     float64
 8   Household net wealth                       30 non-null     float64
 9   Housing expenditure                        38 non-null     float64
 10  Labour market insecurity                   35 non-null     float64
```

```
 11  Life expectancy                                  42 non-null     float64
 12  Life satisfaction                                42 non-null     float64
 13  Long-term unemployment rate                      40 non-null     float64
 14  Personal earnings                                36 non-null     float64
 15  Quality of support network                       42 non-null     float64
 16  Rooms per person                                 39 non-null     float64
 17  Self-reported health                             40 non-null     float64
 18  Stakeholder engagement for developing regulations  40 non-null     float64
 19  Student skills                                   40 non-null     float64
 20  Time devoted to leisure and personal care        23 non-null     float64
 21  Voter turnout                                    42 non-null     float64
 22  Water quality                                    42 non-null     float64
 23  Years in education                               40 non-null     float64
dtypes: float64(24)
memory usage: 8.2+ KB
```

The dataframe contains missing values. We shall determine the number of missing values for each country and feature.

In [6]:
```
nan_cols = Rating.isna().sum().sort_values(ascending=False)
nan_rows = Rating.isna().sum(axis=1).sort_values(ascending=False)
print(f'Nan in rows: {nan_rows[nan_rows > 0]}', f'\n\nNan in columns: {nan_cols[nan_cols
```

```
Nan in rows: Country
South Africa       9
Brazil             9
Colombia           6
Russia             5
Costa Rica         5
Israel             5
Iceland            4
Türkiye            3
Switzerland        3
Chile              3
Japan              2
Mexico             2
Australia          2
Lithuania          2
Sweden             2
Czech Republic     2
Slovak Republic    1
Slovenia           1
Portugal           1
New Zealand        1
Spain              1
Luxembourg         1
Korea              1
Latvia             1
Denmark            1
dtype: int64

Nan in columns: Indicator
Time devoted to leisure and personal care            19
Household net wealth                                 12
Labour market insecurity                              7
Household net adjusted disposable income              6
Personal earnings                                     6
Housing expenditure                                   4
Dwellings without basic facilities                    3
Rooms per person                                      3
Self-reported health                                  2
Student skills                                        2
Employees working very long hours                     2
Long-term unemployment rate                           2
Stakeholder engagement for developing regulations     2
```

```
Years in education                              2
Educational attainment                          1
dtype: int64
```

The feature "Time devoted to leisure and personal care" has a high number of missing values. It would be incredibly challenging to replace such a high number of missing values without losing the veracity of the values. Furthermore, this feature seems contentious in terms of evaluating the overall work-life balance indicator, which was planned to be calculated using this feature. We have opted to remove this feature from the research. We shall also exclude the row "OECD - Total" from the dataframe, which represents the average values of features among all OECD member countries. This row is unnecessary for our research and may distort calculated indicators.

In [7]:
```
Rating = Rating.drop(columns='Time devoted to leisure and personal care')
Rating = Rating.drop(index='OECD - Total')
```

Next, we shall explore the features' fundamental statistical characteristics.

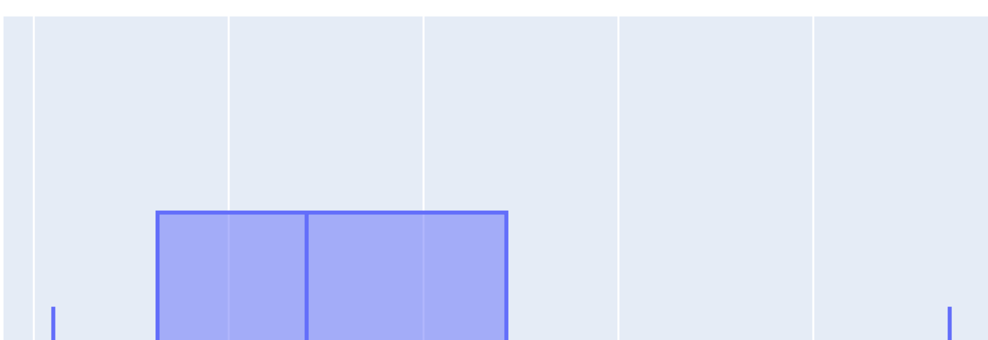
In [8]:
```
Rating.describe()
```

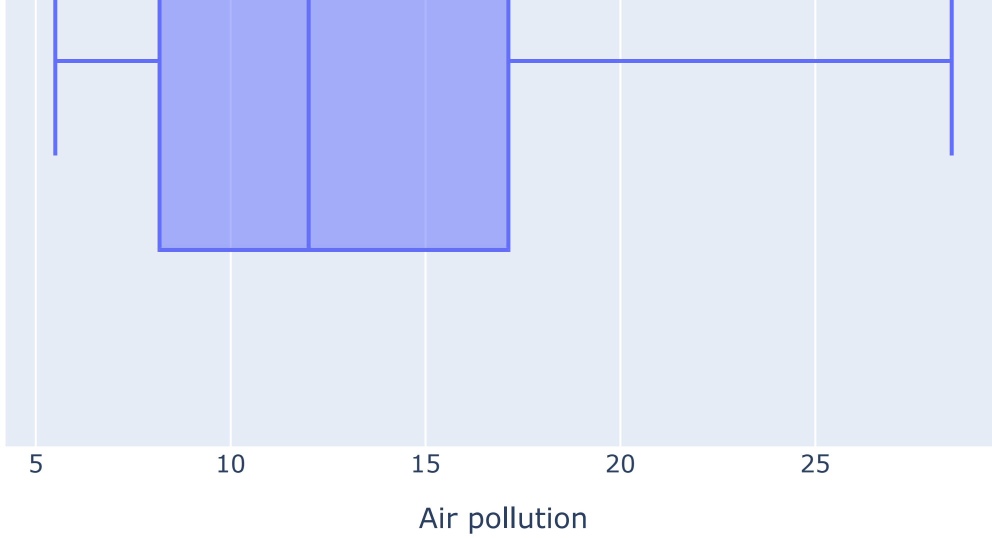
Out[8]:

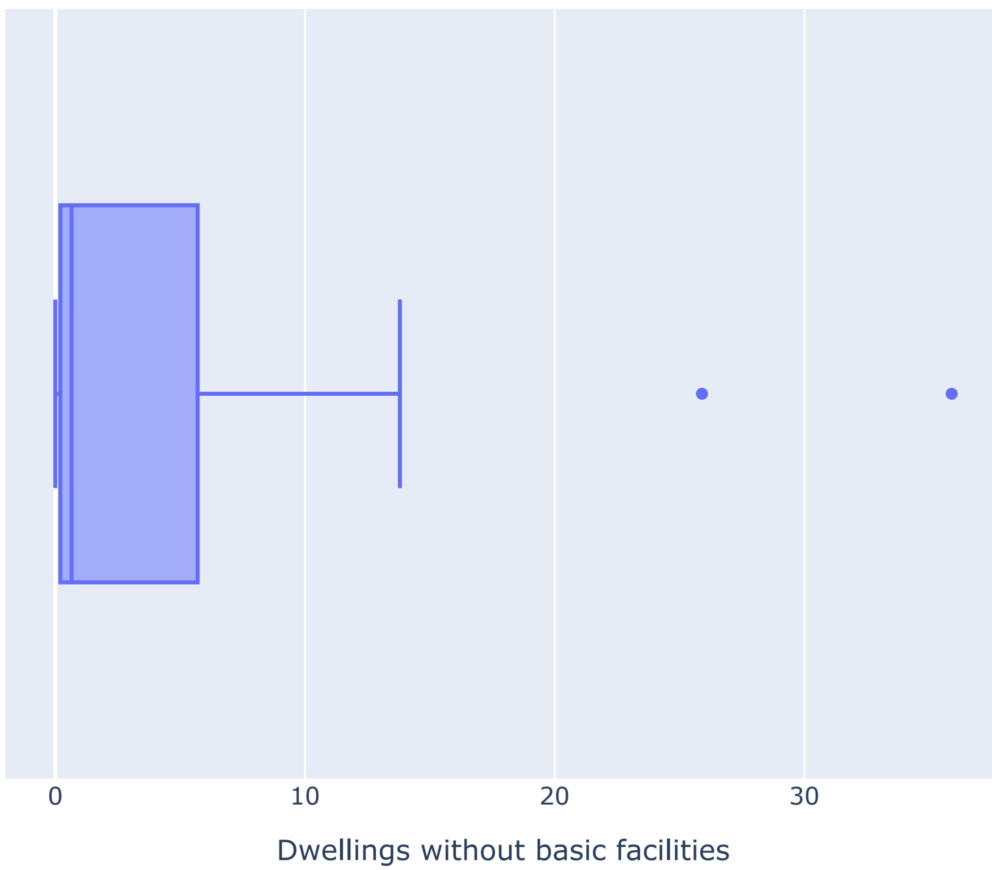
Indicator	Air pollution	Dwellings without basic facilities	Educational attainment	Employees working very long hours	Employment rate	Feeling safe walking alone at night	Homicide rate	Household net adjusted disposable income	Ho ne
count	41.000000	38.000000	40.00000	39.000000	41.000000	41.000000	41.000000	35.000000	2
mean	13.509756	4.294737	78.10000	7.189744	67.682927	72.073171	3.290244	29573.114286	32395
std	6.394169	7.589180	15.68406	7.199481	8.818841	14.299284	6.251792	8880.424174	18552
min	5.500000	0.000000	42.00000	0.100000	39.000000	40.000000	0.200000	9338.000000	7924
25%	8.300000	0.200000	72.25000	2.350000	65.000000	66.000000	0.500000	23729.500000	18862
50%	12.000000	0.650000	83.00000	4.500000	70.000000	76.000000	0.900000	29431.000000	29473
75%	17.000000	5.500000	89.25000	10.600000	74.000000	82.000000	1.900000	34934.000000	37034
max	28.500000	35.900000	95.00000	27.000000	80.000000	93.000000	26.800000	51147.000000	94116

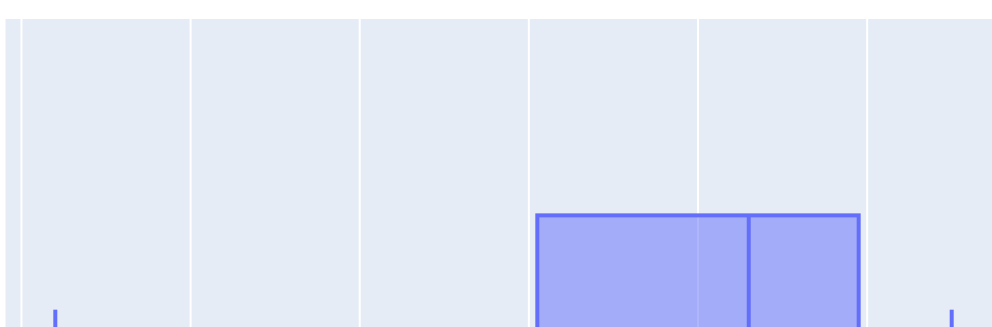
We explore the distribution of features using interactive boxplots. We can hover over the outlier to find out which country the value belongs to.

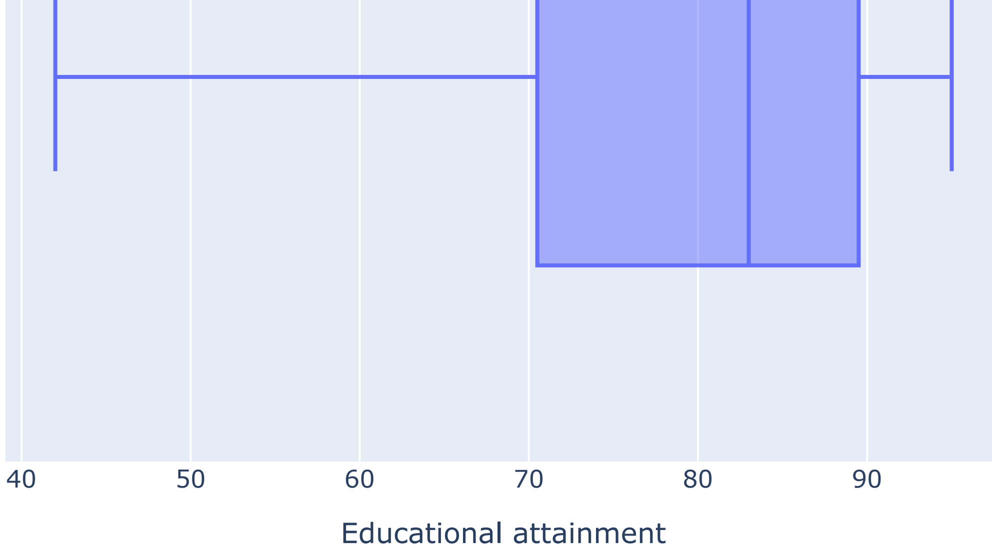
In [9]:
```
for i in Rating.columns:
    fig = px.box(Rating, x=i, hover_data=[Rating.index])
    fig.show()
```

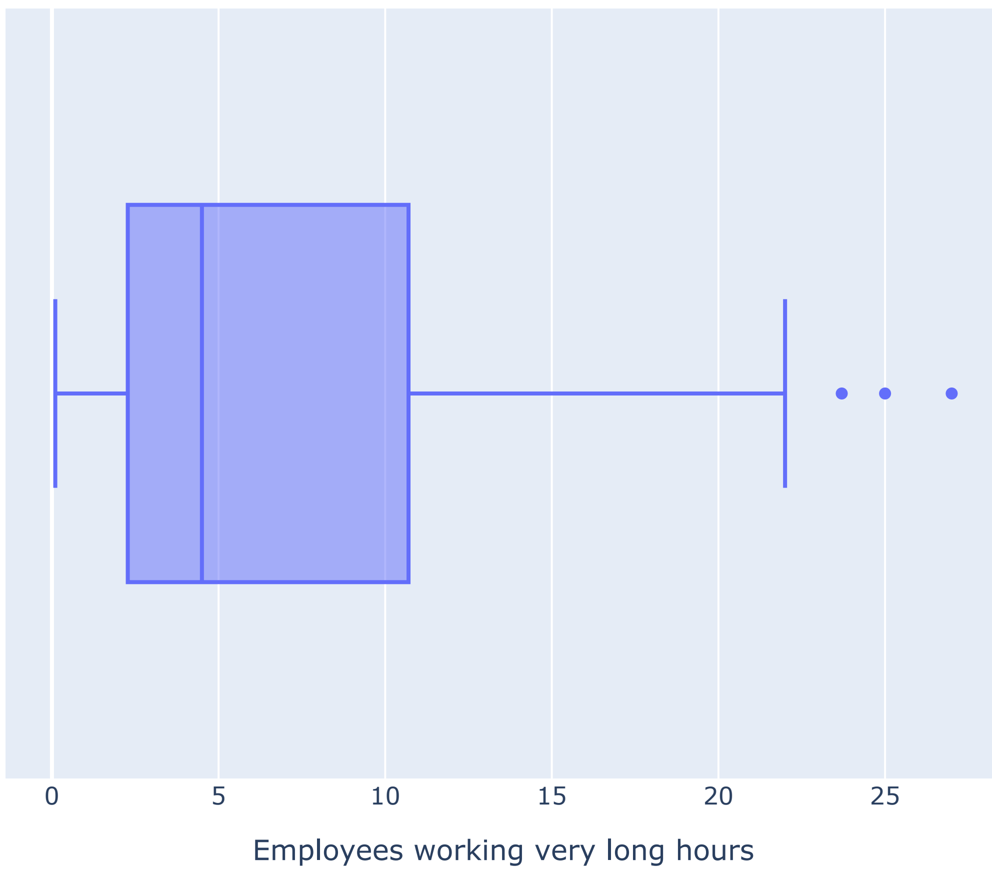


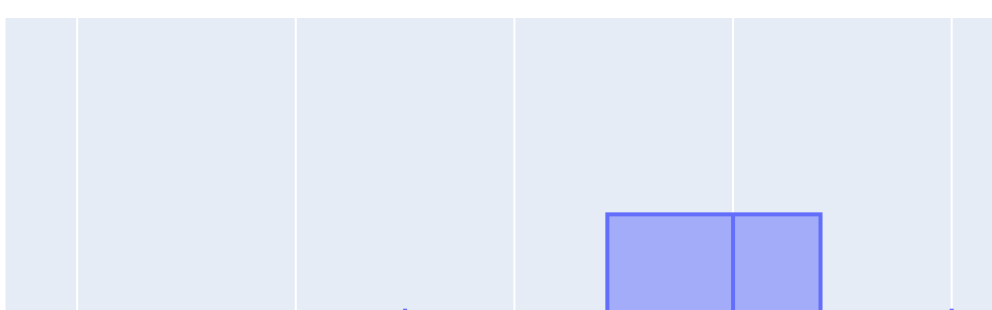

5
10
15
20
25
Air pollution


0
10
20
30
Dwellings without basic facilities

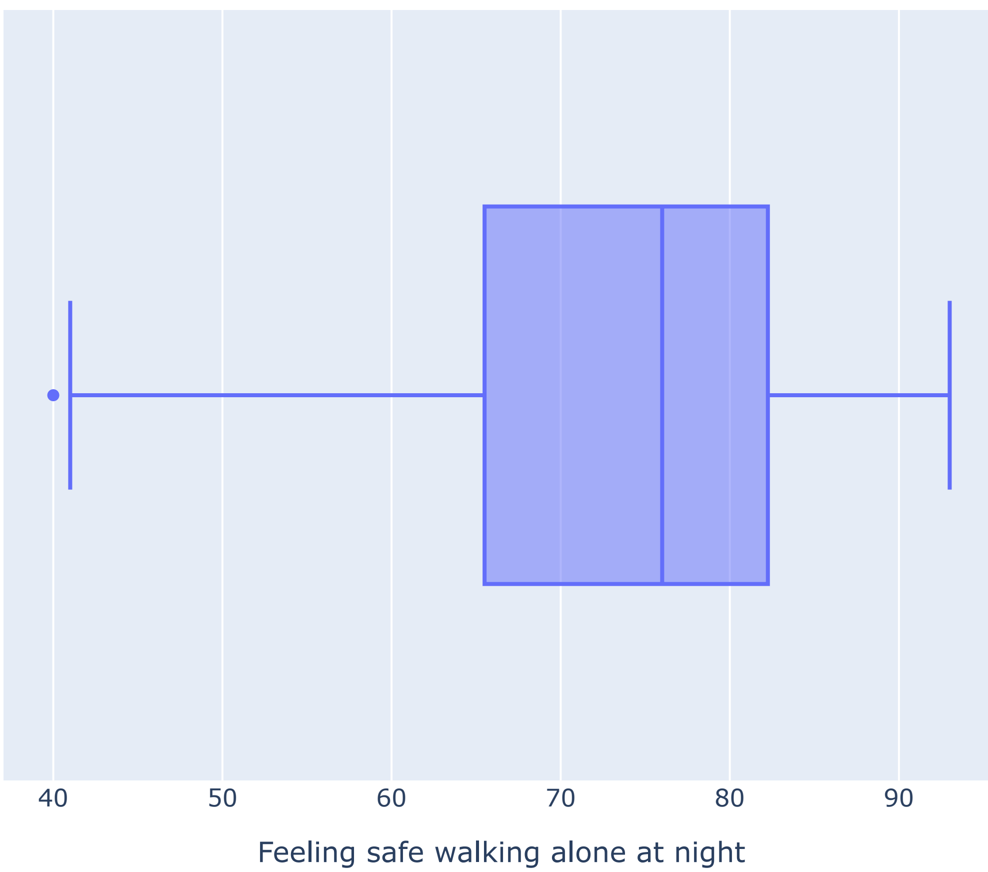


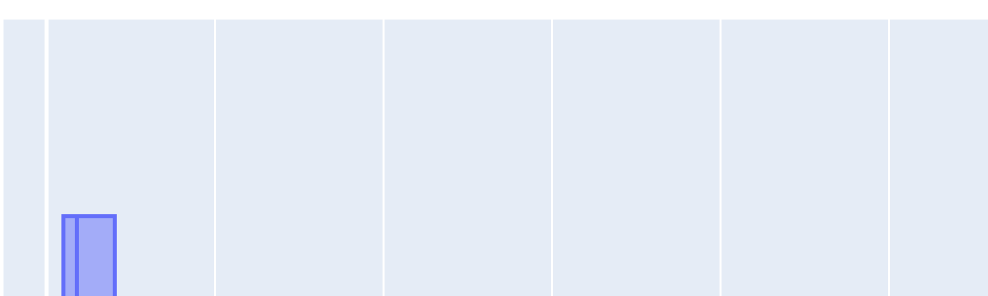

40
50
60
70
80
90
Educational attainment


0
5
10
15
20
25
Employees working very long hours

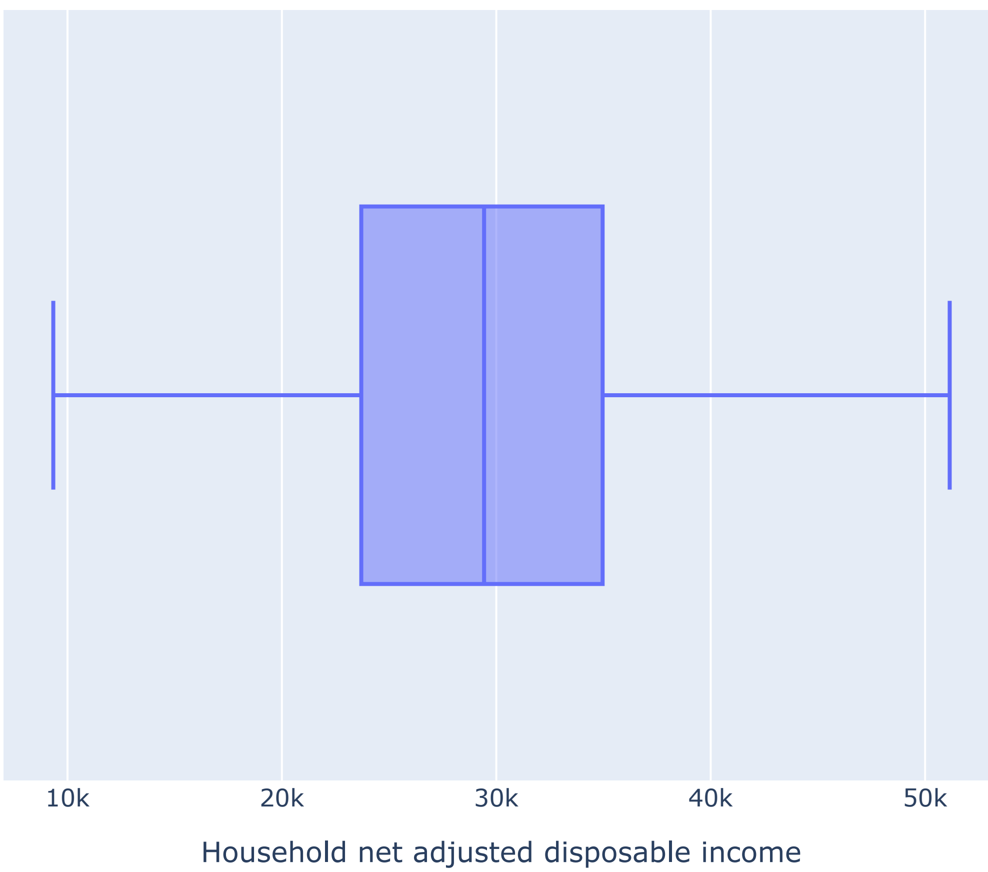


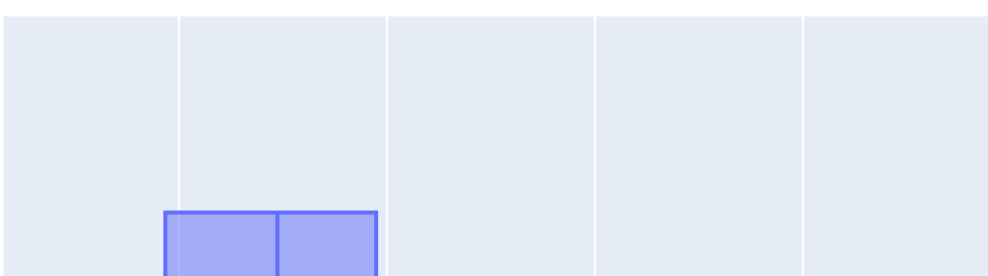

40
50
60
70
80
Employment rate


40
50
60
70
80
90
Feeling safe walking alone at night

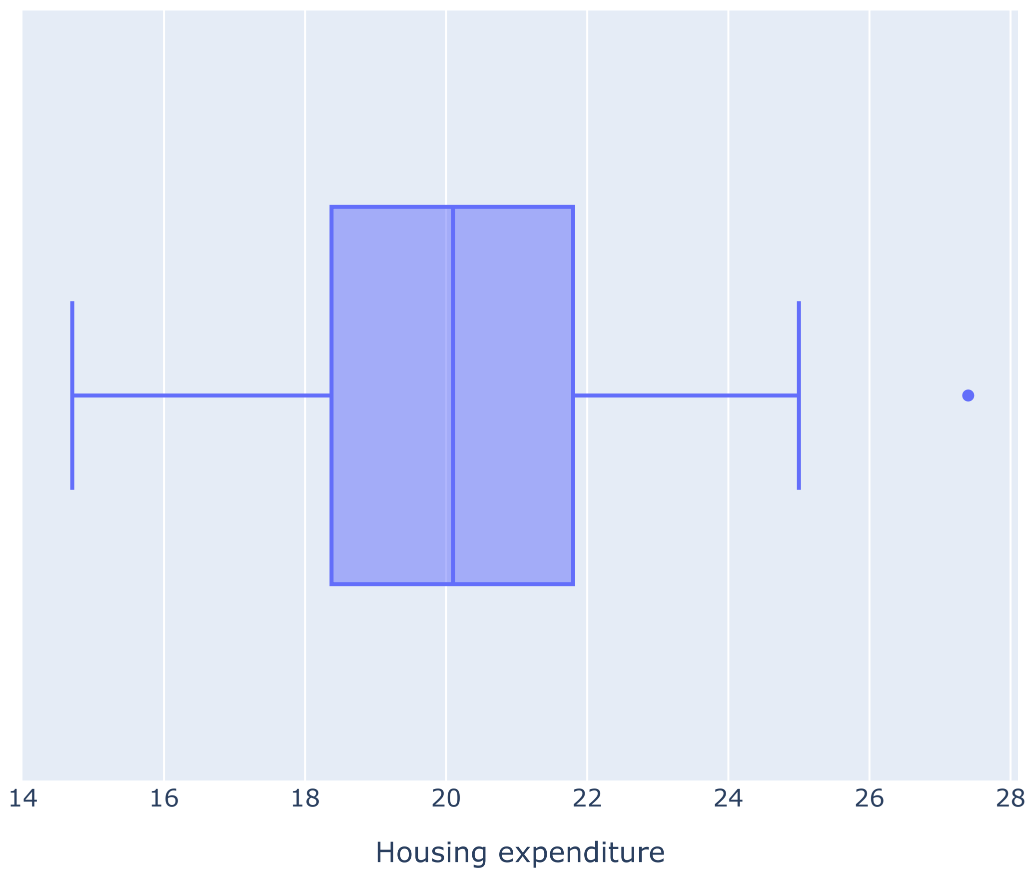


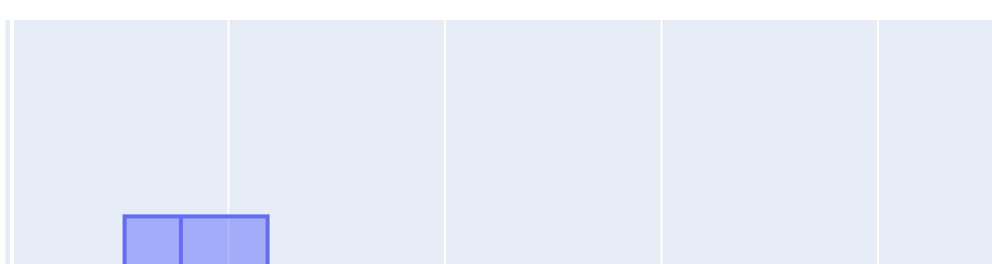

0
5
10
15
20
25
Homicide rate


10k
20k
30k
40k
50k
Household net adjusted disposable income

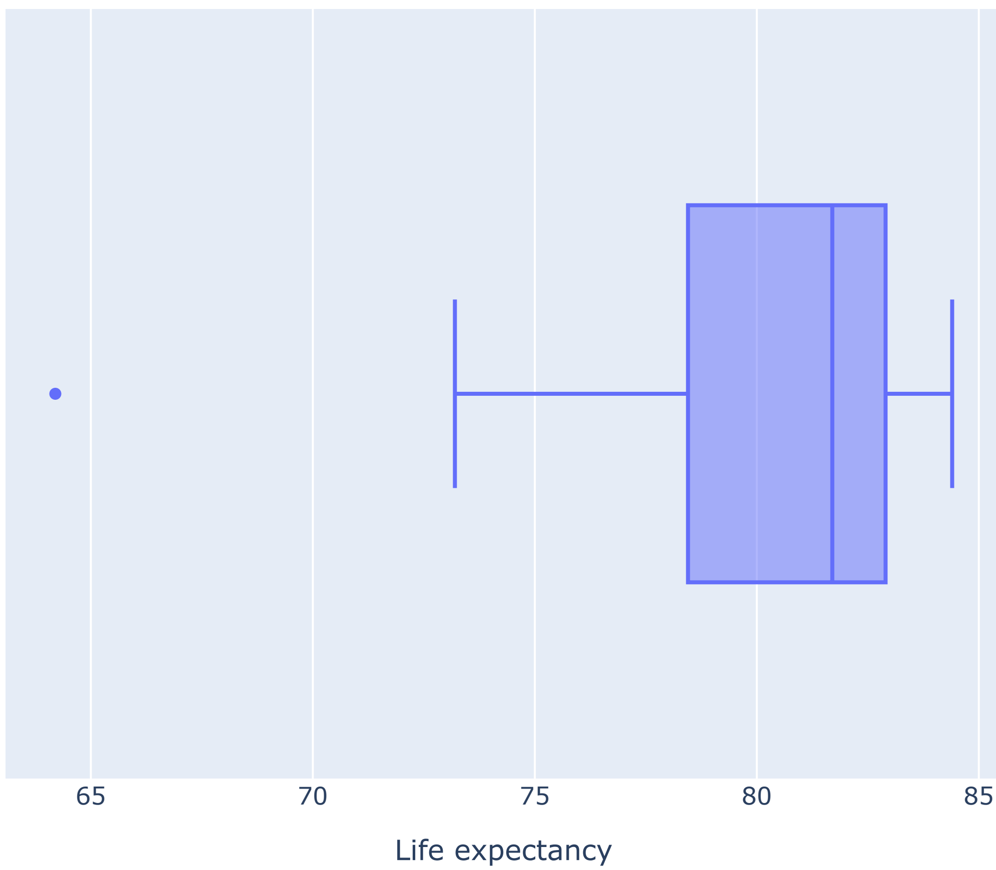


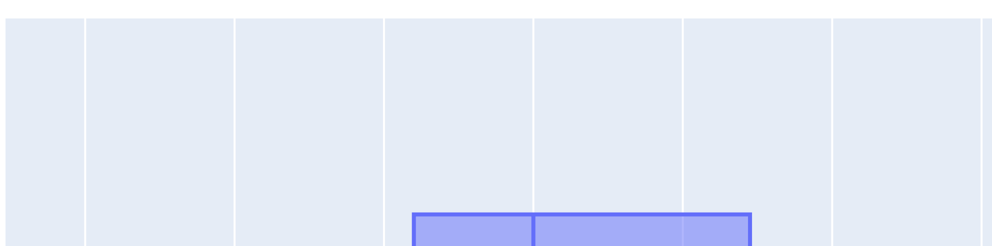

0.2M
0.4M
0.6M
0.8M
Household net wealth

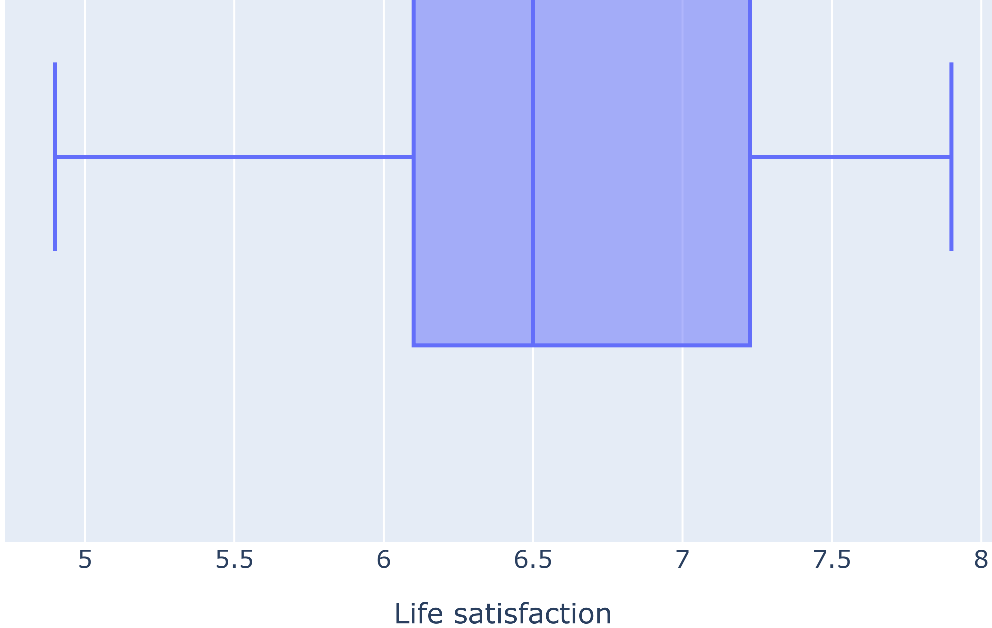

14
16
18
20
22
24
26
28
Housing expenditure



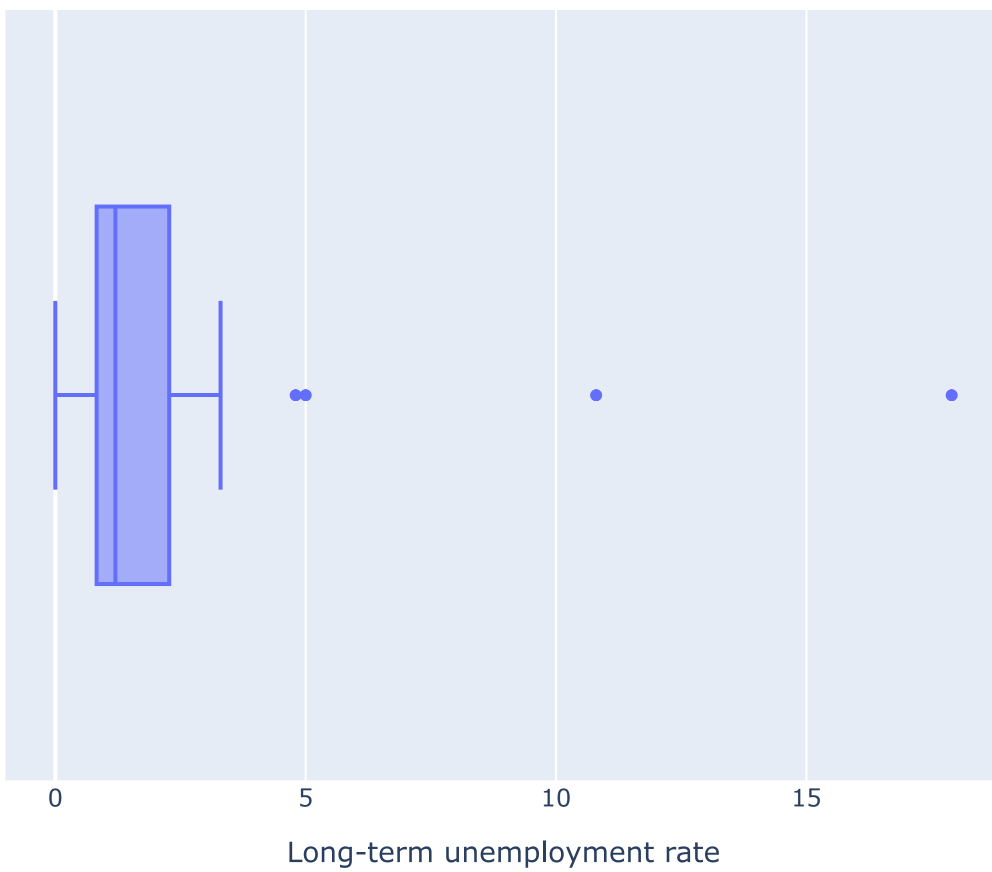

0
5
10
15
20
Labour market insecurity


65
70
75
80
85
Life expectancy

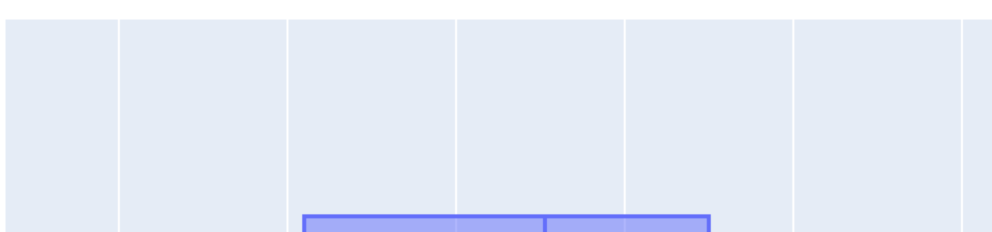




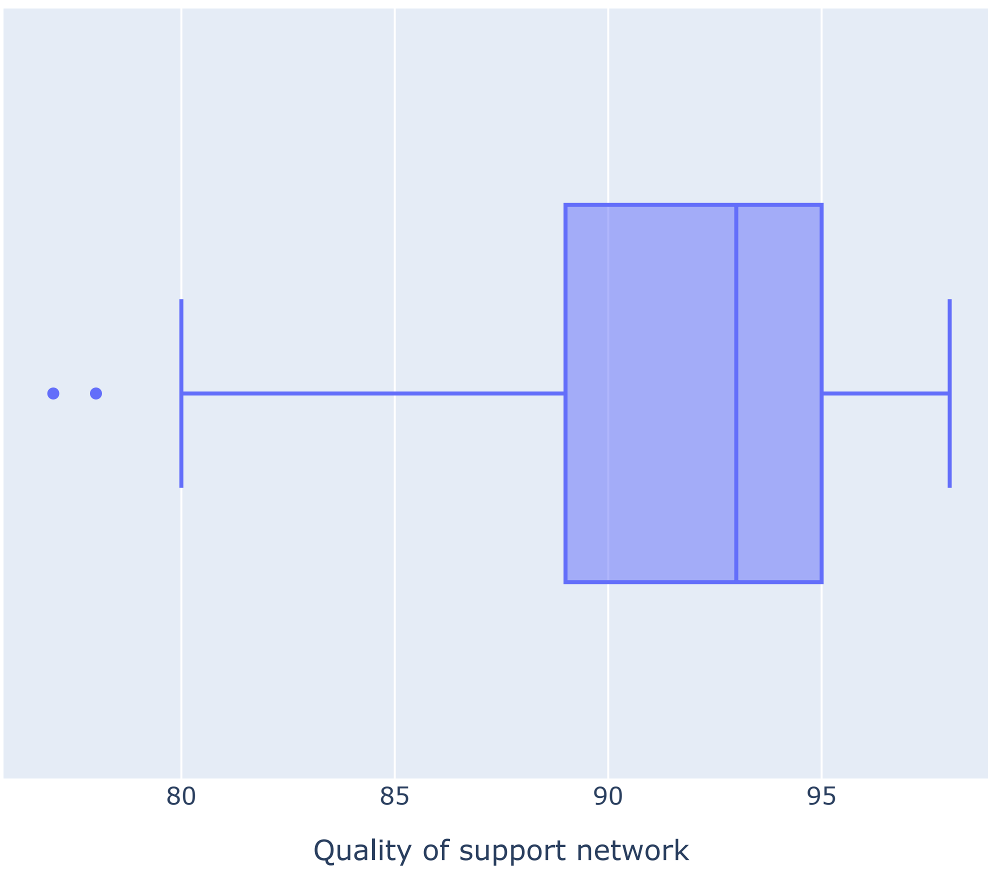
5
5.5
6
6.5
7
7.5
8
Life satisfaction



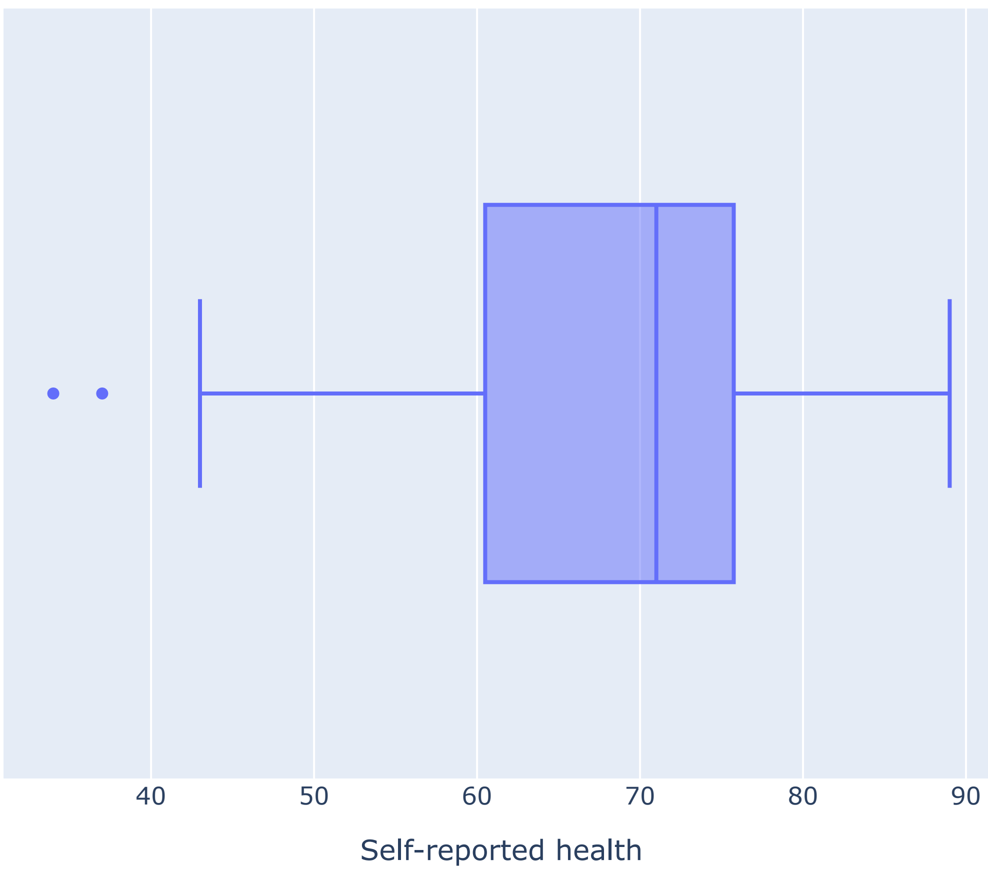
0
5
10
15
Long-term unemployment rate

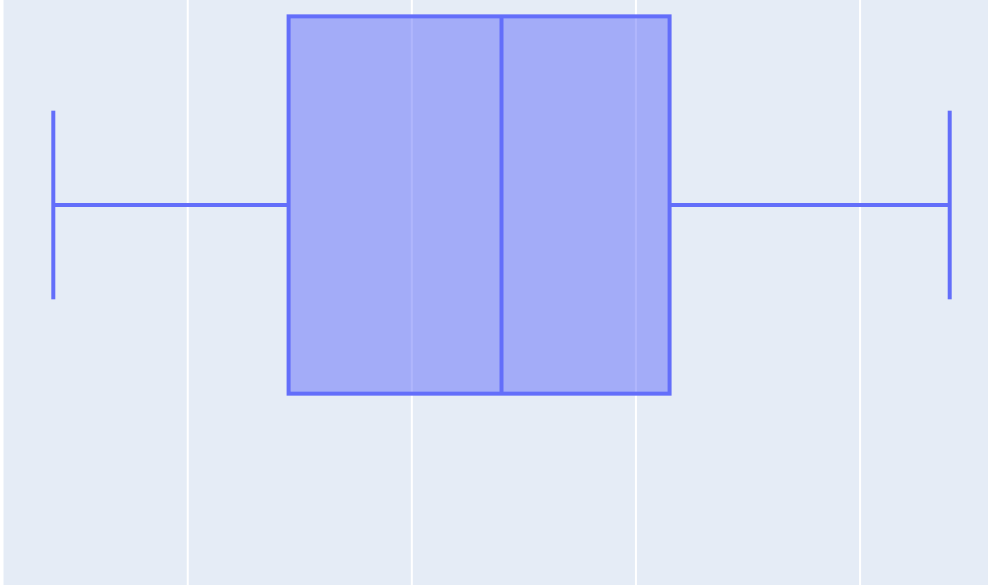



20k
30k
40k
50k
60k
70k
Personal earnings

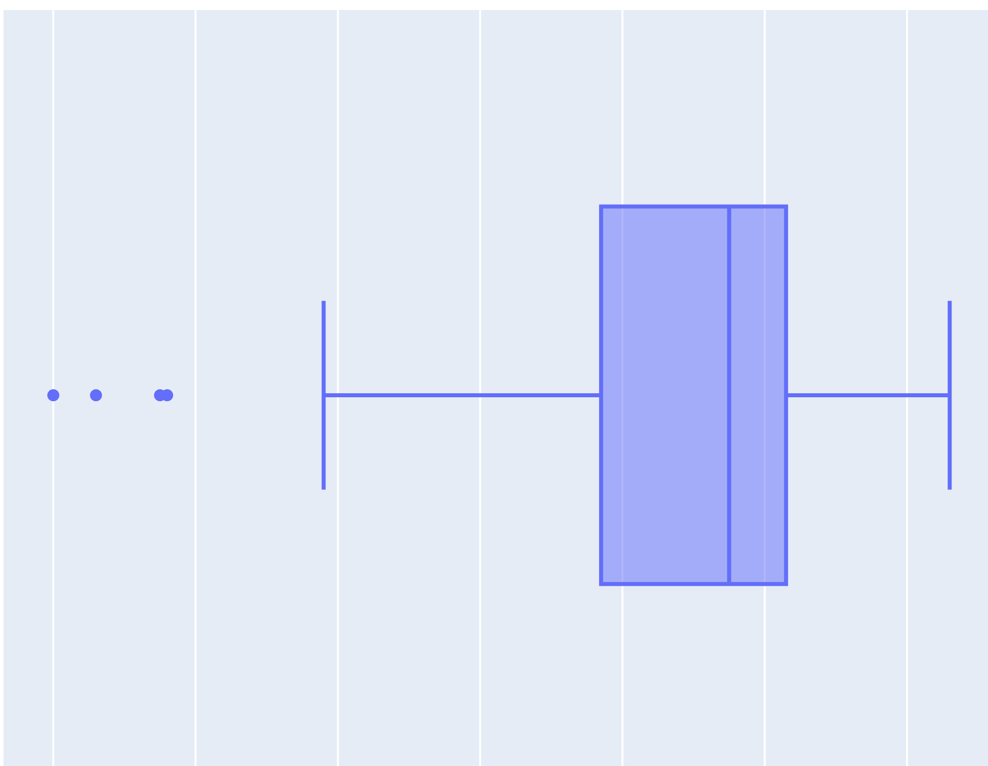

80
85
90
95
Quality of support network


1
1.5
2
2.5
Rooms per person

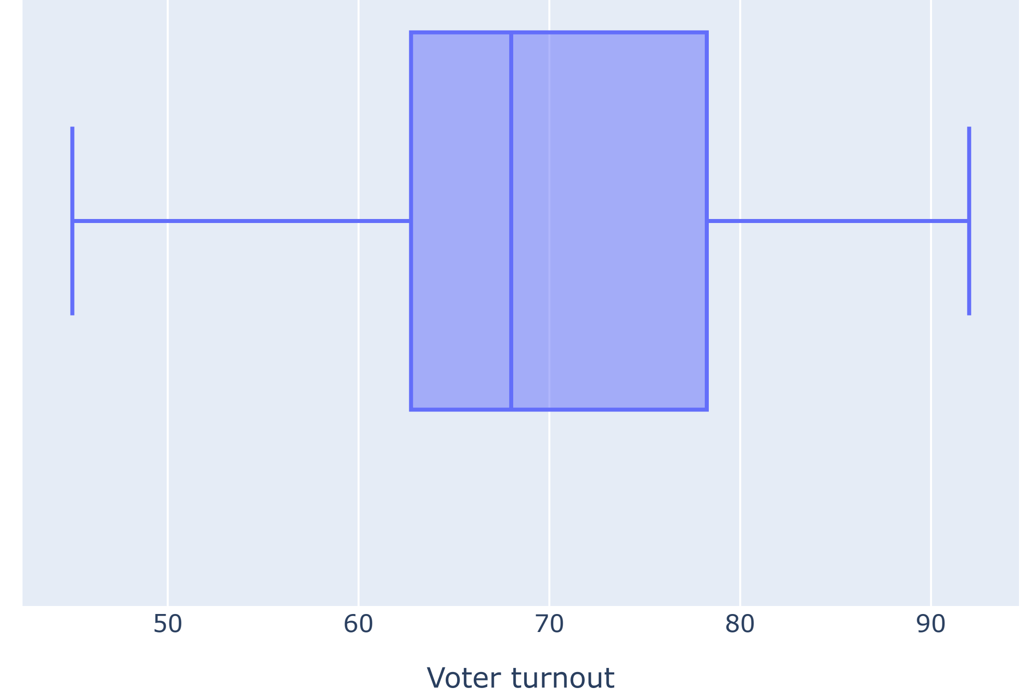

40
50
60
70
80
90
Self-reported health

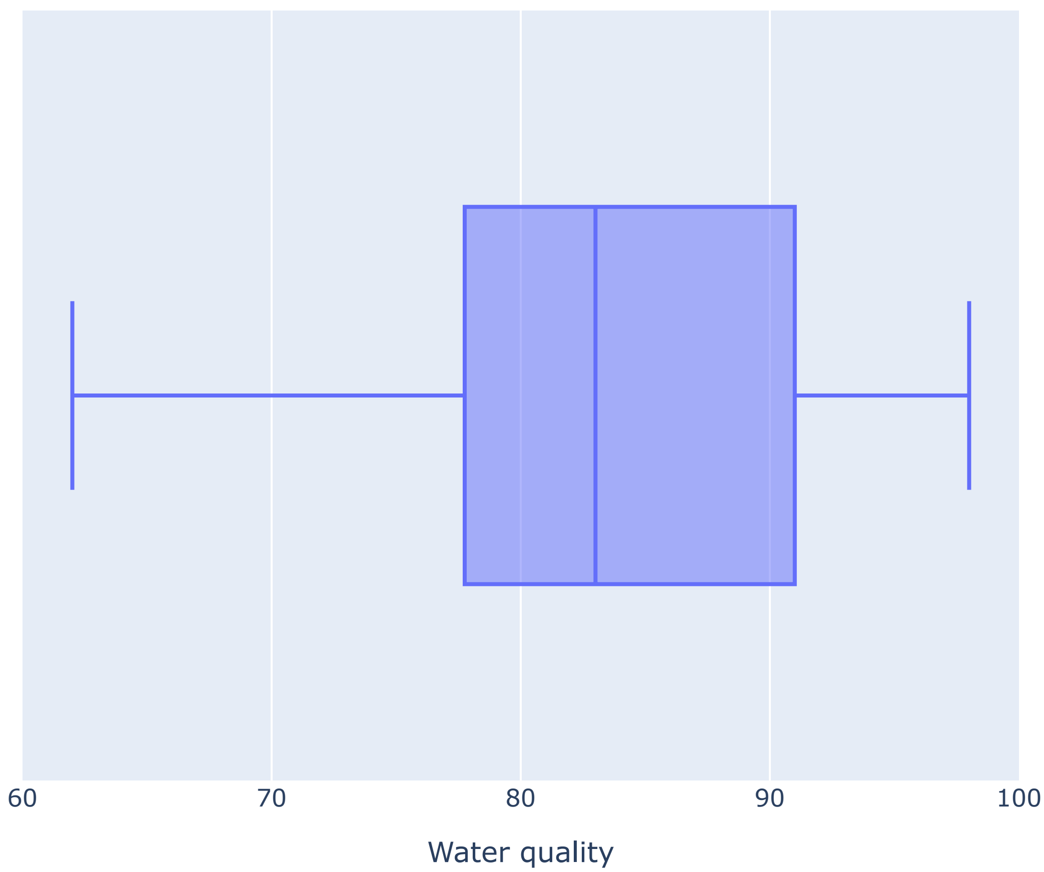


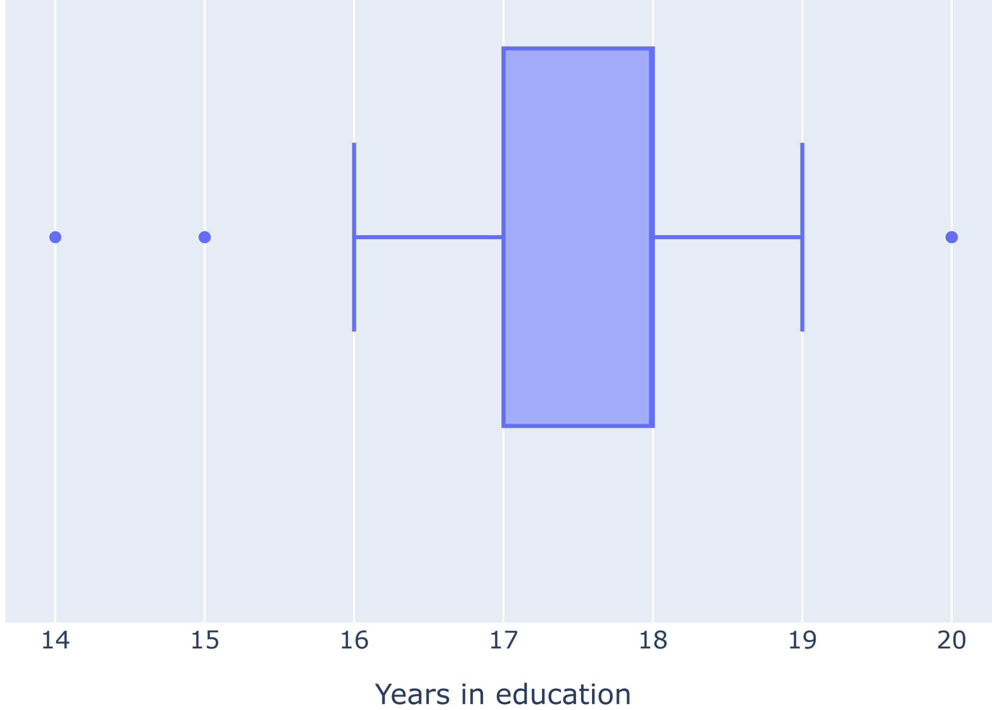
1.5
2
2.5
3
Stakeholder engagement for developing regulations



400
420
440
460
480
500
520
Student skills


50
60
70
80
90
Voter turnout


60
70
80
90
100
Water quality

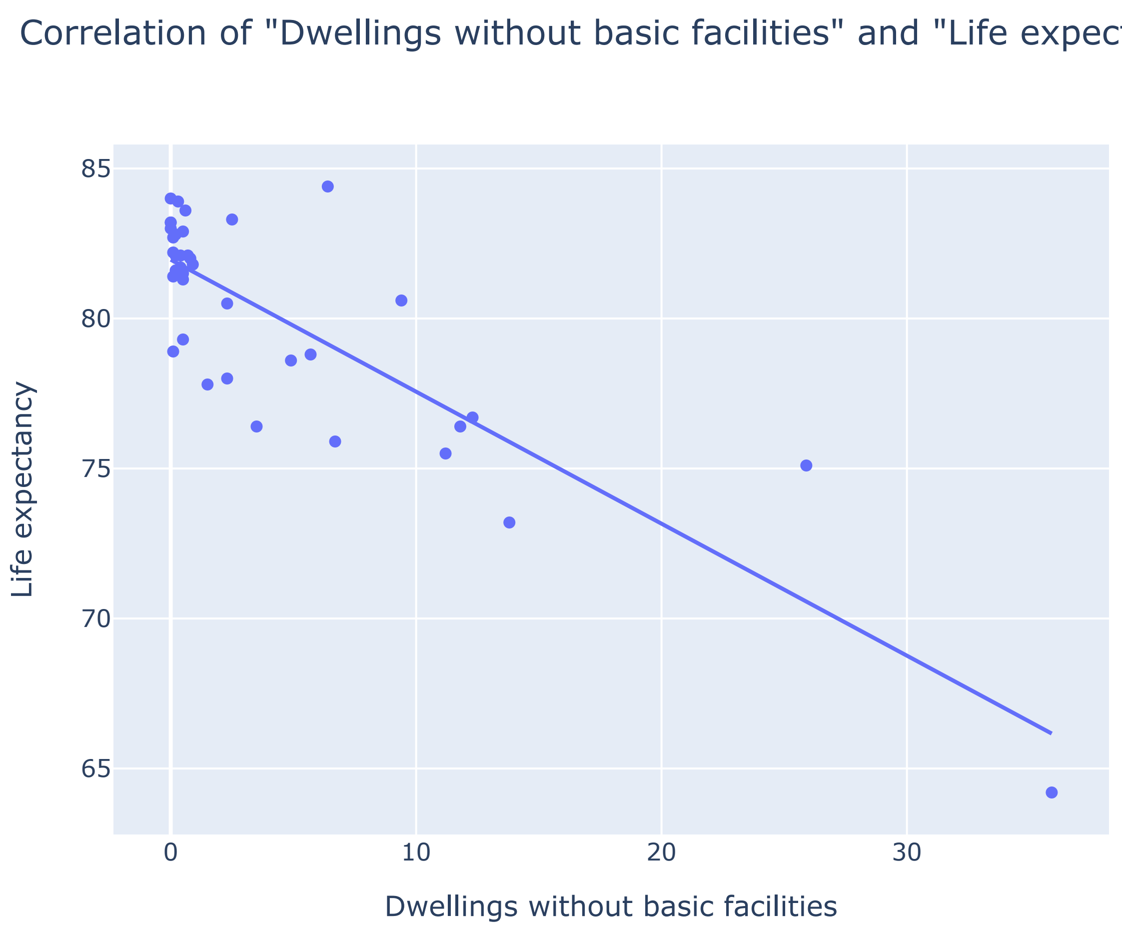


Let us explore the correlation between the features. To visualize the correlation, we shall construct a heatmap graph utilizing Plotly. The graph is interactive, allowing us to zoom in and out and pan across the plot as needed.

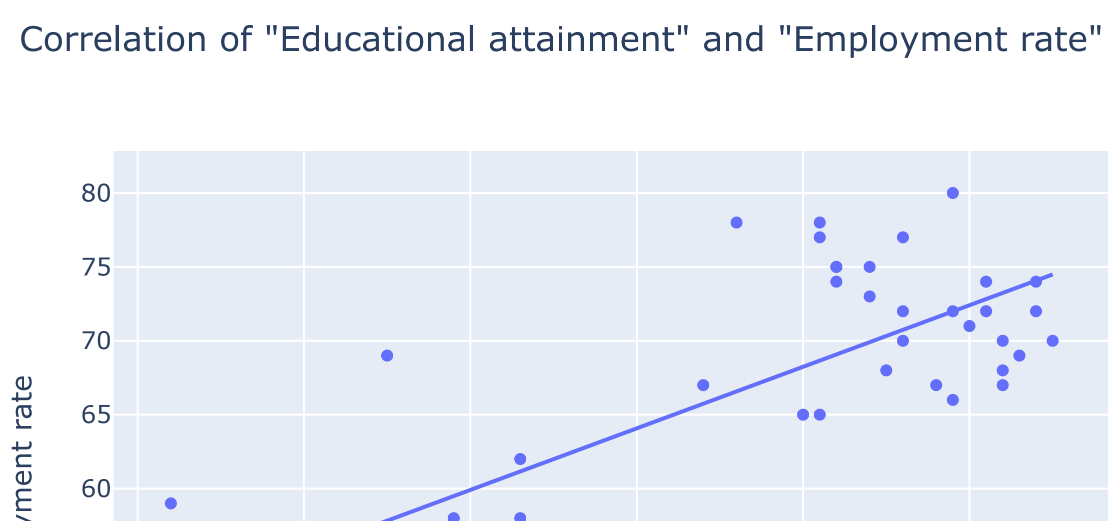
```
In [10]: px.imshow(Rating.corr(), text_auto=True)
```

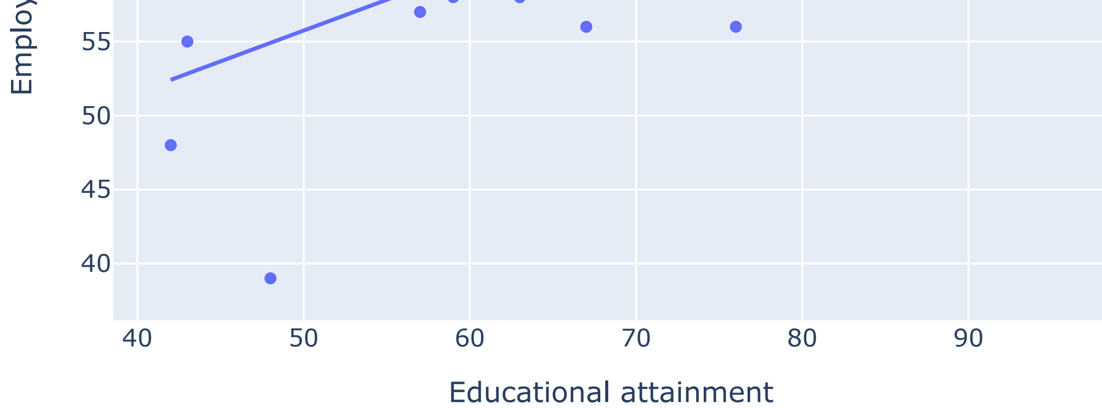
Next, we shall construct a scatter plot with a trend line to investigate the spread of feature values around the trend line. We shall generate a loop that will examine the correlation between the features and plot a graph if it surpasses an absolute value of 0.7.

In [11]:
```
for c in Rating.columns:
    for i in Rating.columns:
        if (abs(Rating.corr().loc[i, c]) > 0.7) & (abs(Rating.corr().loc[i, c]) !=1):
            fig = px.scatter(Rating, y=i, x=c, hover_data=[Rating.index], trendline="ols
            fig.show()
            print(f'Corr_coef of "{c}" and "{i}" = {Rating.corr().loc[i, c]}')
```



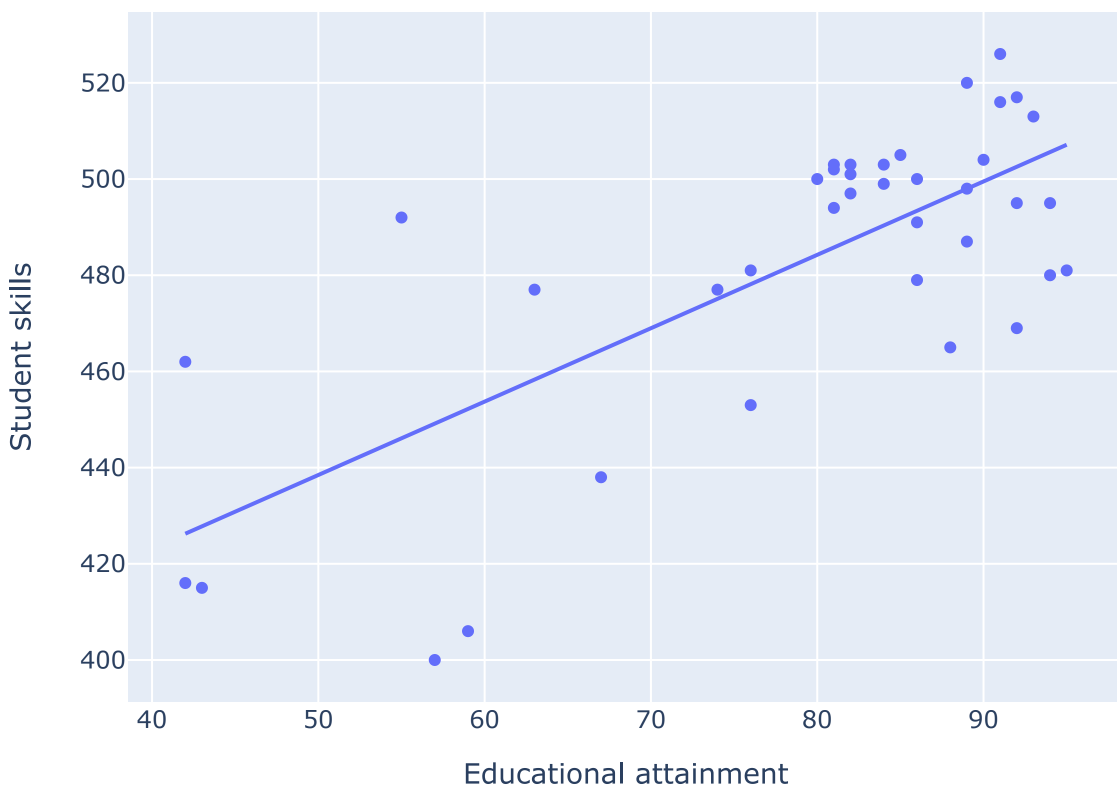

```
Corr_coef of "Dwellings without basic facilities" and "Life expectancy" = -0.84910675670
51078
```





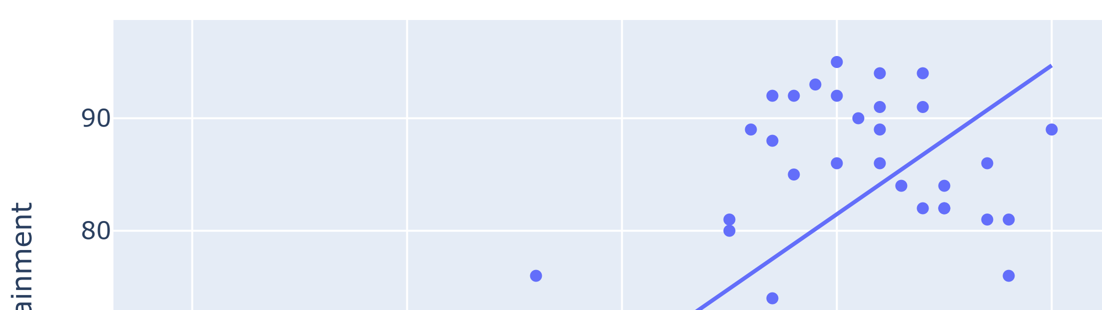

```
Corr_coef of "Educational attainment" and "Employment rate" = 0.742192168999446
```

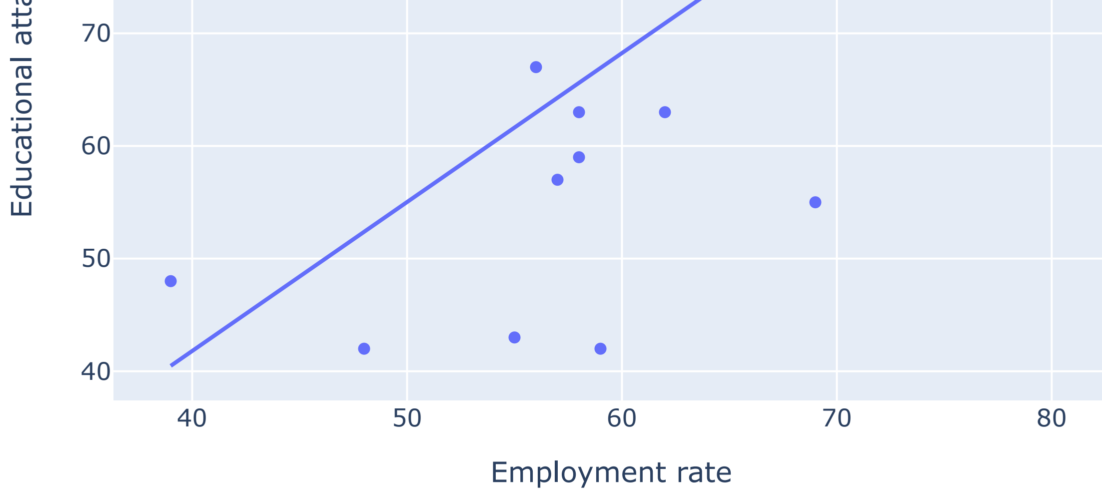
Correlation of "Educational attainment" and "Student skills"




```
Corr_coef of "Educational attainment" and "Student skills" = 0.7280897847632455
```

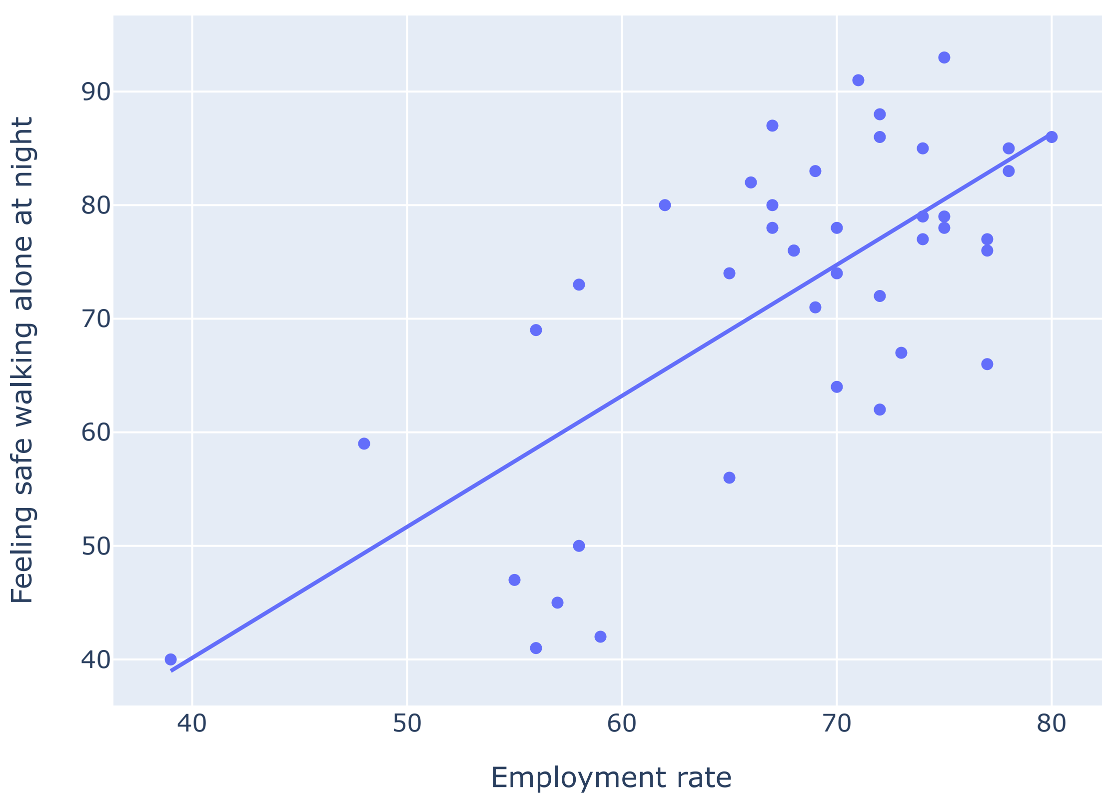
Correlation of "Employment rate" and "Educational attainment"





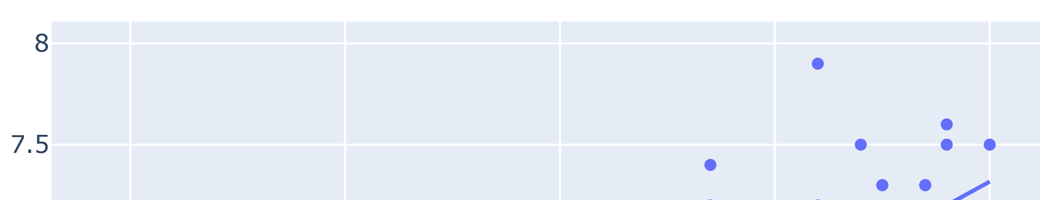

```
Corr_coef of "Employment rate" and "Educational attainment" = 0.742192168999446
```

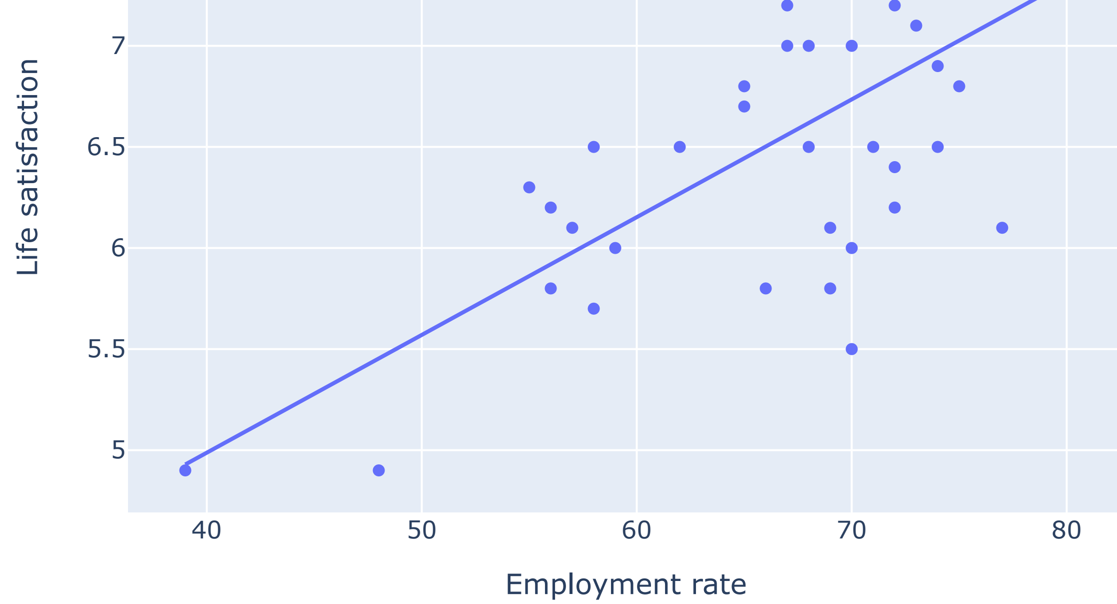
Correlation of "Employment rate" and "Feeling safe walking alone at nigh




```
Corr_coef of "Employment rate" and "Feeling safe walking alone at night" = 0.71151127441
10984
```

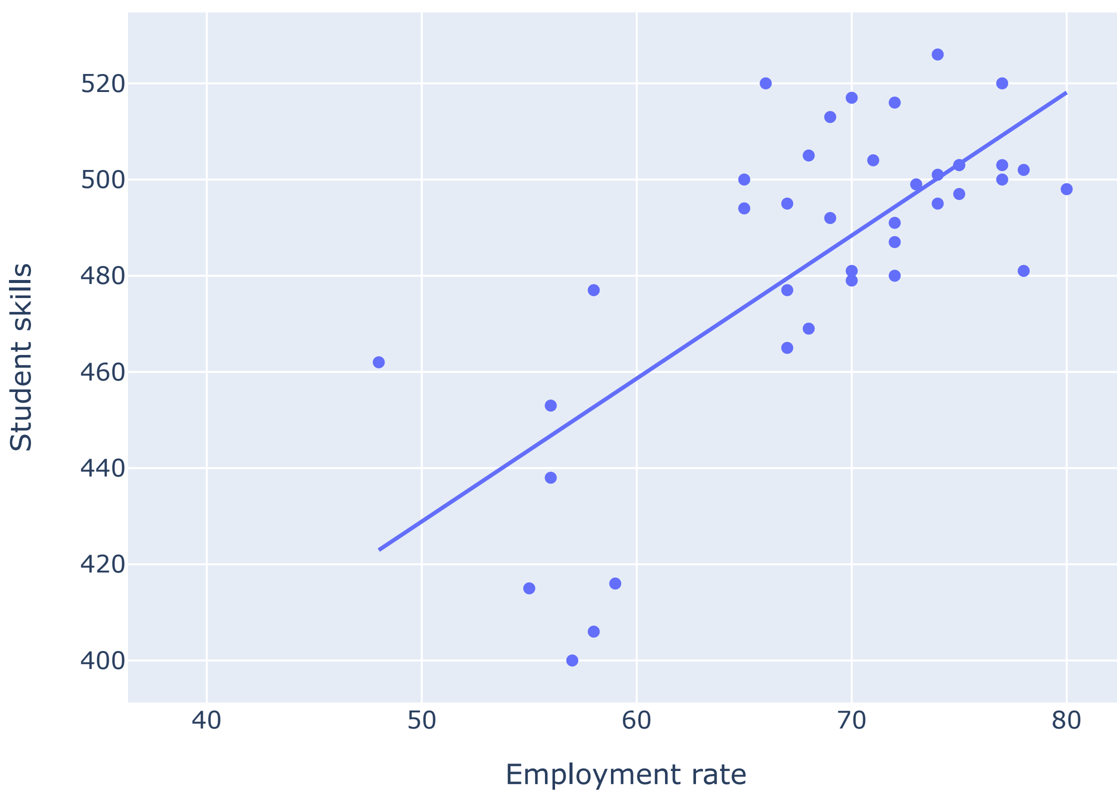
Correlation of "Employment rate" and "Life satisfaction"





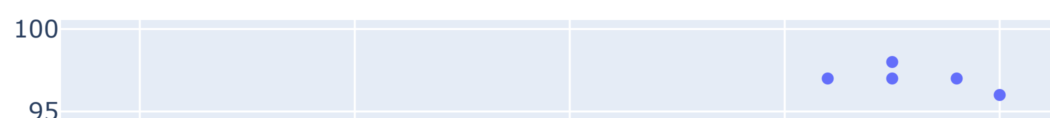

```
Corr_coef of "Employment rate" and "Life satisfaction" = 0.7082618221255973
```

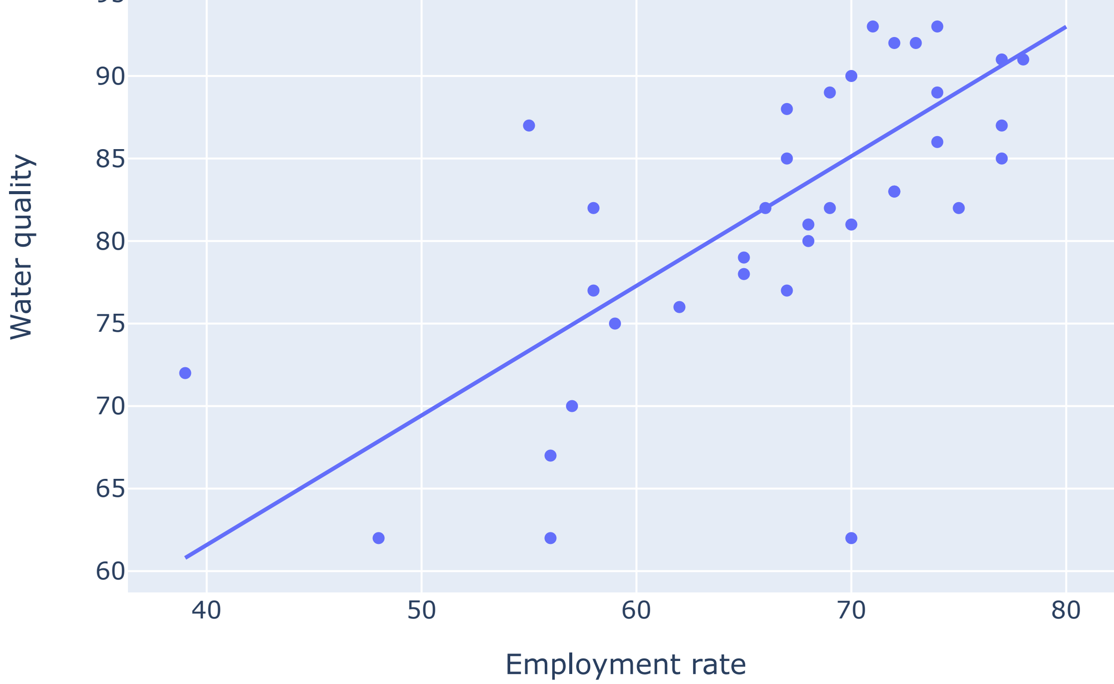
Correlation of "Employment rate" and "Student skills"




```
Corr_coef of "Employment rate" and "Student skills" = 0.717330689097553
```

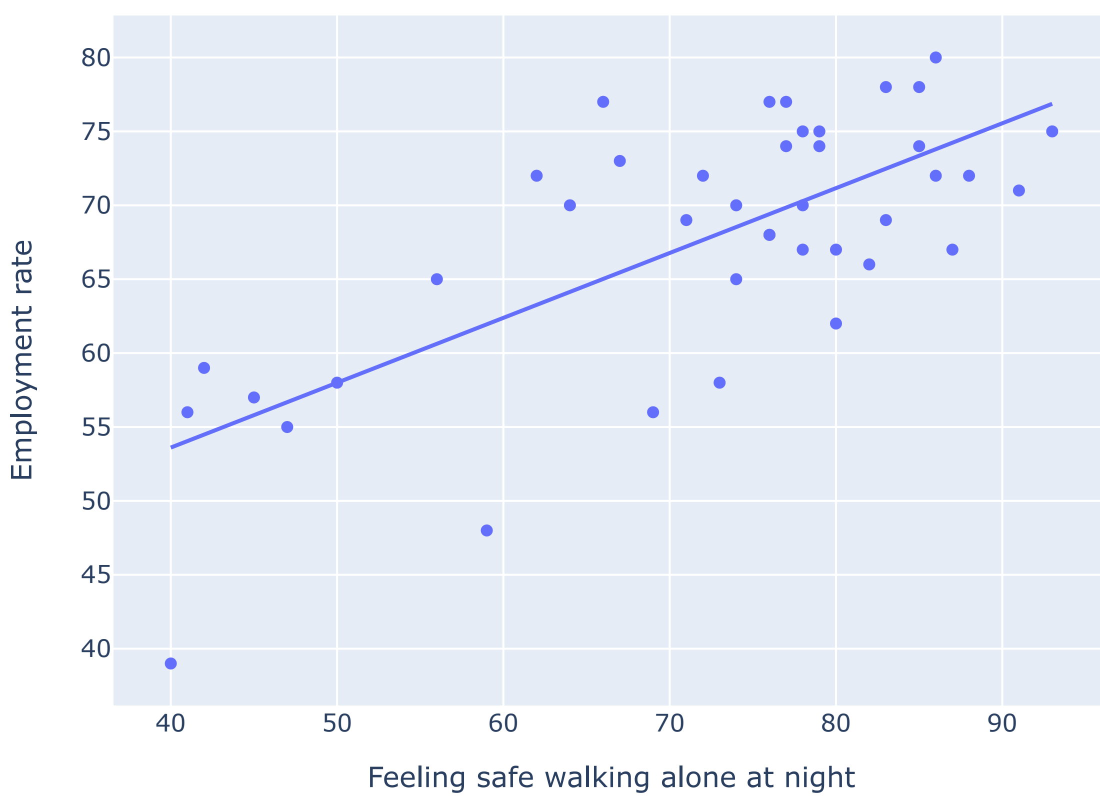
Correlation of "Employment rate" and "Water quality"





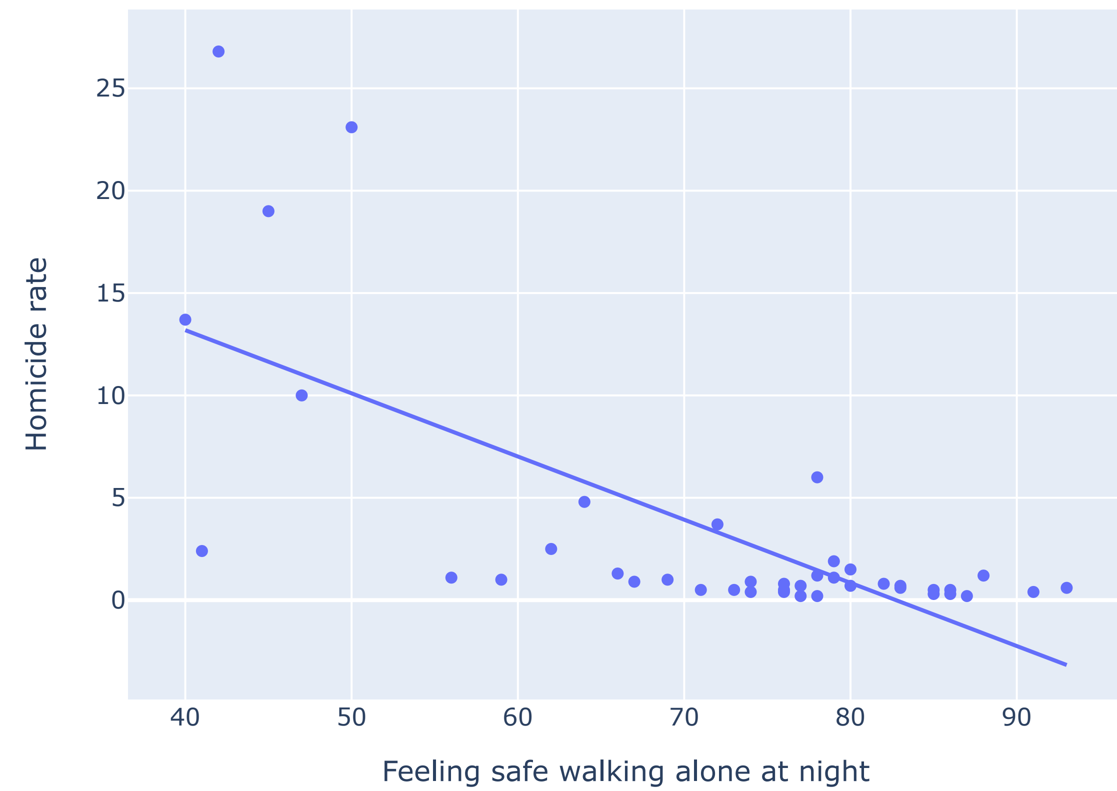

```
Corr_coef of "Employment rate" and "Water quality" = 0.7122696113676107
```

Correlation of "Feeling safe walking alone at night" and "Employment rate"



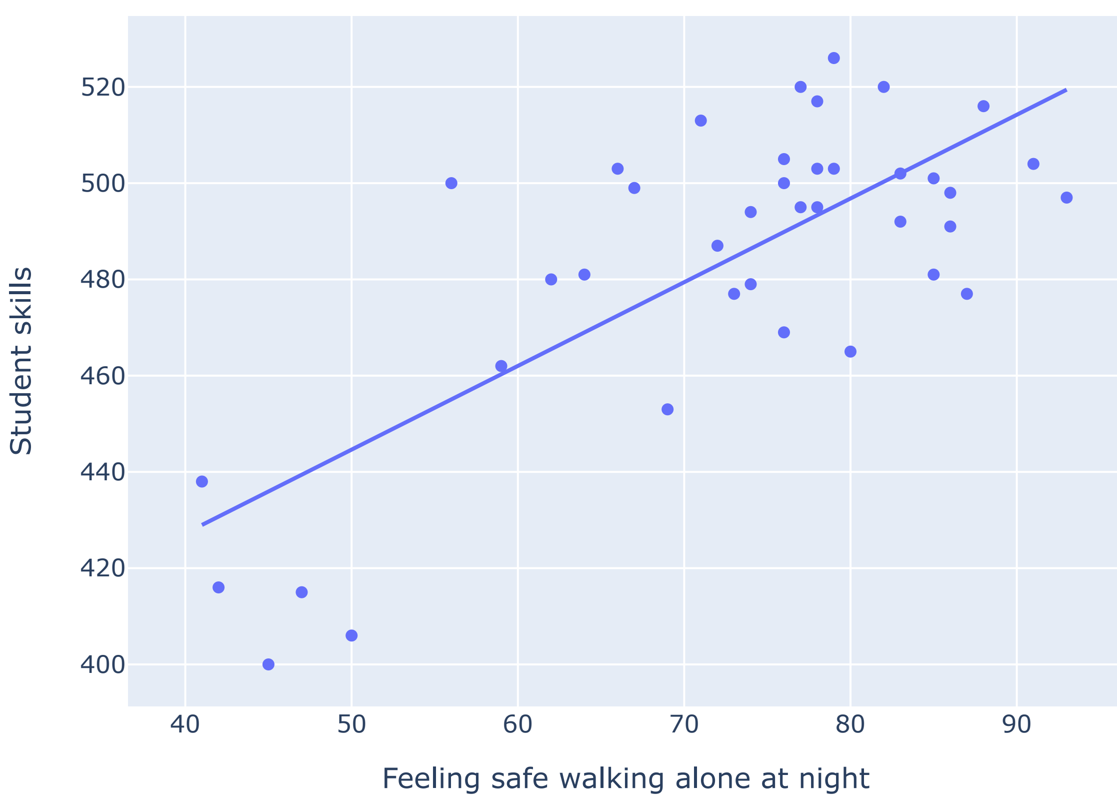

```
Corr_coef of "Feeling safe walking alone at night" and "Employment rate" = 0.71151127441
10984
```

Correlation of "Feeling safe walking alone at night" and "Homicide rate"



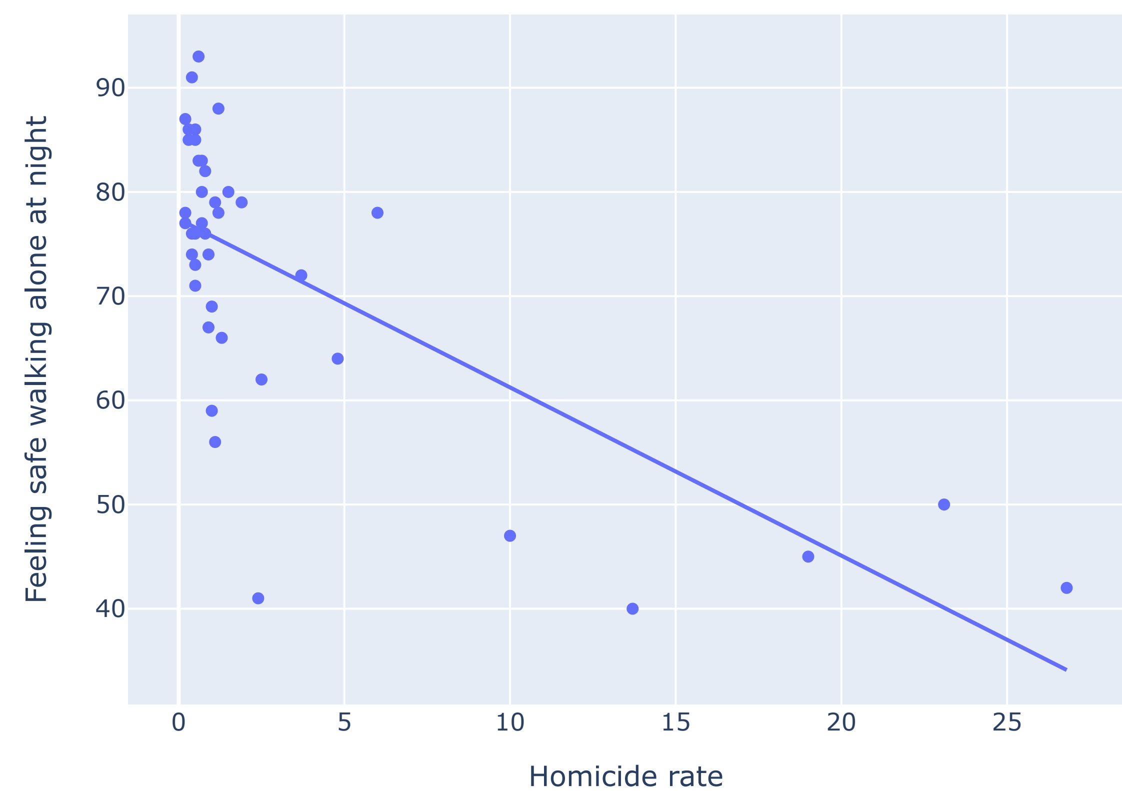

```
Corr_coef of "Feeling safe walking alone at night" and "Homicide rate" = -0.705838797888
8833
```

Correlation of "Feeling safe walking alone at night" and "Student skills"



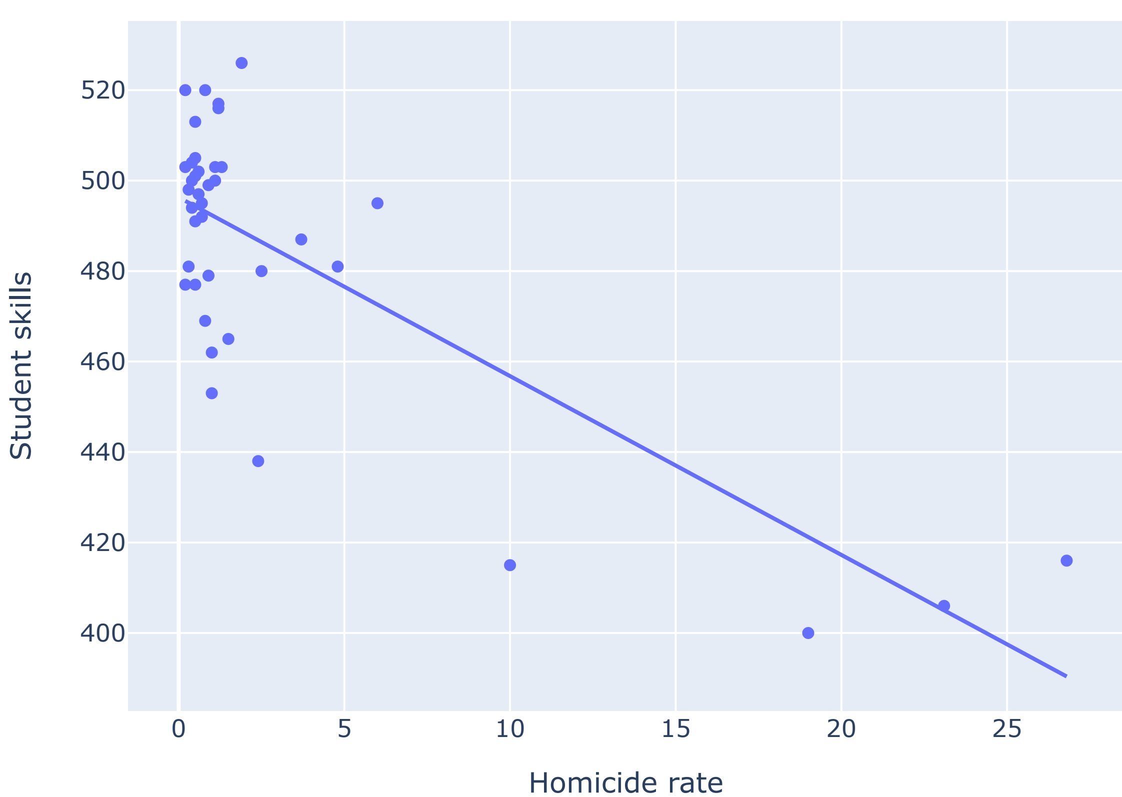

```
Corr_coef of "Feeling safe walking alone at night" and "Student skills" = 0.748058094763
088
```

Correlation of "Homicide rate" and "Feeling safe walking alone at night"

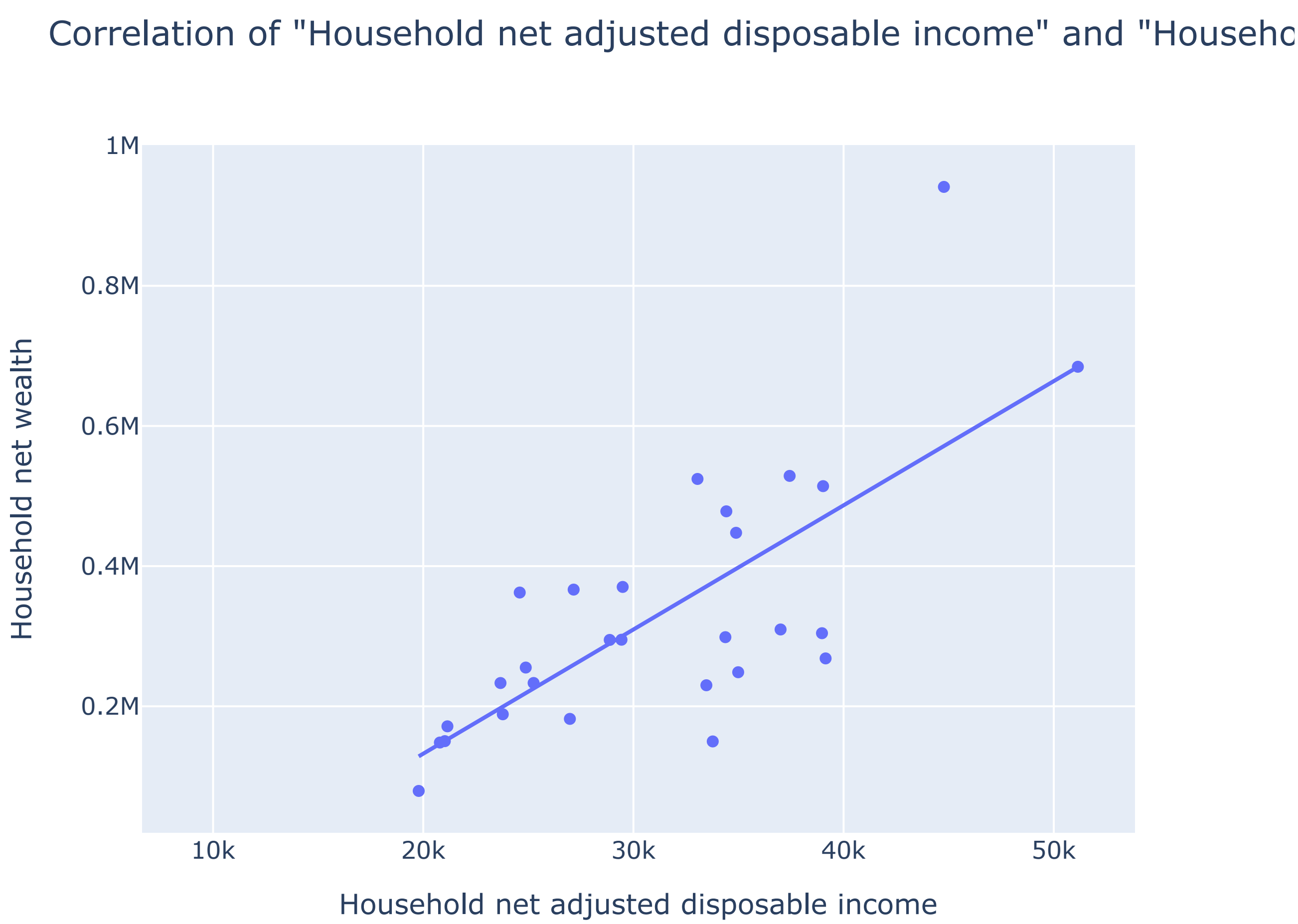



```
Corr_coef of "Homicide rate" and "Feeling safe walking alone at night" = -0.705838797888
8833
```

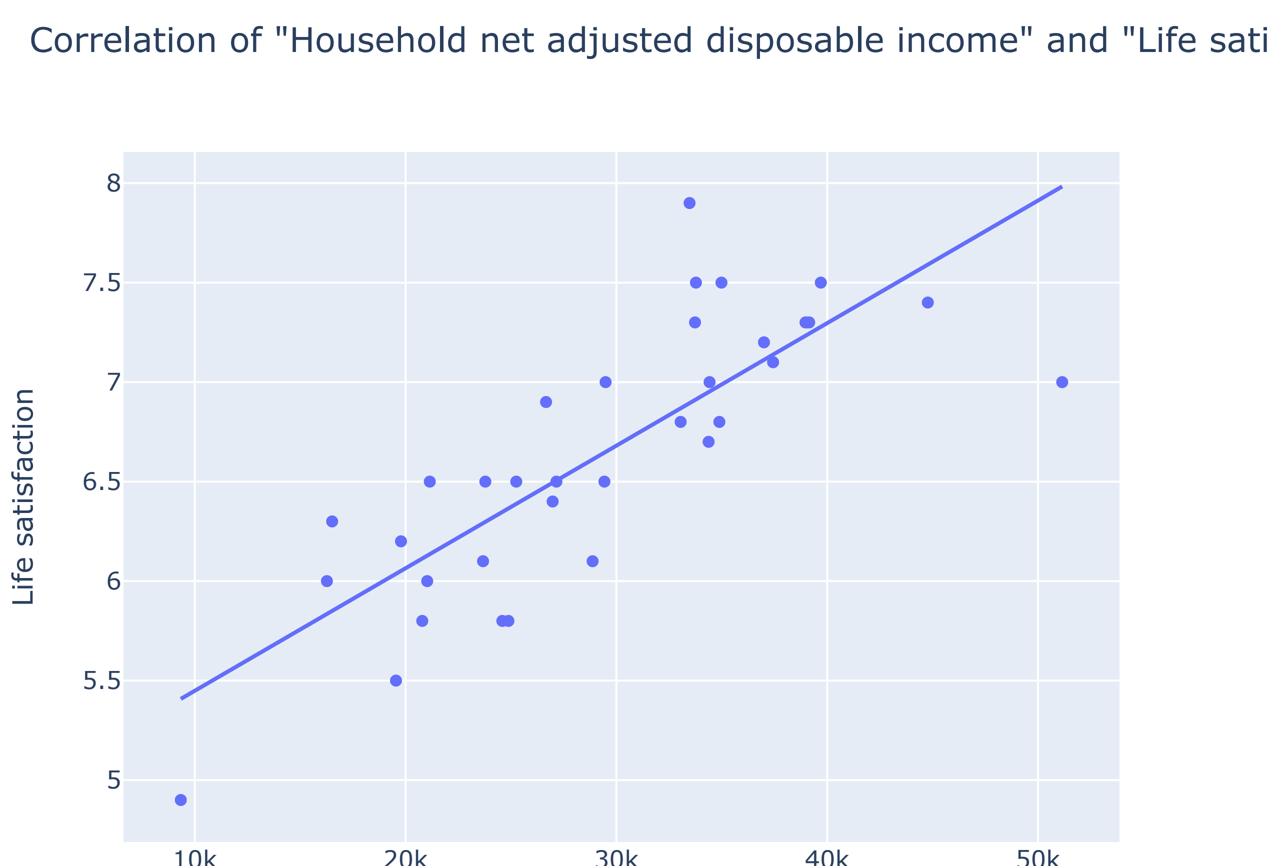
Correlation of "Homicide rate" and "Student skills"



```
Corr_coef of "Homicide rate" and "Student skills" = -0.7687731742066118
```

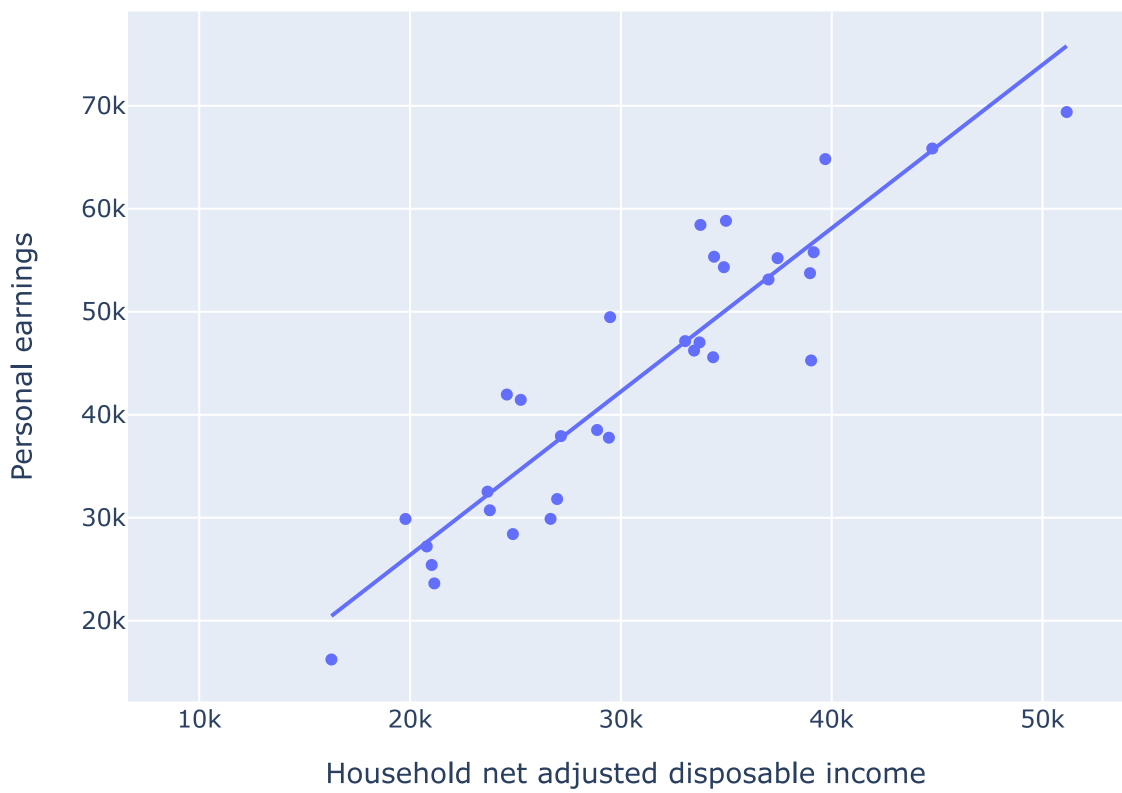



```
Corr_coef of "Household net adjusted disposable income" and "Household net wealth" = 0.7
412799751880902
```



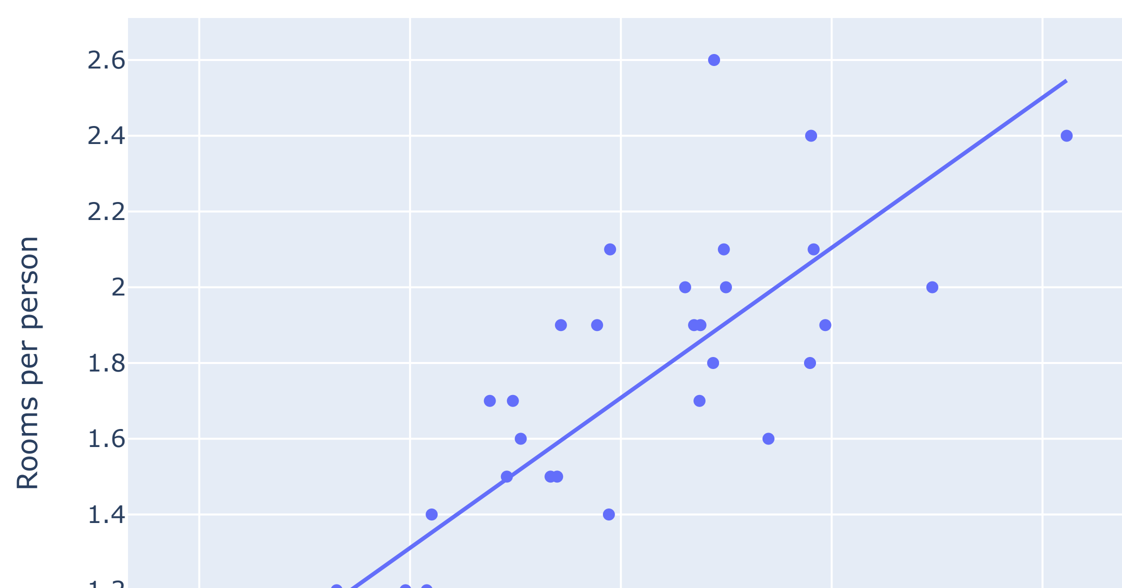

```
Corr_coef of "Household net adjusted disposable income" and "Life satisfaction" = 0.8109
686465037127
```

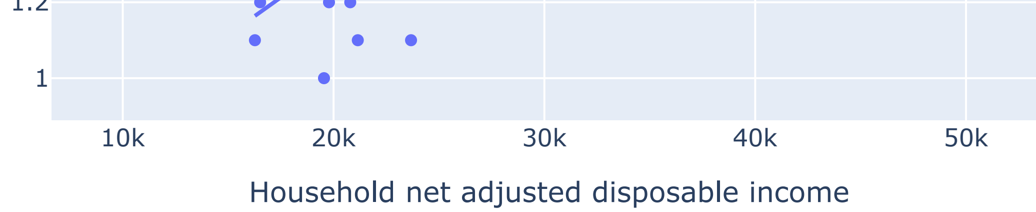
Correlation of "Household net adjusted disposable income" and "Personal




```
Corr_coef of "Household net adjusted disposable income" and "Personal earnings" = 0.9226
125412771972
```

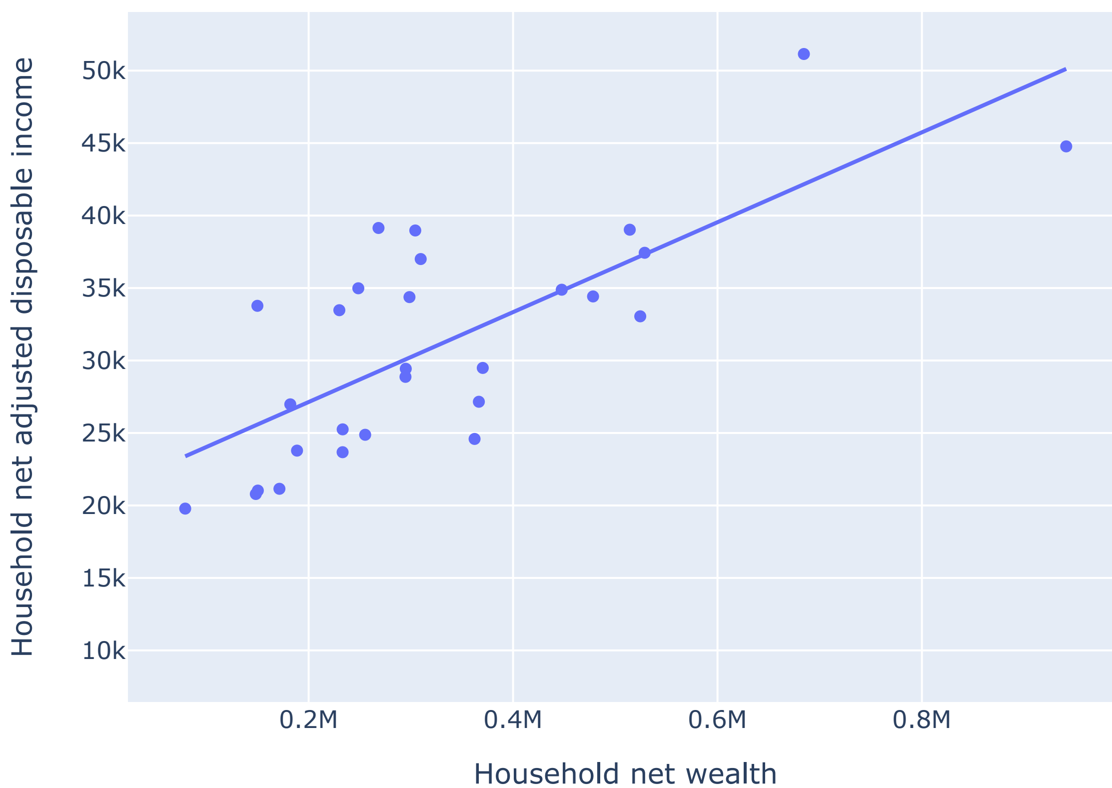
Correlation of "Household net adjusted disposable income" and "Rooms p





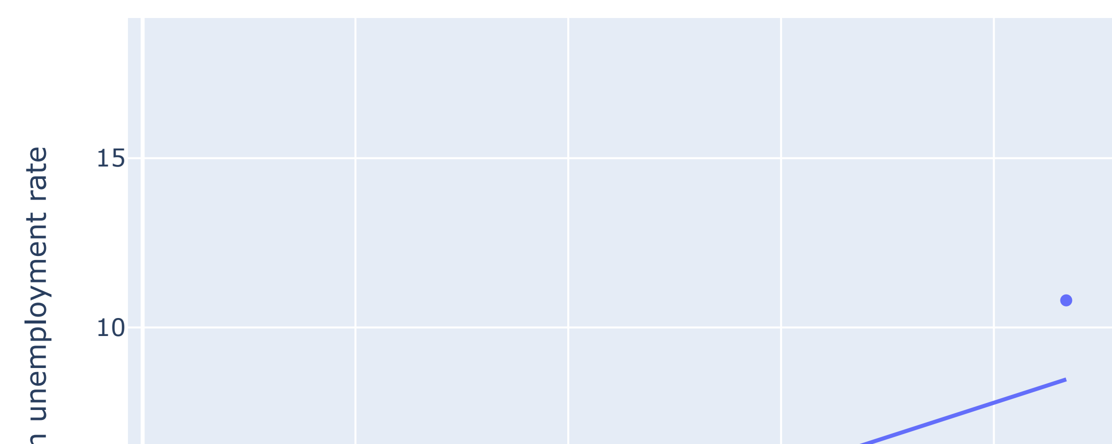

```
Corr_coef of "Household net adjusted disposable income" and "Rooms per person" = 0.80091
51260152061
```

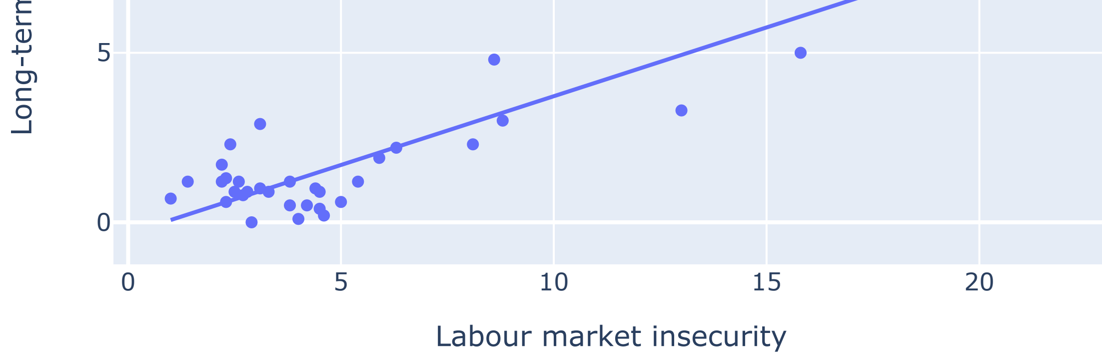
Correlation of "Household net wealth" and "Household net adjusted dispo




```
Corr_coef of "Household net wealth" and "Household net adjusted disposable income" = 0.7
412799751880902
```

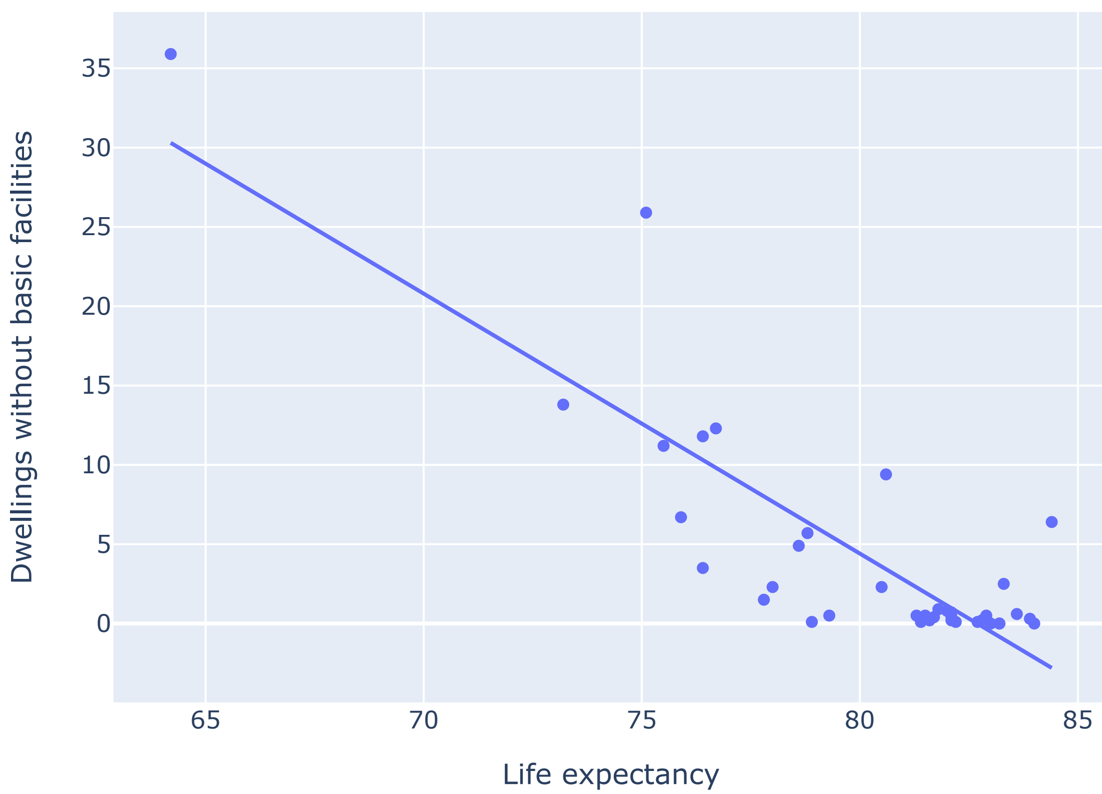
Correlation of "Labour market insecurity" and "Long-term unemployment





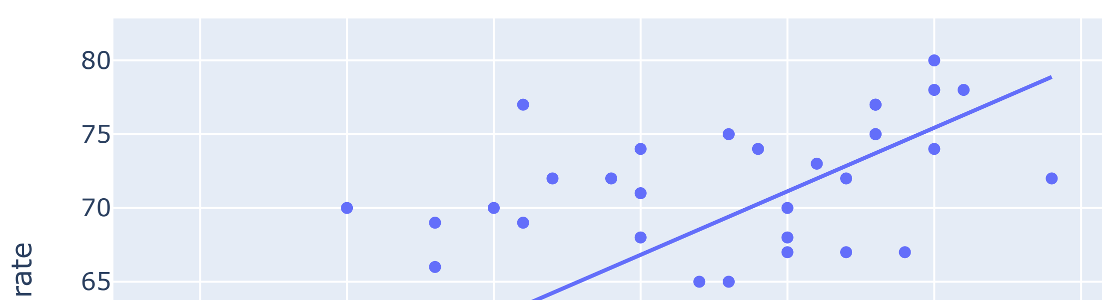

```
Corr_coef of "Labour market insecurity" and "Long-term unemployment rate" = 0.8718381718
523023
```

Correlation of "Life expectancy" and "Dwellings without basic facilities"



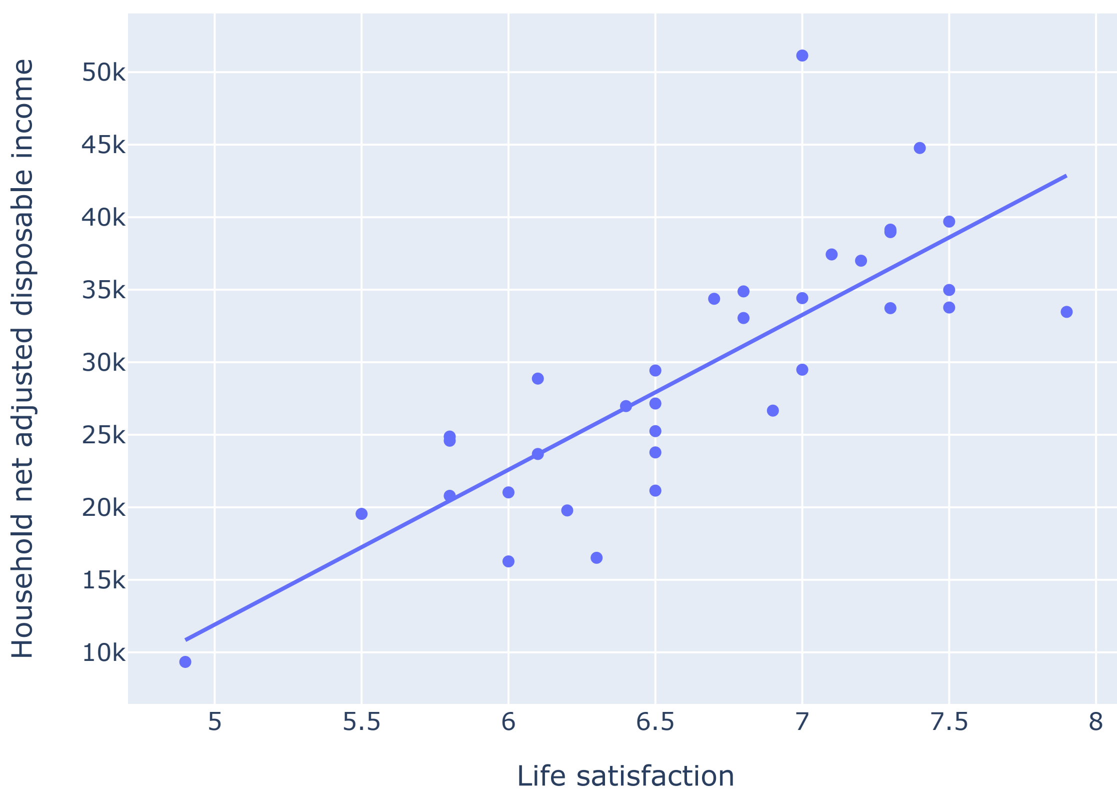

```
Corr_coef of "Life expectancy" and "Dwellings without basic facilities" = -0.84910675670
51078
```

Correlation of "Life satisfaction" and "Employment rate"



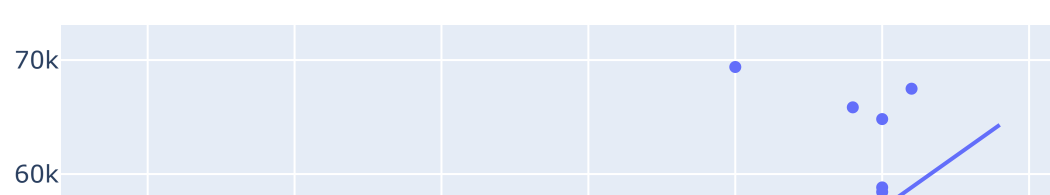

```
Corr_coef of "Life satisfaction" and "Employment rate" = 0.7082618221255973
```

Correlation of "Life satisfaction" and "Household net adjusted disposable



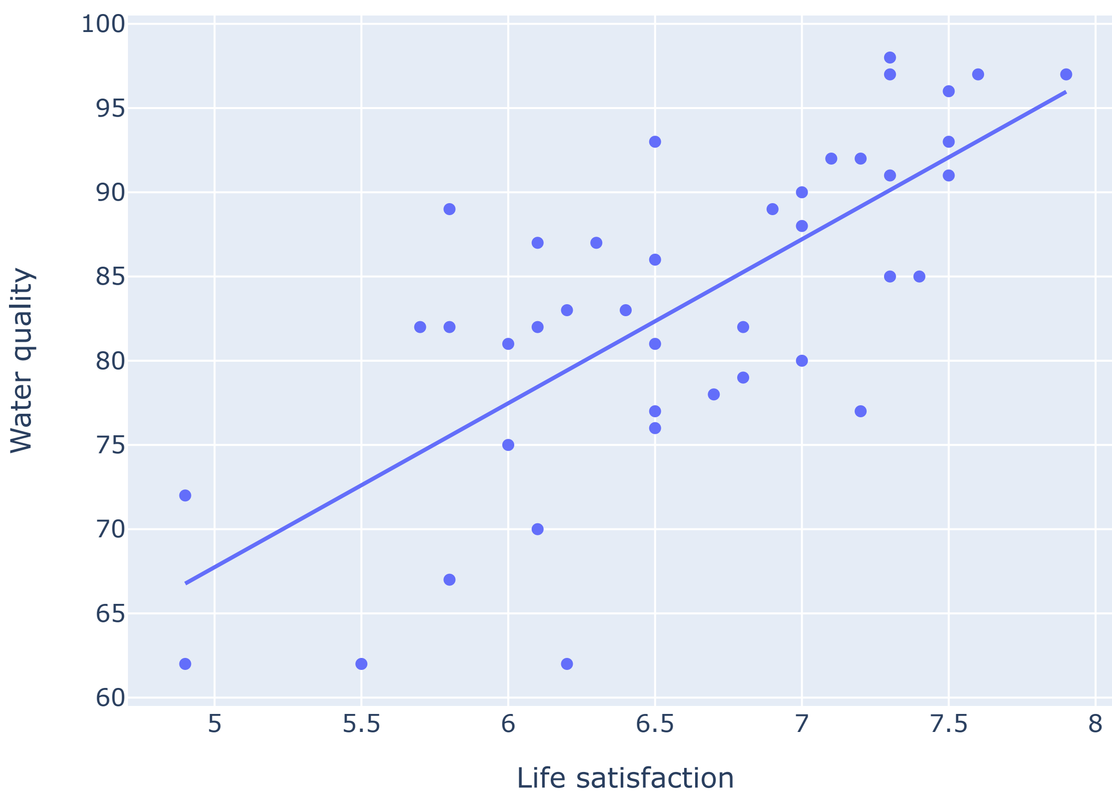

```
Corr_coef of "Life satisfaction" and "Household net adjusted disposable income" = 0.8109
686465037127
```

Correlation of "Life satisfaction" and "Personal earnings"



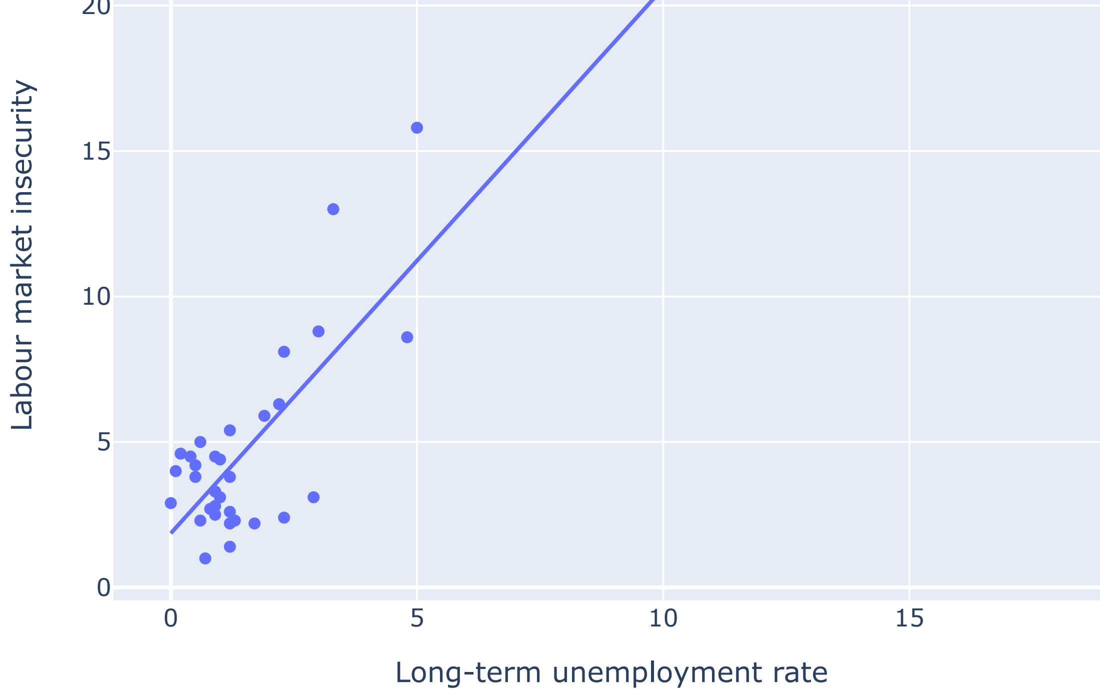

```
Corr_coef of "Life satisfaction" and "Personal earnings" = 0.7721451682502471
```

Correlation of "Life satisfaction" and "Water quality"



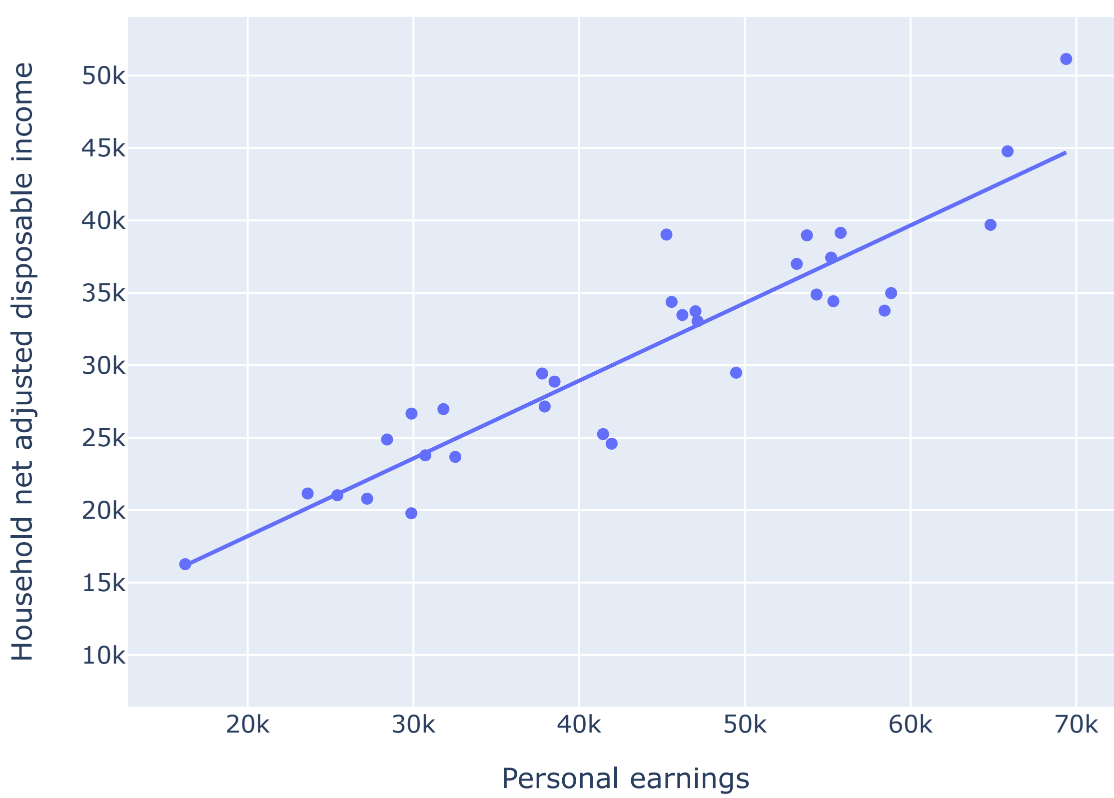

```
Corr_coef of "Life satisfaction" and "Water quality" = 0.726316058357149
```

Correlation of "Long-term unemployment rate" and "Labour market insec




```
Corr_coef of "Long-term unemployment rate" and "Labour market insecurity" = 0.8718381718
523023
```

Correlation of "Personal earnings" and "Household net adjusted disposab

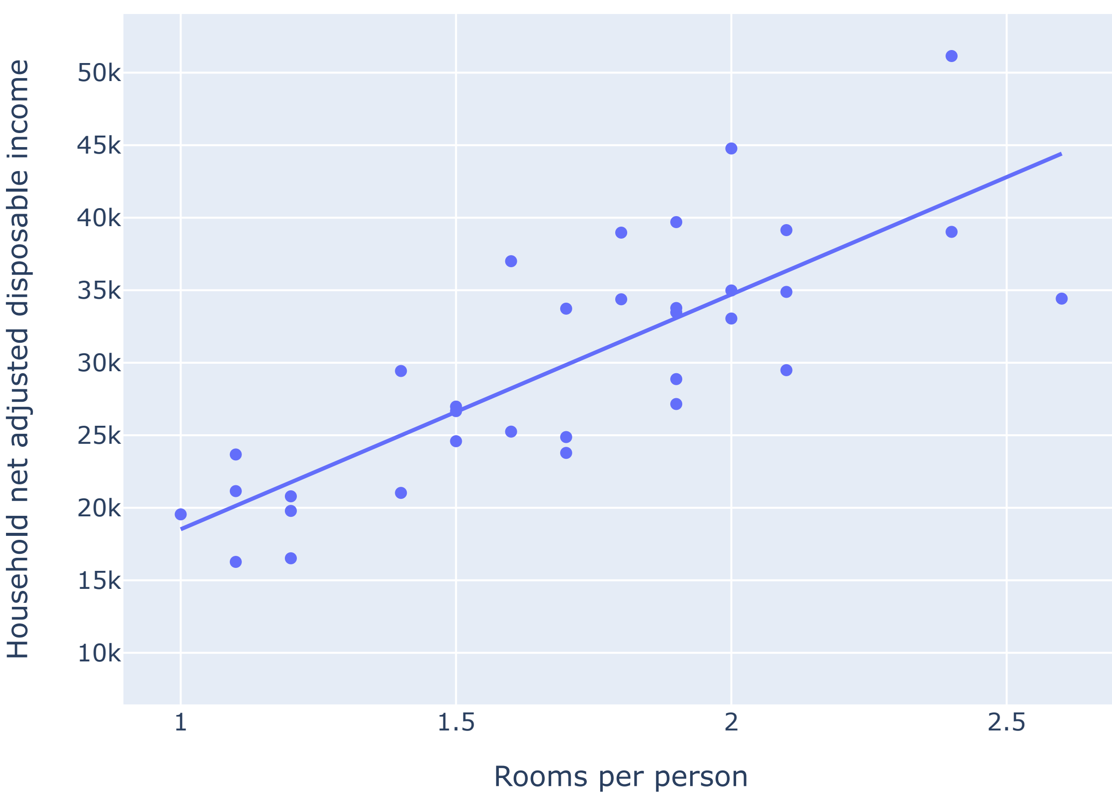



```
Corr_coef of "Personal earnings" and "Household net adjusted disposable income" = 0.9226
125412771972
```

Correlation of "Personal earnings" and "Life satisfaction"

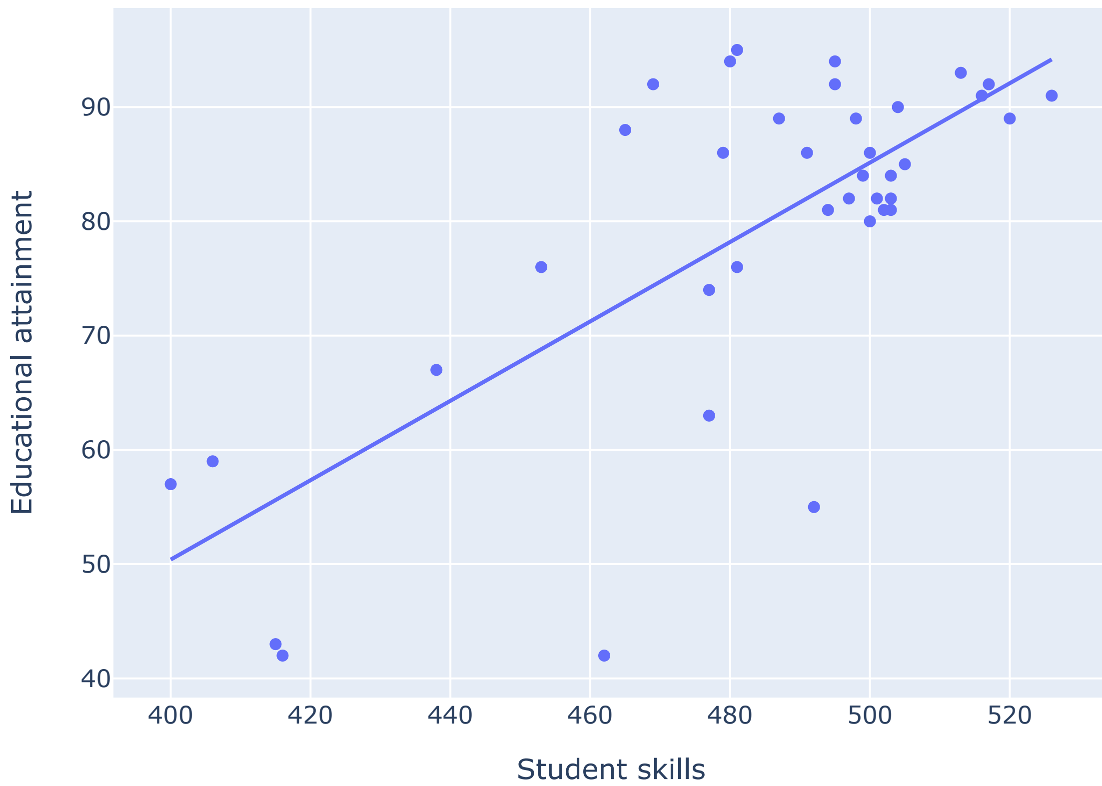

```
Corr_coef of "Personal earnings" and "Life satisfaction" = 0.7721451682502471
```

Correlation of "Rooms per person" and "Household net adjusted disposab



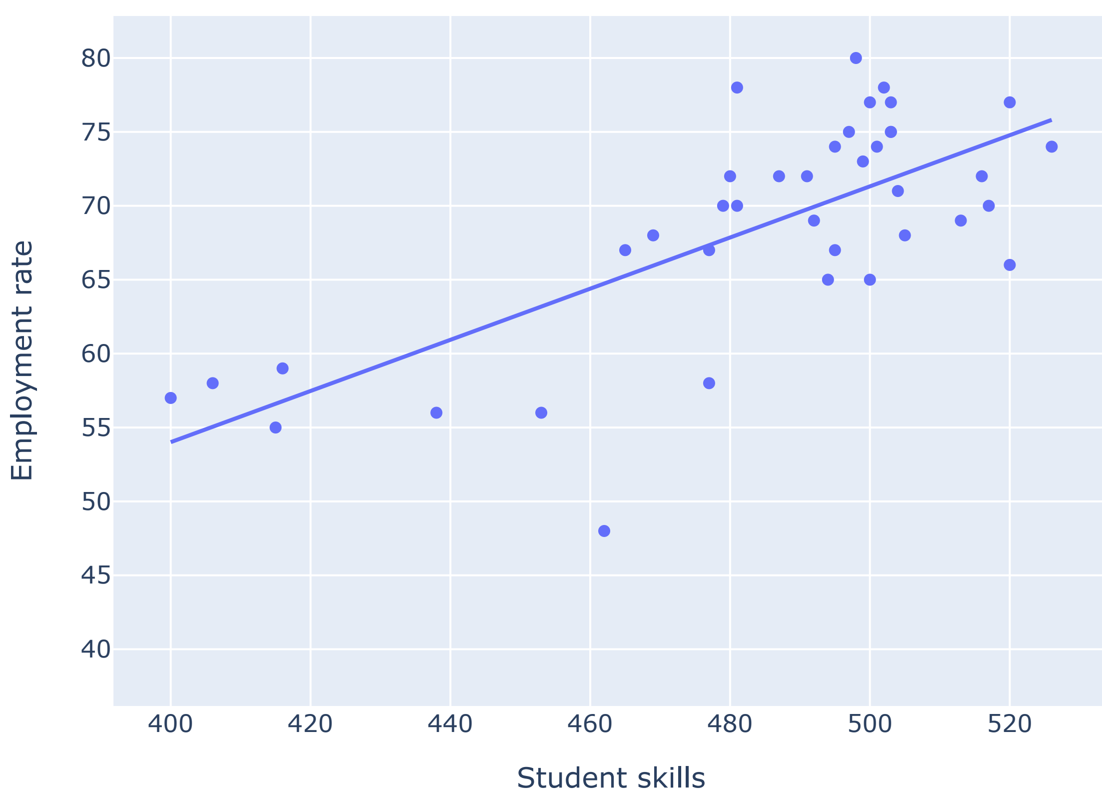

```
Corr_coef of "Rooms per person" and "Household net adjusted disposable income" = 0.80091
51260152061
```

Correlation of "Student skills" and "Educational attainment"

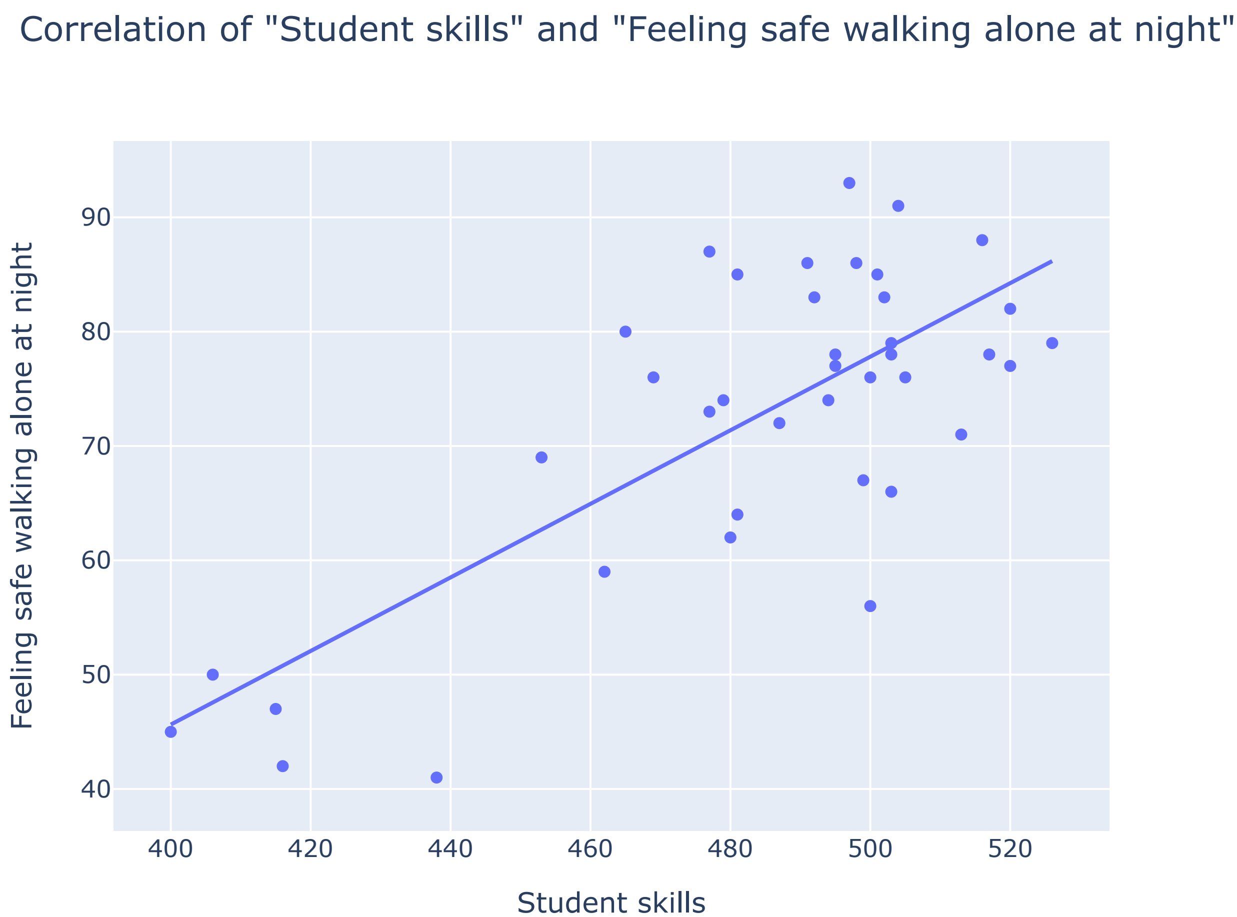


Corr_coef of "Student skills" and "Educational attainment" = 0.7280897847632455

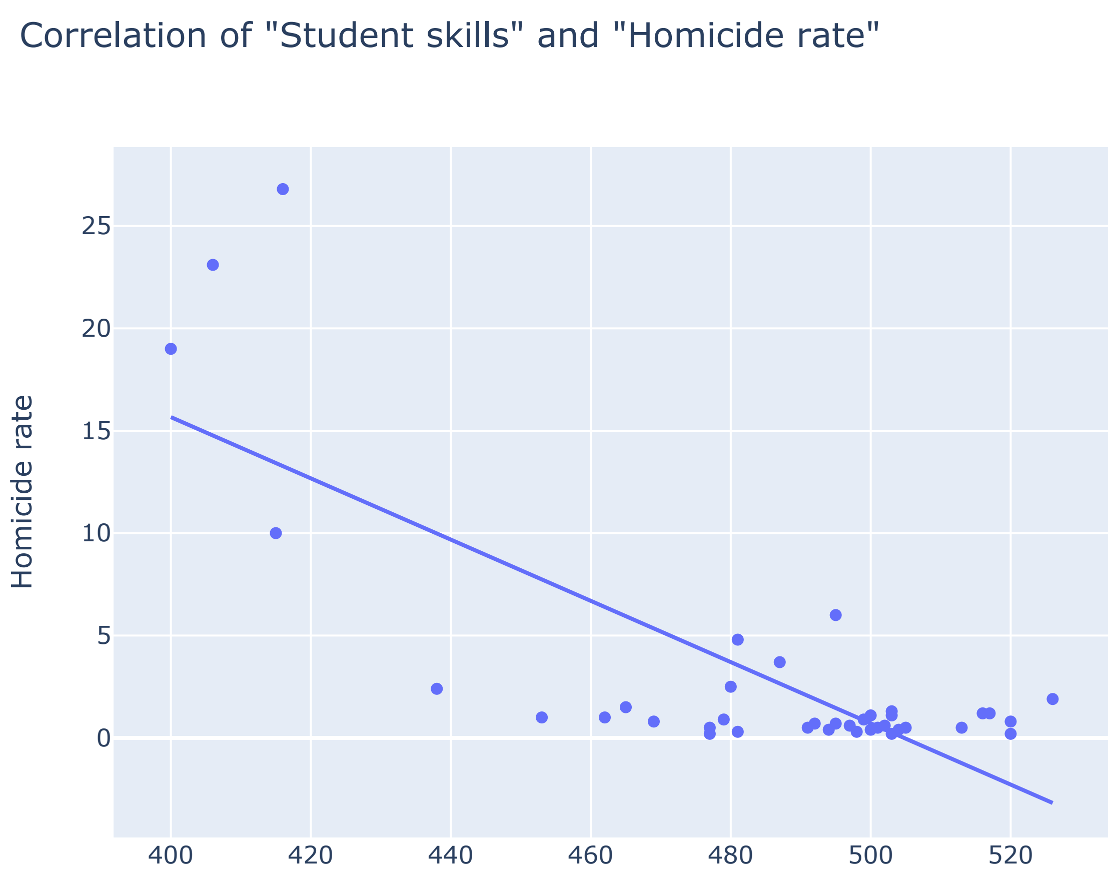
Correlation of "Student skills" and "Employment rate"



```
Corr_coef of "Student skills" and "Employment rate" = 0.717330689097553
```



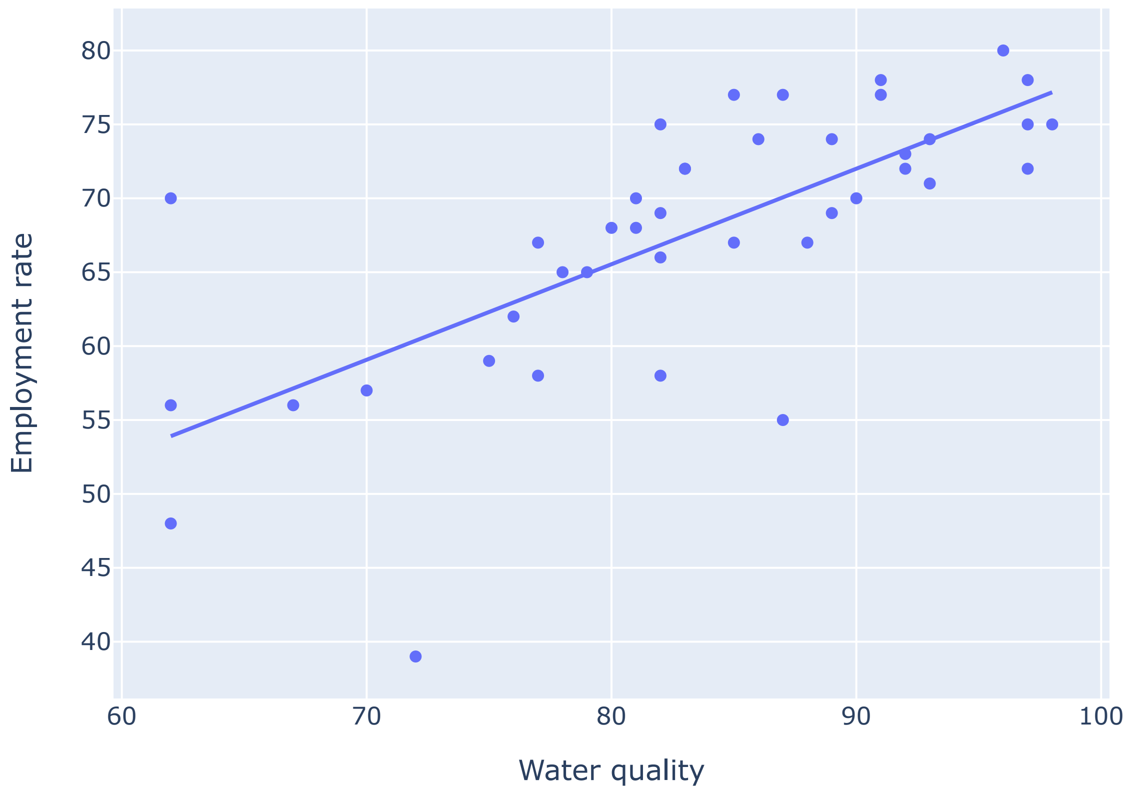

```
Corr_coef of "Student skills" and "Feeling safe walking alone at night" = 0.748058094763
088
```



Student skills

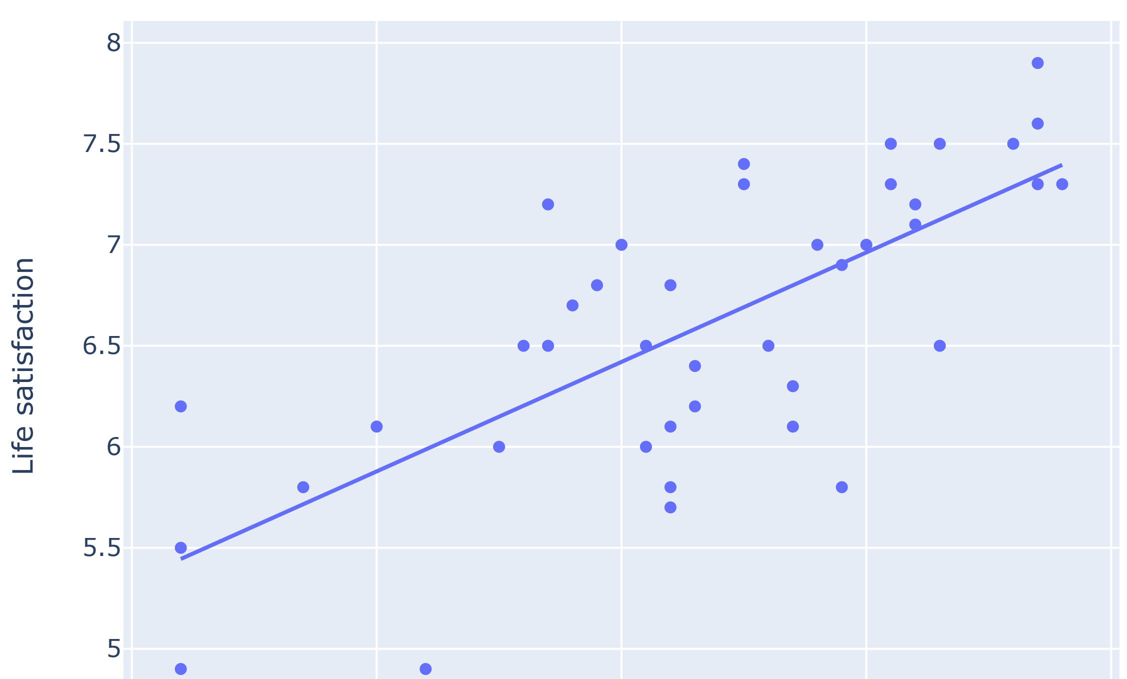
```
Corr_coef of "Student skills" and "Homicide rate" = -0.7687731742066118
```

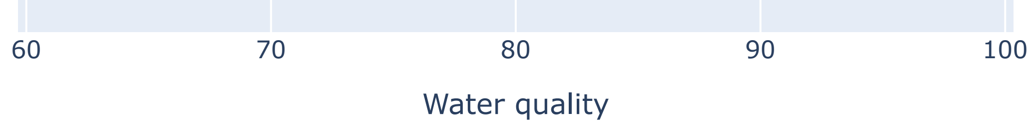
Correlation of "Water quality" and "Employment rate"




```
Corr_coef of "Water quality" and "Employment rate" = 0.7122696113676107
```

Correlation of "Water quality" and "Life satisfaction"






```
Corr_coef of "Water quality" and "Life satisfaction" = 0.726316058357149
```

For ease of future work, we shall generate a dictionary indicating which features have an absolute correlation value greater than 0.7.

In [12]:
```
high_corr_dict = {}
corr_matrix = Rating.corr().abs()
for i in corr_matrix.index:
    for j in corr_matrix.columns:
        if (abs(Rating.corr().loc[i, j]) > 0.7) & (abs(Rating.corr().loc[i, j]) !=1):
            if i not in high_corr_dict:
                high_corr_dict[i] = [j]
            else:
                high_corr_dict[i].append(j)
high_corr_dict
```

Out[12]:
```
{'Dwellings without basic facilities': ['Life expectancy'],
 'Educational attainment': ['Employment rate', 'Student skills'],
 'Employment rate': ['Educational attainment',
  'Feeling safe walking alone at night',
  'Life satisfaction',
  'Student skills',
  'Water quality'],
 'Feeling safe walking alone at night': ['Employment rate',
  'Homicide rate',
  'Student skills'],
 'Homicide rate': ['Feeling safe walking alone at night', 'Student skills'],
 'Household net adjusted disposable income': ['Household net wealth',
  'Life satisfaction',
  'Personal earnings',
  'Rooms per person'],
 'Household net wealth': ['Household net adjusted disposable income'],
 'Labour market insecurity': ['Long-term unemployment rate'],
 'Life expectancy': ['Dwellings without basic facilities'],
 'Life satisfaction': ['Employment rate',
  'Household net adjusted disposable income',
  'Personal earnings',
  'Water quality'],
 'Long-term unemployment rate': ['Labour market insecurity'],
 'Personal earnings': ['Household net adjusted disposable income',
  'Life satisfaction'],
 'Rooms per person': ['Household net adjusted disposable income'],
 'Student skills': ['Educational attainment',
  'Employment rate',
  'Feeling safe walking alone at night',
  'Homicide rate'],
 'Water quality': ['Employment rate', 'Life satisfaction']}
```

Data Wrangling

Handle missing values

We will create a loop that will fill in the missing values in the dataframe utilizing a unique regression model for each feature. The correlated features will vary for each feature. Additionally, we shall use a unique regression model for each missing value, as the corresponding value may also be absent among the correlated features. Linear regression may produce negative values, which are inappropriate for our analysis. Therefore, we have resolved to replace negative values with the minimum value of the feature in these instances.

In [13]:
```
reg = linear_model.LinearRegression()
for key, items in high_corr_dict.items(): #iterate through the dict created in the previ
    if Rating[key].isna().any():
        predict = Rating.loc[Rating[key].isna(), [key] + items] #form dataframe with onl
        for row_ind in range(len(predict.index)):
            predict_row = predict.iloc[row_ind, 1:].dropna().to_frame().transpose() #get
            if not predict_row.empty:
                train = Rating.loc[:, [key] + predict_row.columns.tolist()].dropna() #fo
                reg.fit(train[predict_row.columns], train[key])
                predicted = reg.predict(predict_row)
                print(f'Pridicted value: {key} - {predict_row.index[0]} = {predicted}')
                if predicted >= 0: #check if value of feature >=0. If not we fill the mi
                    Rating.loc[predict_row.index, key] = predicted
                else:
                    Rating.loc[predict_row.index, key] = Rating[key].min()
```

```
Pridicted value: Dwellings without basic facilities - Australia = [-0.50468298]
Pridicted value: Dwellings without basic facilities - Israel = [-0.32104909]
Pridicted value: Dwellings without basic facilities - New Zealand = [0.99864883]
Pridicted value: Educational attainment - Japan = [93.90552215]
Pridicted value: Household net adjusted disposable income - Brazil = [23654.63841578]
Pridicted value: Household net adjusted disposable income - Chile = [22012.48807711]
Pridicted value: Household net adjusted disposable income - Colombia = [16586.7753286]
Pridicted value: Household net adjusted disposable income - Iceland = [41825.30166147]
Pridicted value: Household net adjusted disposable income - Israel = [28160.73196938]
Pridicted value: Household net adjusted disposable income - Türkiye = [10844.89115628]
Pridicted value: Household net wealth - Brazil = [194843.81009083]
Pridicted value: Household net wealth - Colombia = [68416.05355773]
Pridicted value: Household net wealth - Costa Rica = [67167.93398498]
Pridicted value: Household net wealth - Czech Republic = [248674.34260066]
Pridicted value: Household net wealth - Iceland = [519875.03388992]
Pridicted value: Household net wealth - Israel = [275447.42276454]
Pridicted value: Household net wealth - Mexico = [62731.78642051]
Pridicted value: Household net wealth - Russia = [121349.75242362]
Pridicted value: Household net wealth - South Africa = [-61247.80539139]
Pridicted value: Household net wealth - Sweden = [374291.81648731]
Pridicted value: Household net wealth - Switzerland = [476152.13501893]
Pridicted value: Household net wealth - Türkiye = [-16370.90790552]
Pridicted value: Labour market insecurity - Colombia = [3.92362841]
Pridicted value: Labour market insecurity - Costa Rica = [4.67306733]
Pridicted value: Labour market insecurity - Lithuania = [6.54666461]
Pridicted value: Labour market insecurity - Russia = [3.92362841]
Pridicted value: Labour market insecurity - South Africa = [35.40006273]
Pridicted value: Labour market insecurity - Switzerland = [5.04778678]
Pridicted value: Long-term unemployment rate - Chile = [2.68342068]
Pridicted value: Personal earnings - Brazil = [31040.1507386]
Pridicted value: Personal earnings - Colombia = [19512.23061536]
Pridicted value: Personal earnings - Costa Rica = [21477.71964118]
Pridicted value: Personal earnings - Russia = [23073.66994545]
Pridicted value: Personal earnings - South Africa = [6347.3532691]
Pridicted value: Personal earnings - Türkiye = [8511.53373956]
Pridicted value: Rooms per person - Australia = [1.96005331]
Pridicted value: Rooms per person - Brazil = [1.45739839]
Pridicted value: Rooms per person - South Africa = [0.93510634]
Pridicted value: Student skills - South Africa = [403.29317733]
Pridicted value: Student skills - Spain = [477.48205627]
```

We shall verify the results of missing value imputation by displaying the number of missing values for each country and feature on the screen. We shall then generate a dataset from the remaining missing values to investigate them.

In [14]:
```
nan_cols = Rating.isna().sum().sort_values(ascending=False)
nan_rows = Rating.isna().sum(axis=1).sort_values(ascending=False)
print(nan_rows[nan_rows > 0], nan_cols[nan_cols > 0])
Rating_miss = Rating.loc[Rating.isna().any(axis=1), Rating.isna().any()]
Rating_miss
```

```
Country
Brazil          4
South Africa    3
Japan           1
Korea           1
Colombia        1
Costa Rica      1
Israel          1
Iceland         1
Russia          1
dtype: int64 Indicator
Housing expenditure                                    4
Years in education                                     2
Employees working very long hours                      2
Stakeholder engagement for developing regulations      2
Self-reported health                                   2
Long-term unemployment rate                            1
Labour market insecurity                               1
dtype: int64
```

Out[14]:

Indicator Country	Employees working very long hours	Housing expenditure	Labour market insecurity	Long-term unemployment rate	Self-reported health	Stakeholder engagement for developing regulations	Years in education
Brazil	5.6	NaN	NaN	NaN	NaN	2.2	16.0
Colombia	23.7	NaN	3.923628	1.1	80.0	1.4	14.0
Costa Rica	22.0	17.0	4.673067	1.5	73.0	1.8	NaN
Iceland	11.7	NaN	1.000000	0.7	77.0	2.1	19.0
Israel	14.1	NaN	4.600000	0.2	74.0	2.5	16.0
Japan	NaN	21.8	2.700000	0.8	37.0	1.4	16.0
Korea	NaN	14.7	2.900000	0.0	34.0	2.9	17.0
Russia	0.1	17.4	3.923628	1.1	43.0	NaN	16.0
South Africa	15.4	18.1	35.400063	17.9	NaN	NaN	NaN

The dataset contains missing values in columns that do not exhibit significant correlation with other columns or where the corresponding values in columns with significant correlation are absent. To fill in the missing values, we will employ the method of nearest neighbors. However, the nearest neighbors method is highly sensitive to non-normalized data. Hence, we will first normalize our dataset using the min-max normalization method.

In [15]:
```
min_max_scaler = MinMaxScaler()
Rating_normalized = pd.DataFrame(min_max_scaler.fit_transform(Rating), columns=Rating.co
```

Next, using the normalized dataset, we will fill in the missing values, while selecting the number of nearest neighbors to be 5. We will display the dataset on the screen to investigate the filled-in values.

In [16]:
```
imputer = KNNImputer(n_neighbors=5)
Rating_normalized_imputed = pd.DataFrame(imputer.fit_transform(Rating_normalized), colum
Rating_normalized_imputed
```

Out[16]:

Indicator	Air pollution_norm	Dwellings without basic facilities_norm	Educational attainment_norm	Employees working very long hours_norm	Employment rate_norm	Feeling safe walking alone at night_norm	Homicide rate_norm
Country							
Australia	0.052174	0.000000	0.792453	0.460967	0.829268	0.509434	0.026316
Austria	0.291304	0.022284	0.830189	0.193309	0.804878	0.867925	0.011278
Belgium	0.317391	0.019499	0.716981	0.156134	0.634146	0.301887	0.033835
Brazil	0.269565	0.186630	0.283019	0.204461	0.439024	0.094340	0.706767
Canada	0.069565	0.005571	0.943396	0.118959	0.756098	0.716981	0.037594
Chile	0.778261	0.261838	0.471698	0.282528	0.414634	0.018868	0.082707
Colombia	0.743478	0.342618	0.320755	0.877323	0.463415	0.188679	0.860902
Costa Rica	0.521739	0.064067	0.018868	0.814126	0.390244	0.132075	0.368421
Czech Republic	0.500000	0.013928	0.981132	0.163569	0.853659	0.698113	0.018797
Denmark	0.195652	0.013928	0.754717	0.037175	0.853659	0.849057	0.011278
Estonia	0.017391	0.158774	0.924528	0.078067	0.853659	0.735849	0.063910
Finland	0.000000	0.011142	0.924528	0.130112	0.804878	0.905660	0.037594
France	0.256522	0.013928	0.735849	0.282528	0.634146	0.641509	0.007519
Germany	0.282609	0.002786	0.830189	0.141264	0.926829	0.679245	0.007519
Greece	0.391304	0.011142	0.641509	0.163569	0.414634	0.547170	0.030075
Hungary	0.486957	0.097493	0.830189	0.052045	0.756098	0.641509	0.026316
Iceland	0.039130	0.000000	0.641509	0.431227	0.951220	0.849057	0.003759
Ireland	0.100000	0.005571	0.811321	0.171004	0.707317	0.679245	0.011278
Israel	0.617391	0.000000	0.867925	0.520446	0.682927	0.754717	0.048872
Italy	0.452174	0.016713	0.396226	0.118959	0.463415	0.622642	0.011278
Japan	0.356522	0.178273	0.979349	0.101859	0.926829	0.698113	0.000000
Korea	0.947826	0.069638	0.886792	0.193309	0.658537	0.792453	0.022556
Latvia	0.313043	0.311978	0.886792	0.055762	0.804878	0.603774	0.131579
Lithuania	0.217391	0.328691	0.981132	0.033457	0.804878	0.415094	0.086466
Luxembourg	0.195652	0.002786	0.603774	0.100372	0.682927	0.886792	0.000000
Mexico	0.643478	0.721448	0.000000	1.000000	0.487805	0.037736	1.000000
Netherlands	0.291304	0.002786	0.735849	0.007435	0.951220	0.811321	0.015038
New Zealand	0.021739	0.027818	0.735849	0.516729	0.926829	0.490566	0.041353

Norway	0.052174	0.000000	0.754717	0.048327	0.878049	1.000000	0.015038
Poland	0.752174	0.064067	0.962264	0.152416	0.731707	0.584906	0.011278
Portugal	0.121739	0.025070	0.245283	0.204461	0.731707	0.811321	0.018797
Russia	0.273913	0.384401	1.000000	0.000000	0.756098	0.452830	0.172932
Slovak Republic	0.565217	0.041783	0.943396	0.152416	0.707317	0.679245	0.022556
Slovenia	0.500000	0.005571	0.905660	0.204461	0.780488	0.962264	0.007519
South Africa	1.000000	1.000000	0.113208	0.568773	0.000000	0.000000	0.507519
Spain	0.195652	0.008357	0.396226	0.089219	0.560976	0.754717	0.018797
Sweden	0.013043	0.000000	0.792453	0.029740	0.878049	0.735849	0.033835
Switzerland	0.200000	0.000000	0.886792	0.011152	1.000000	0.867925	0.003759
Türkiye	0.939130	0.136490	0.000000	0.925651	0.219512	0.358491	0.030075
United Kingdom	0.200000	0.013928	0.754717	0.397770	0.878049	0.716981	0.000000
United States	0.095652	0.002786	0.943396	0.382900	0.682927	0.716981	0.218045

Although all values seem plausible, we have observed that the "Employees working very long hours_norm" feature in Japan is not consistent with the OCID study on the distribution of citizens use of their time, available at https://stats.oecd.org/Index.aspx?DataSetCode=BLI. Therefore, we will assign the maximum value of this feature in Japan relative to our sample.

In [17]:
```
Rating_normalized_imputed.loc['Japan','Employees working very long hours_norm'] = Rating
```

Once we convert the normalized values back to their original scale, we can fill our dataset with the predicted values.

In [18]:
```
Rating_nonnormalized_imputed = pd.DataFrame(min_max_scaler.inverse_transform(Rating_norm
for i in Rating_miss.index:
    for c in Rating_miss.columns:
        if pd.isna(Rating.loc[i, c]):
            Rating.loc[i, c] = Rating_nonnormalized_imputed.loc[i, c + '_norm']
```

Finally, we check if there are any remaining missing values.

In [19]:
```
nan_cols = Rating.isna().sum().sort_values(ascending=False)
nan_rows = Rating.isna().sum(axis=1).sort_values(ascending=False)
print(nan_rows[nan_rows > 0], nan_cols[nan_cols > 0])
```

```
Series([], dtype: int64) Series([], dtype: int64)
```

Preparing Data

To enable subsequent comparative analyses, we need to define some overall features for comparing countries. These overall features already exist in the OECD-collected data, and they consist of groups of existing features in our dataset. The following are the overall features and their specific constituent features:

'Housing': 'Dwellings without basic facilities', 'Housing expenditure', 'Rooms per person'.
'Income': 'Household net adjusted disposable income', 'Household net wealth'.

'Jobs': 'Labour market insecurity', 'Employment rate', 'Long-term unemployment rate', 'Personal earnings'.
'Community': 'Quality of support network'.
'Education': 'Educational attainment', 'Student skills', 'Years in education'.
'Environment': 'Air pollution', 'Water quality'.
'Civic engagement': 'Stakeholder engagement for developing regulations', 'Voter turnout'.
'Health': 'Life expectancy', 'Self-reported health'.
'Life satisfaction': 'Life satisfaction'.
'Safety': 'Feeling safe walking alone at night', 'Homicide rate'.
'Work-life balance': 'Employees working very long hours'.

Further information regarding project features and other related details can be accessed via the following link: __.

Certain features exhibit negative implications, such as a high level of air pollution, which is considered undesirable. To facilitate subsequent comparative analyses, such features need to be inverted. This can be achieved by subtracting the normalized value of the feature from 1.

In [20]:
```
Rating_normalized_imputed = np.around(Rating_normalized_imputed, decimals=2)
Rating_normalized_imputed['Air pollution_norm'] = 1 - Rating_normalized_imputed['Air pol
Rating_normalized_imputed['Dwellings without basic facilities_norm'] = 1 - Rating_normal
Rating_normalized_imputed['Employees working very long hours_norm'] = 1 - Rating_normali
Rating_normalized_imputed['Homicide rate_norm'] = 1 - Rating_normalized_imputed['Homicid
Rating_normalized_imputed['Housing expenditure_norm'] = 1 - Rating_normalized_imputed['H
Rating_normalized_imputed['Labour market insecurity_norm'] = 1 - Rating_normalized_imput
Rating_normalized_imputed['Long-term unemployment rate_norm'] = 1 - Rating_normalized_im
```

We will create columns in the dataframe for overall features by adding the normalized values of the individual features that constitute them.

In [21]:
```
Rating_normalized_imputed['Housing'] = Rating_normalized_imputed[['Dwellings without bas
Rating_normalized_imputed['Income'] = Rating_normalized_imputed[['Household net adjusted
Rating_normalized_imputed['Jobs'] = Rating_normalized_imputed[['Labour market insecurity
          'Employment rate_norm', 'Long-term unemployment rate_norm', 'Personal earnings
Rating_normalized_imputed['Community'] = Rating_normalized_imputed['Quality of support n
Rating_normalized_imputed['Education'] = Rating_normalized_imputed[['Educational attainm
Rating_normalized_imputed['Environment'] = Rating_normalized_imputed[['Air pollution_nor
Rating_normalized_imputed['Civic engagement'] = Rating_normalized_imputed[['Stakeholder
Rating_normalized_imputed['Health'] = Rating_normalized_imputed[['Life expectancy_norm',
          'Self-reported health_norm']].sum(axis=1)
Rating_normalized_imputed['Life satisfaction_ov'] = Rating_normalized_imputed['Life sati
Rating_normalized_imputed['Safety'] = Rating_normalized_imputed[['Feeling safe walking a
Rating_normalized_imputed['Work-life balance'] = Rating_normalized_imputed['Employees wo
```

The resultant values of the overall features must be normalized to avoid any feature having a disproportionate influence.

In [22]:
```
Rating_normalized_imputed[['Housing', 'Income', 'Jobs', 'Community', 'Education', 'Envir
                           'Health', 'Life satisfaction_ov', 'Safety', 'Work-life balanc
                           'Health', 'Life satisfaction_ov', 'Safety', 'Work-life balanc
Rating_normalized_imputed
```

Out[22]:

Indicator	Air pollution_norm	Dwellings without basic facilities_norm	Educational attainment_norm	Employees working very long hours_norm	Employment rate_norm	Feeling safe walking alone at night_norm	Homicide rate_norm

Country							
Australia	0.95	1.00	0.79	0.54	0.83	0.51	0.97
Austria	0.71	0.98	0.83	0.81	0.80	0.87	0.99
Belgium	0.68	0.98	0.72	0.84	0.63	0.30	0.97
Brazil	0.73	0.81	0.28	0.80	0.44	0.09	0.29
Canada	0.93	0.99	0.94	0.88	0.76	0.72	0.96
Chile	0.22	0.74	0.47	0.72	0.41	0.02	0.92
Colombia	0.26	0.66	0.32	0.12	0.46	0.19	0.14
Costa Rica	0.48	0.94	0.02	0.19	0.39	0.13	0.63
Czech Republic	0.50	0.99	0.98	0.84	0.85	0.70	0.98
Denmark	0.80	0.99	0.75	0.96	0.85	0.85	0.99
Estonia	0.98	0.84	0.92	0.92	0.85	0.74	0.94
Finland	1.00	0.99	0.92	0.87	0.80	0.91	0.96
France	0.74	0.99	0.74	0.72	0.63	0.64	0.99
Germany	0.72	1.00	0.83	0.86	0.93	0.68	0.99
Greece	0.61	0.99	0.64	0.84	0.41	0.55	0.97
Hungary	0.51	0.90	0.83	0.95	0.76	0.64	0.97
Iceland	0.96	1.00	0.64	0.57	0.95	0.85	1.00
Ireland	0.90	0.99	0.81	0.83	0.71	0.68	0.99
Israel	0.38	1.00	0.87	0.48	0.68	0.75	0.95
Italy	0.55	0.98	0.40	0.88	0.46	0.62	0.99
Japan	0.64	0.82	0.98	0.00	0.93	0.70	1.00
Korea	0.05	0.93	0.89	0.81	0.66	0.79	0.98
Latvia	0.69	0.69	0.89	0.94	0.80	0.60	0.87
Lithuania	0.78	0.67	0.98	0.97	0.80	0.42	0.91
Luxembourg	0.80	1.00	0.60	0.90	0.68	0.89	1.00
Mexico	0.36	0.28	0.00	0.00	0.49	0.04	0.00
Netherlands	0.71	1.00	0.74	0.99	0.95	0.81	0.98
New Zealand	0.98	0.97	0.74	0.48	0.93	0.49	0.96
Norway	0.95	1.00	0.75	0.95	0.88	1.00	0.98
Poland	0.25	0.94	0.96	0.85	0.73	0.58	0.99
Portugal	0.88	0.97	0.25	0.80	0.73	0.81	0.98
Russia	0.73	0.62	1.00	1.00	0.76	0.45	0.83
Slovak Republic	0.43	0.96	0.94	0.85	0.71	0.68	0.98
Slovenia	0.50	0.99	0.91	0.80	0.78	0.96	0.99
South Africa	0.00	0.00	0.11	0.43	0.00	0.00	0.49

Spain	0.80	0.99	0.40	0.91	0.56	0.75	0.98
Sweden	0.99	1.00	0.79	0.97	0.88	0.74	0.97
Switzerland	0.80	1.00	0.89	0.99	1.00	0.87	1.00
Türkiye	0.06	0.86	0.00	0.07	0.22	0.36	0.97
United Kingdom	0.80	0.99	0.75	0.60	0.88	0.72	1.00
United States	0.90	1.00	0.94	0.62	0.68	0.72	0.78

Let us merge the dataset with the computed overall features and the original dataset.

In [23]:
```
Rating_fin = Rating.merge(Rating_normalized_imputed, left_index=True, right_index=True)
Rating_fin = np.around(Rating_fin, decimals=2)
```

Than we will calculate the overall country ratings based on the overall features and add a corresponding column to the dataset. We will display the resulting dataset on the screen and analyze the obtained results.

In [24]:
```
Rating_fin['Total_Score'] = Rating_fin[['Housing', 'Income', 'Jobs', 'Community', 'Educa
                                   'Health', 'Life satisfaction_ov', 'Safety', 'Work-life balanc
Rating_fin.sort_values('Total_Score', ascending=False)
```

Out[24]:

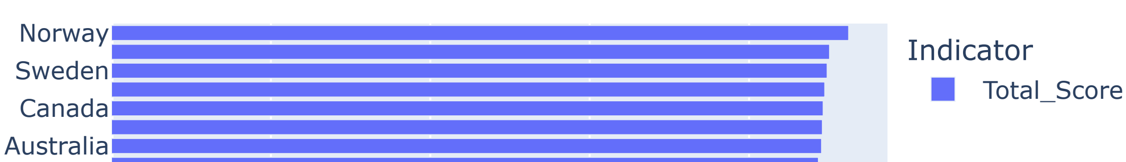
Indicator Country	Air pollution	Dwellings without basic facilities	Educational attainment	Employees working very long hours	Employment rate	Feeling safe walking alone at night	Homicide rate	Household net adjusted disposable income	Househ net we
Norway	6.7	0.0	82.00	1.4	75.0	93.0	0.6	39144.00	26835
Iceland	6.4	0.0	76.00	11.7	78.0	85.0	0.3	41825.30	51987
Sweden	5.8	0.0	84.00	0.9	75.0	79.0	1.1	33730.00	37429
Netherlands	12.2	0.1	81.00	0.3	78.0	83.0	0.6	34984.00	24859
Canada	7.1	0.2	92.00	3.3	70.0	78.0	1.2	34421.00	47824
United States	7.7	0.1	92.00	10.4	67.0	78.0	6.0	51147.00	68450
Australia	6.7	0.0	84.00	12.5	73.0	67.0	0.9	37433.00	52876
Finland	5.5	0.4	91.00	3.6	72.0	88.0	1.2	33471.00	23003
Switzerland	10.1	0.0	89.00	0.4	80.0	86.0	0.3	39697.00	47615
Luxembourg	10.0	0.1	74.00	2.8	67.0	87.0	0.2	44773.00	94116
Denmark	10.0	0.5	82.00	1.1	74.0	85.0	0.5	33774.00	14986
New Zealand	6.0	1.0	81.00	14.0	77.0	66.0	1.3	39024.00	51416
Germany	12.0	0.1	86.00	3.9	77.0	76.0	0.4	38971.00	30431
Ireland	7.8	0.2	85.00	4.7	68.0	76.0	0.5	29488.00	37034
United Kingdom	10.1	0.5	82.00	10.8	75.0	78.0	0.2	33049.00	52442
Belgium	12.8	0.7	80.00	4.3	65.0	56.0	1.1	34884.00	44760

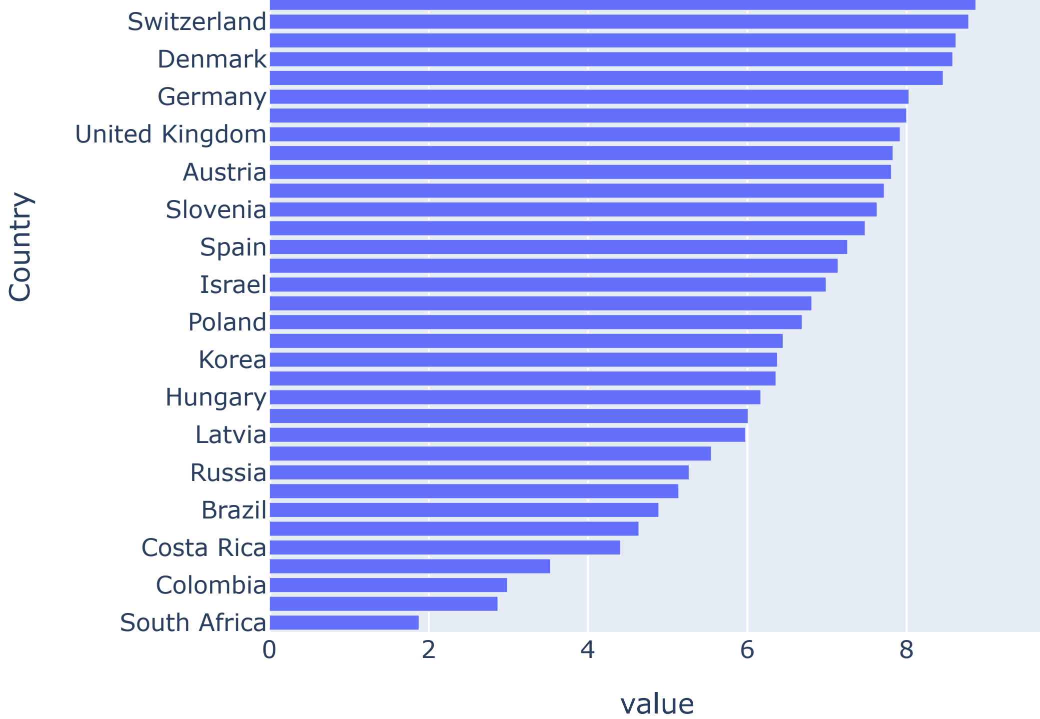
Austria	12.2	0.8	86.00	5.3	72.0	86.0	0.5	37001.00	30963
Estonia	5.9	5.7	91.00	2.2	74.0	79.0	1.9	23784.00	18862
Slovenia	17.0	0.2	90.00	5.6	71.0	91.0	0.4	25250.00	23328
France	11.4	0.5	81.00	7.7	65.0	74.0	0.4	34375.00	29863
Spain	10.0	0.3	63.00	2.5	62.0	80.0	0.7	27155.00	36653
Czech Republic	17.0	0.5	94.00	4.5	74.0	77.0	0.7	26664.00	24867
Israel	19.7	0.0	88.00	14.1	67.0	80.0	1.5	28160.73	27544
Italy	15.9	0.6	63.00	3.3	58.0	73.0	0.5	29431.00	29502
Poland	22.8	2.3	93.00	4.2	69.0	71.0	0.5	23675.00	23322
Slovak Republic	18.5	1.5	92.00	4.2	68.0	76.0	0.8	21149.00	17142
Korea	27.3	2.5	89.00	5.3	66.0	82.0	0.8	24590.00	36234
Lithuania	10.5	11.8	94.00	1.0	72.0	62.0	2.5	26976.00	18203
Hungary	16.7	3.5	86.00	1.5	70.0	74.0	0.9	21026.00	15029
Portugal	8.3	0.9	55.00	5.6	69.0	83.0	0.7	24877.00	25530
Latvia	12.7	11.2	89.00	1.6	72.0	72.0	3.7	19783.00	7924
Japan	13.7	6.4	93.91	27.0	77.0	77.0	0.2	28872.00	29473
Russia	11.8	13.8	95.00	0.1	70.0	64.0	4.8	19546.00	12134
Greece	14.5	0.4	76.00	4.5	56.0	69.0	1.0	20791.00	14832
Brazil	11.7	6.7	57.00	5.6	57.0	45.0	19.0	23654.64	19484
Chile	23.4	9.4	67.00	7.7	56.0	41.0	2.4	22012.49	13578
Costa Rica	17.5	2.3	43.00	22.0	55.0	47.0	10.0	16517.00	6716
Türkiye	27.1	4.9	42.00	25.0	48.0	59.0	1.0	10844.89	6273
Colombia	22.6	12.3	59.00	23.7	58.0	50.0	23.1	16586.78	6841
Mexico	20.3	25.9	42.00	27.0	59.0	42.0	26.8	16269.00	6273
South Africa	28.5	35.9	48.00	15.4	39.0	40.0	13.7	9338.00	6273

Data Visualization and Analysis

We can create a bar chart to visually evaluate the country ratings. Norway tops the rating, along with several other Scandinavian countries, while North American countries feature prominently. At the bottom of the rating, we observe South Africa and countries in South America.

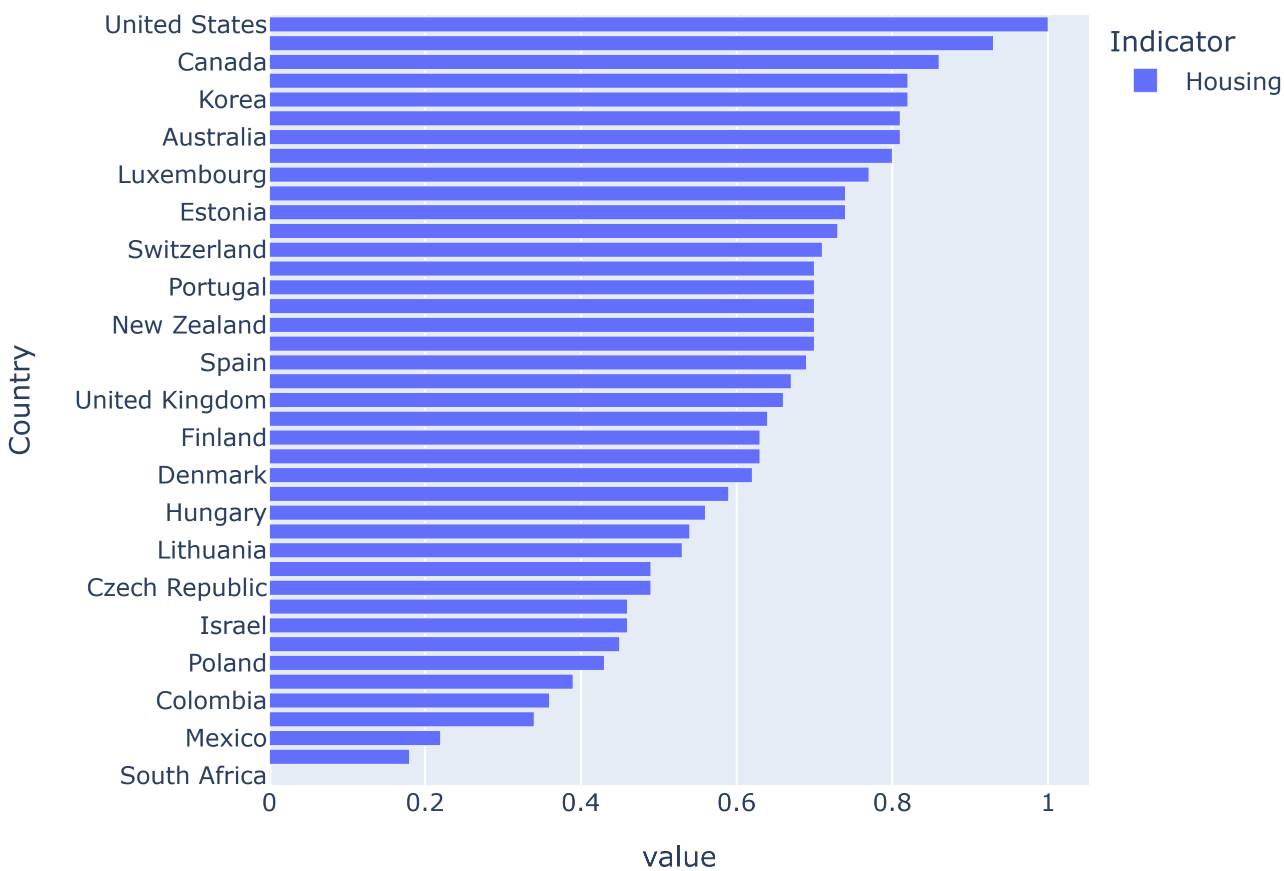
In [25]:
```
px.bar(Rating_fin[['Total_Score']].sort_values('Total_Score'), orientation='h')
```

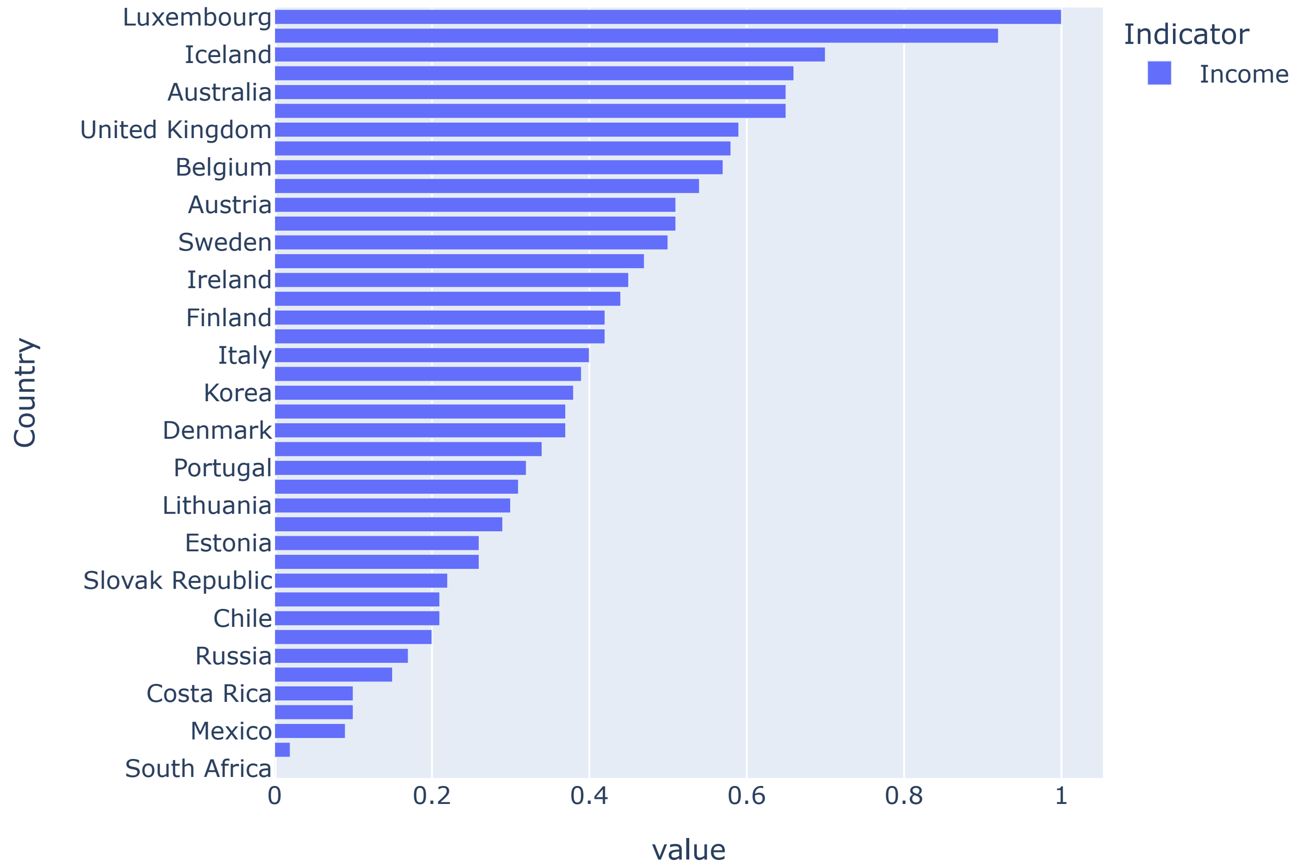


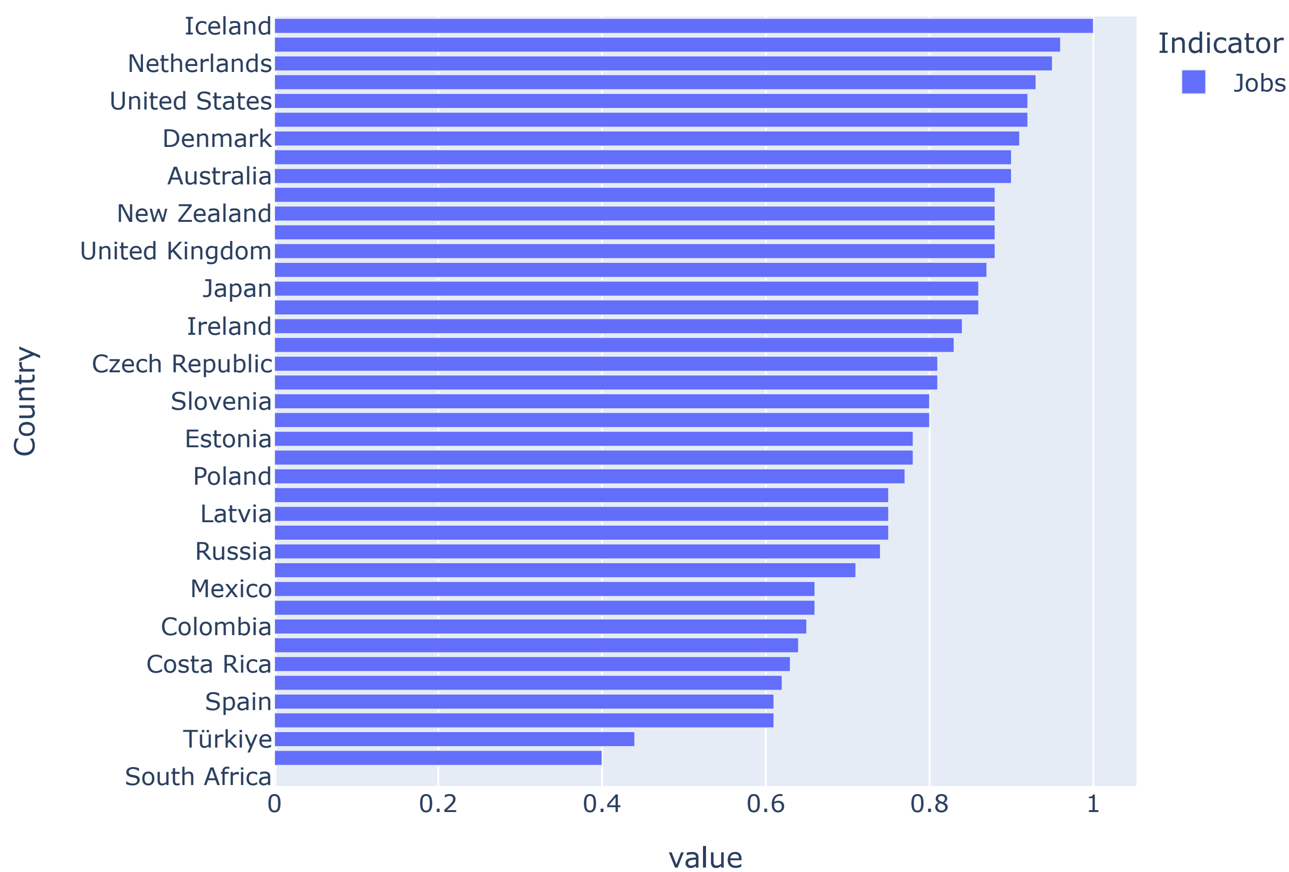


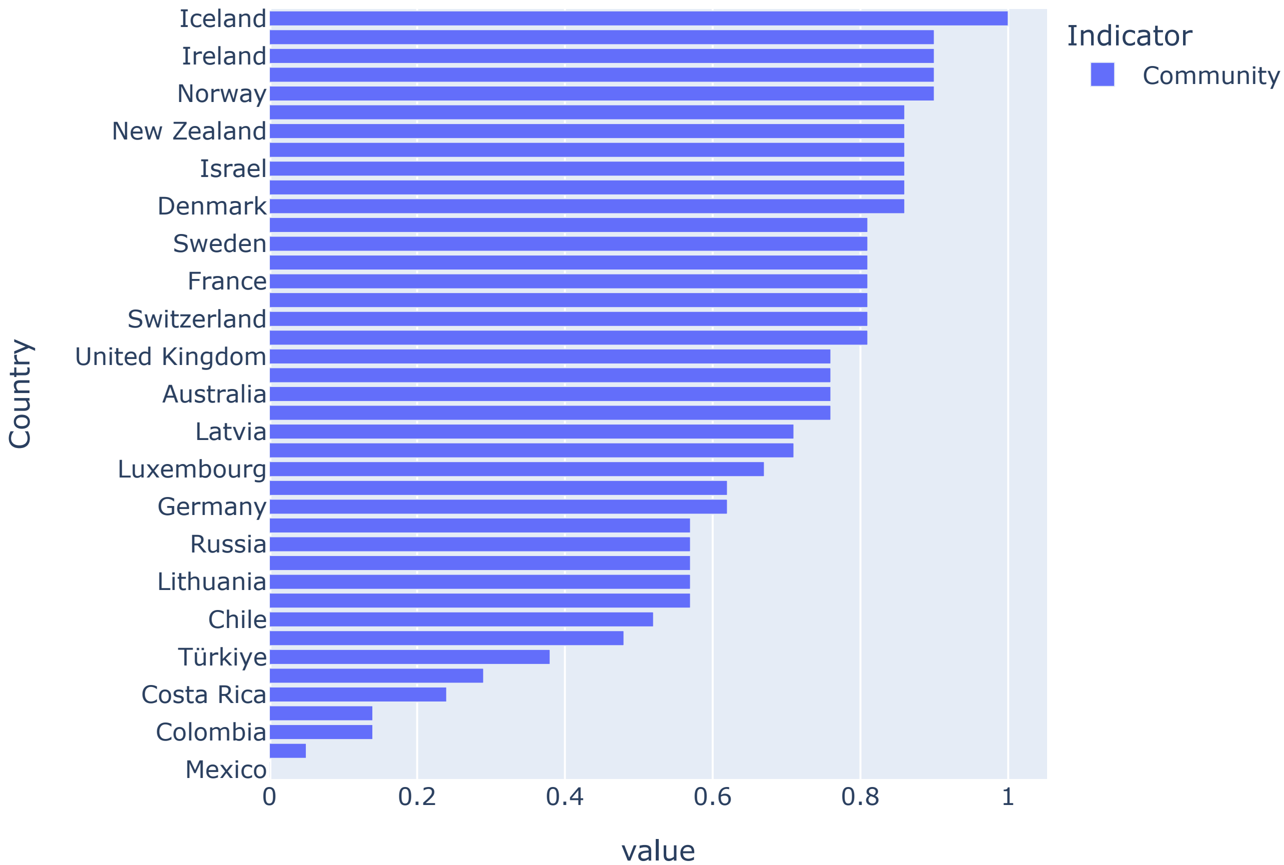
Similar charts can be created to explore all the overall features.

In [26]:
```
Overal_features = ['Housing', 'Income', 'Jobs', 'Community', 'Education', 'Environment',
                   'Health', 'Life satisfaction_ov', 'Safety', 'Work-life balanc
for i in Overal_features:
    fig_bar = px.bar(Rating_fin[[i]].sort_values(i), orientation='h')
    fig_bar.show()
```

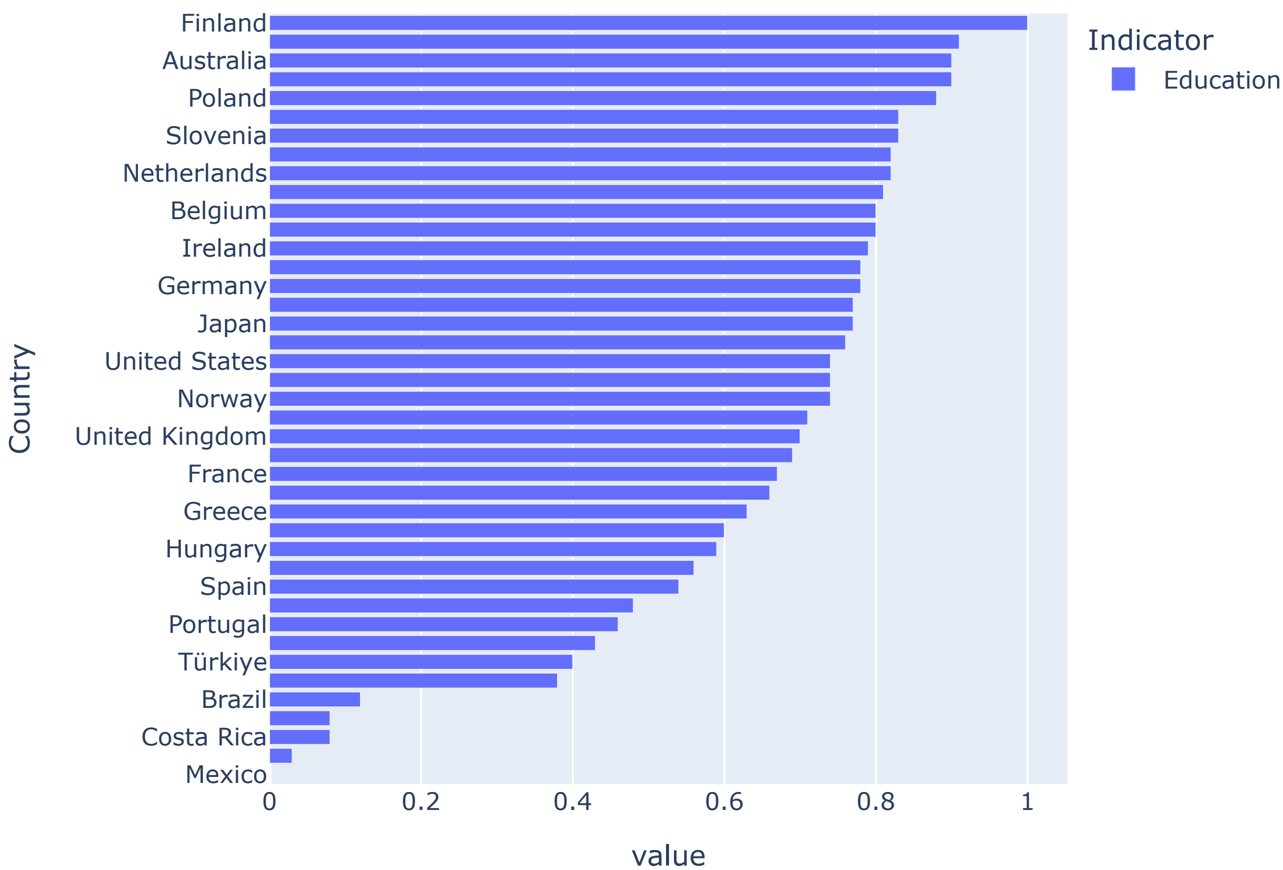



Indicator
Income
Luxembourg
Iceland
Australia
United Kingdom
Belgium
Austria
Sweden
Ireland
Finland
Italy
Korea
Denmark
Portugal
Lithuania
Estonia
Slovak Republic
Chile
Russia
Costa Rica
Mexico
South Africa
Country
0
0.2
0.4
0.6
0.8
1
value

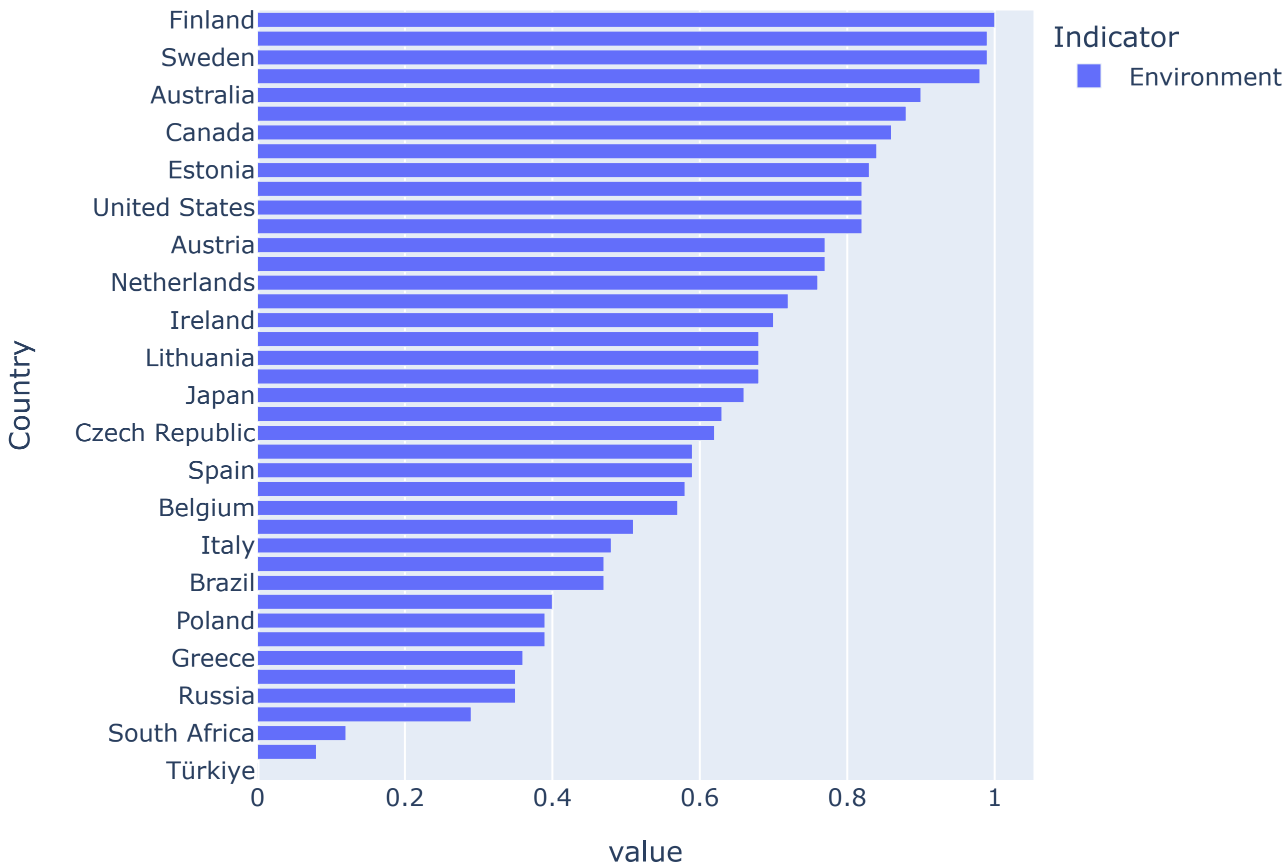

Indicator
Jobs
Iceland
Netherlands
United States
Denmark
Australia
New Zealand
United Kingdom
Japan
Ireland
Czech Republic
Slovenia
Estonia
Poland
Latvia
Russia
Mexico
Colombia
Costa Rica
Spain
Türkiye
South Africa
Country
0
0.2
0.4
0.6
0.8
1
value

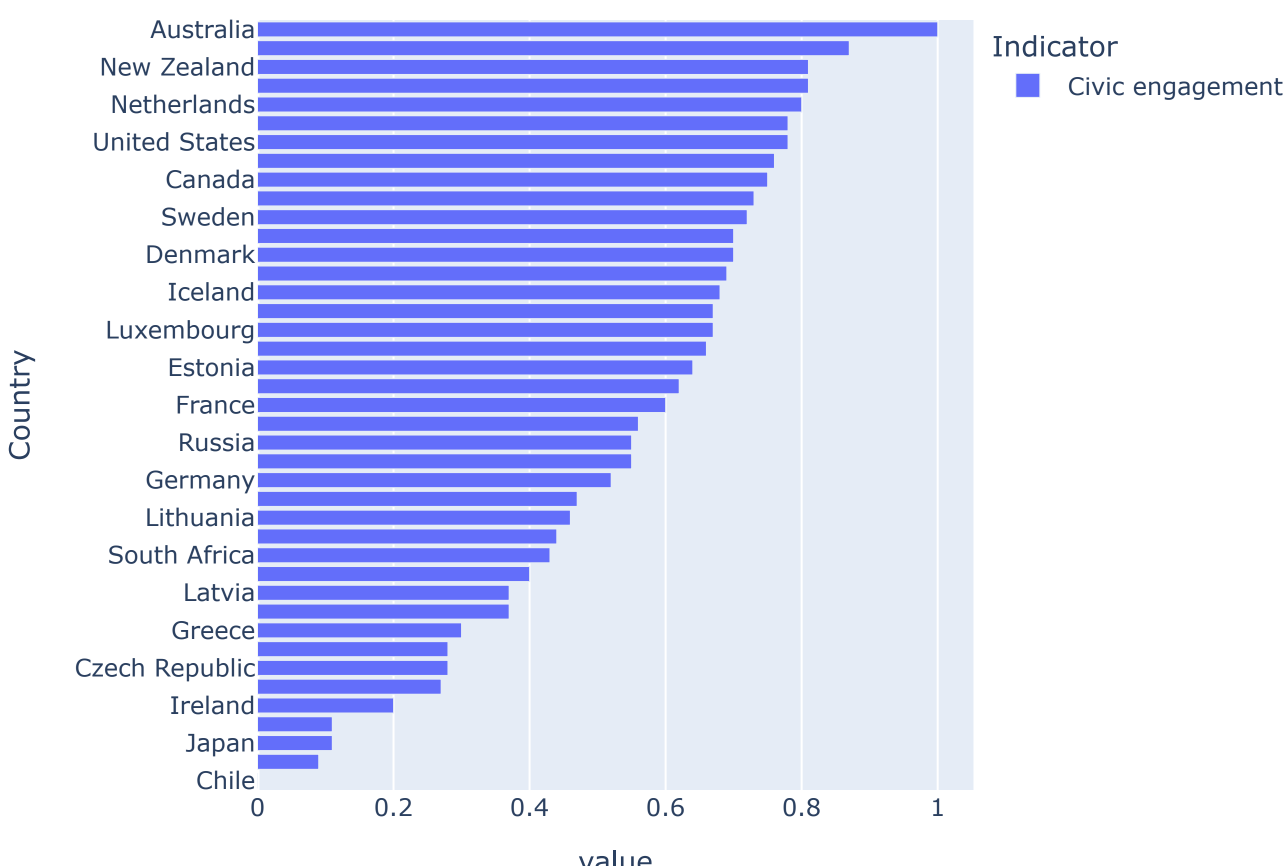


Iceland
Ireland
Norway
New Zealand
Israel
Denmark
Sweden
France
Switzerland
United Kingdom
Australia
Latvia
Luxembourg
Germany
Russia
Lithuania
Chile
Türkiye
Costa Rica
Colombia
Mexico
Country
0
0.2
0.4
0.6
0.8
1
value
Indicator
Community

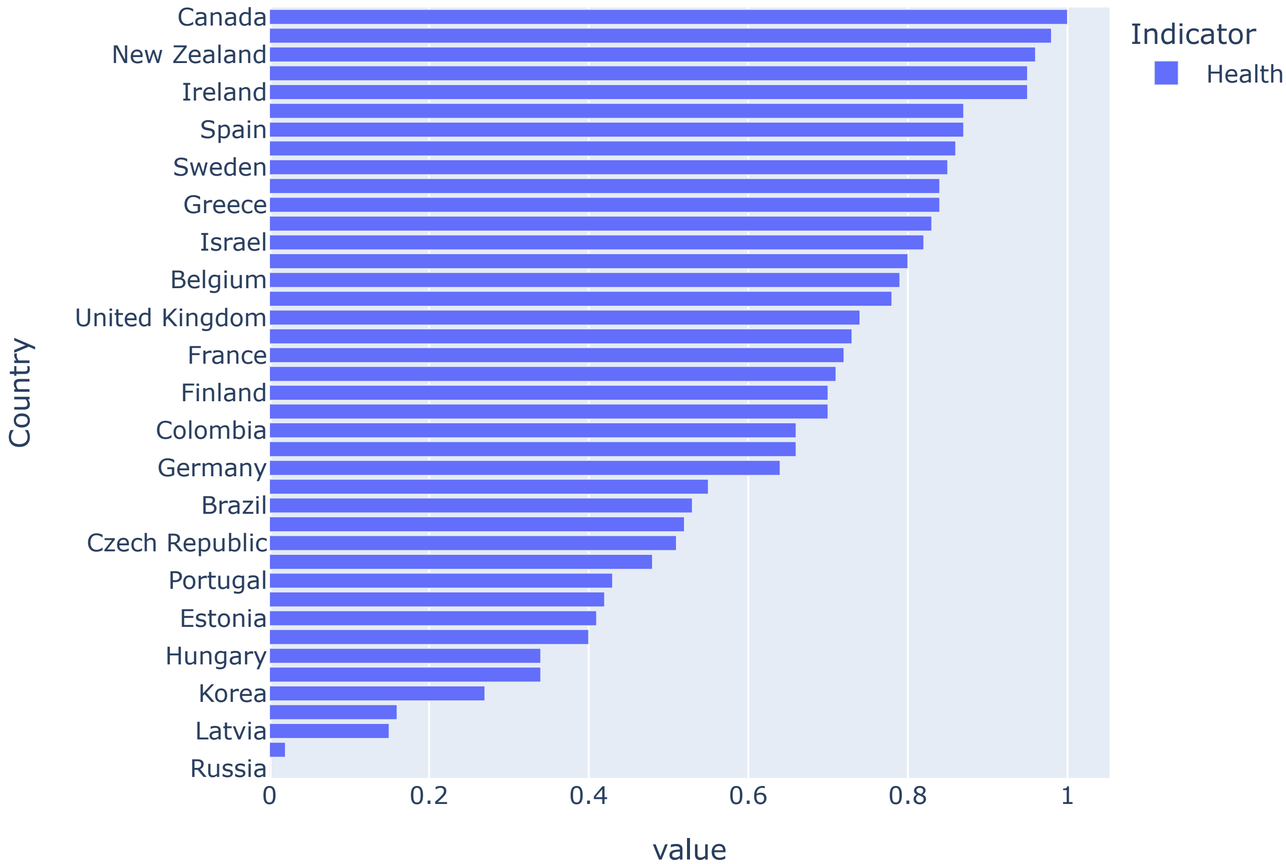


Finland
Australia
Poland
Slovenia
Netherlands
Belgium
Ireland
Germany
Japan
United States
Norway
United Kingdom
France
Greece
Hungary
Spain
Portugal
Türkiye
Brazil
Costa Rica
Mexico
Country
0
0.2
0.4
0.6
0.8
1
value
Indicator
Education

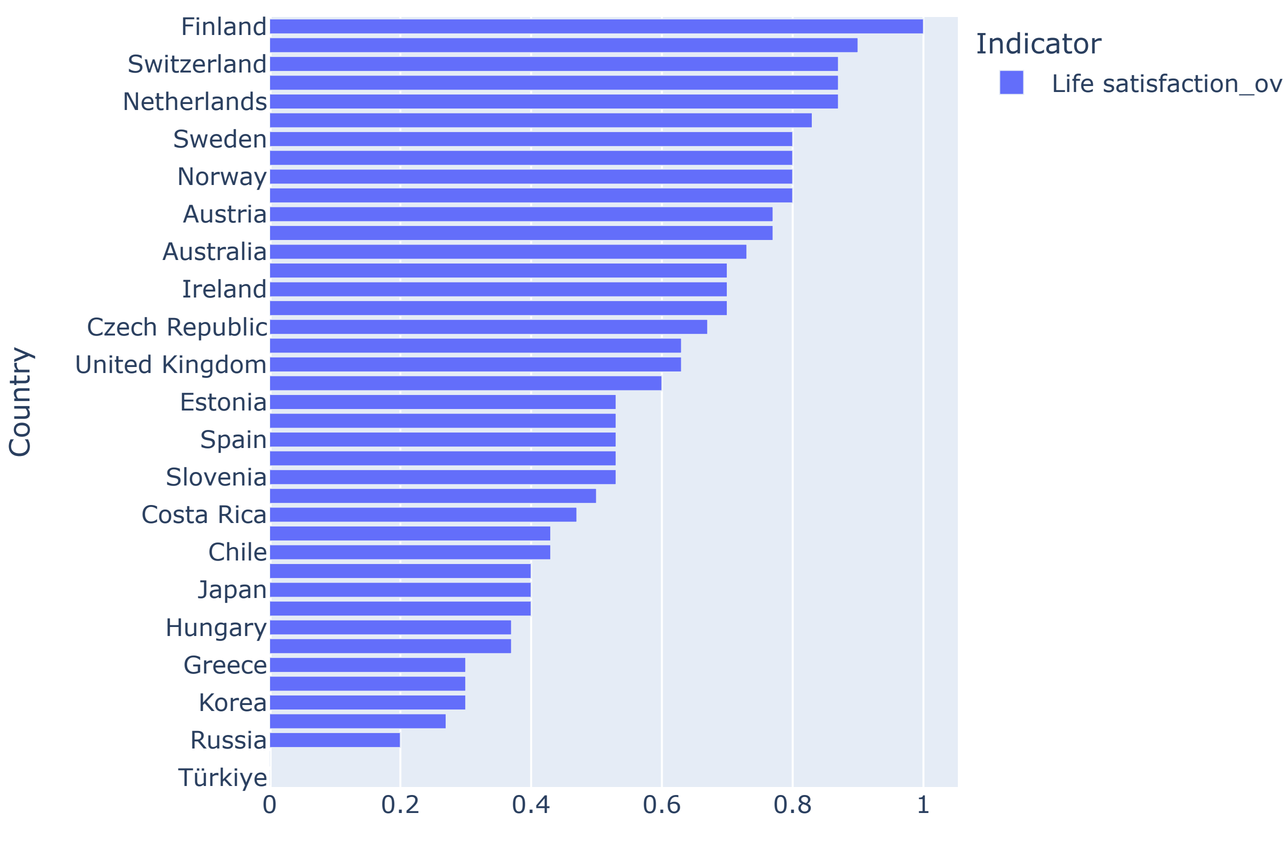

Finland
Sweden
Australia
Canada
Estonia
United States
Austria
Netherlands
Ireland
Lithuania
Japan
Czech Republic
Spain
Belgium
Italy
Brazil
Poland
Greece
Russia
South Africa
Türkiye
Country
0
0.2
0.4
0.6
0.8
1
value
Indicator
Environment


Australia
New Zealand
Netherlands
United States
Canada
Sweden
Denmark
Iceland
Luxembourg
Estonia
France
Russia
Germany
Lithuania
South Africa
Latvia
Greece
Czech Republic
Ireland
Japan
Chile
Country
0
0.2
0.4
0.6
0.8
1
value
Indicator
Civic engagement

value

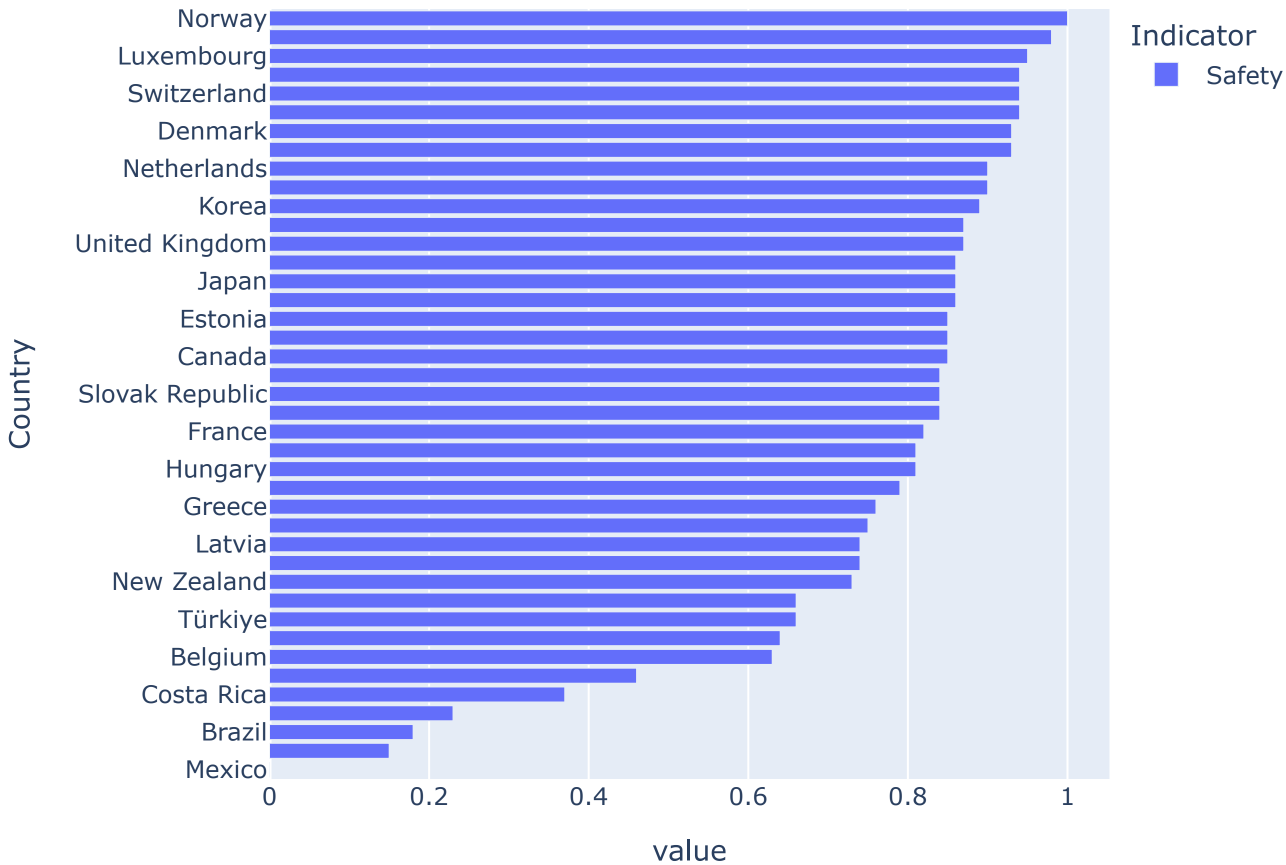


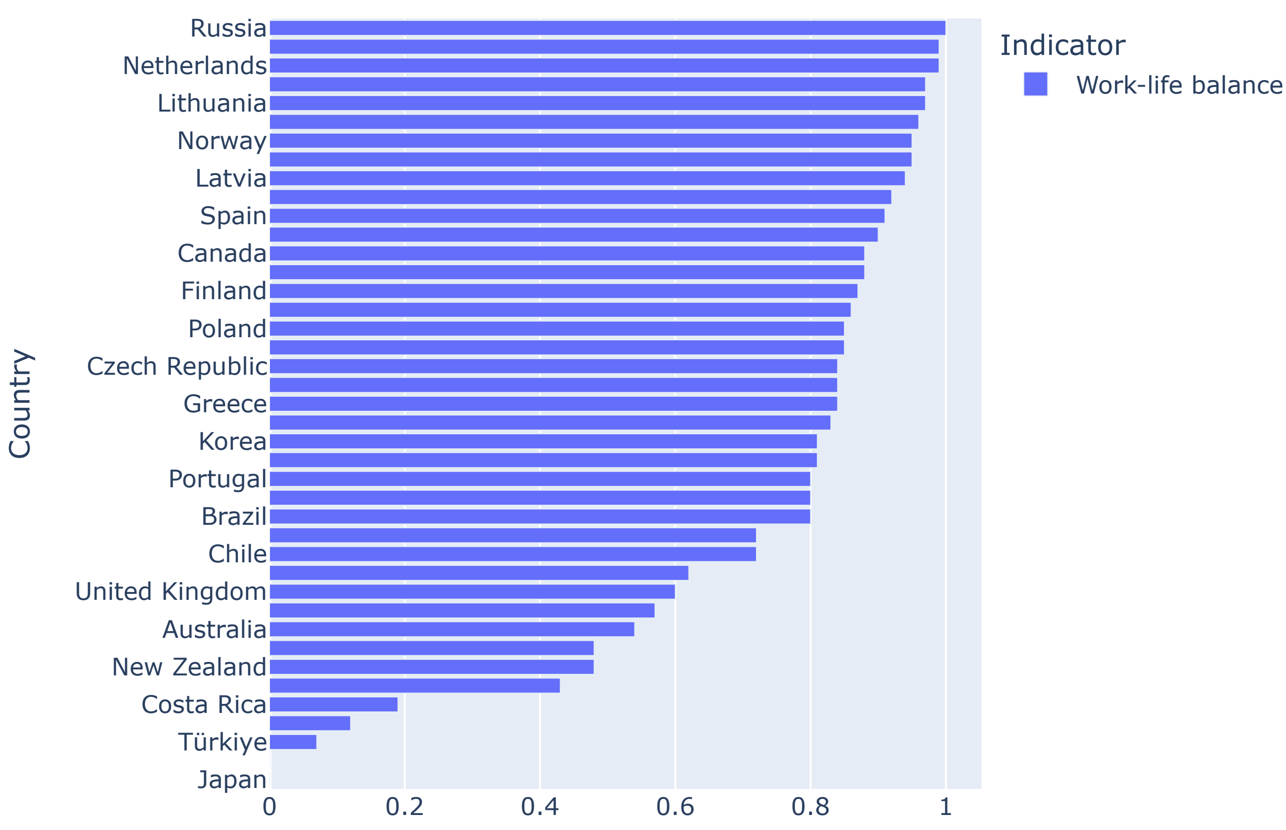
Canada
New Zealand
Ireland
Spain
Sweden
Greece
Israel
Belgium
United Kingdom
France
Finland
Colombia
Germany
Brazil
Czech Republic
Portugal
Estonia
Hungary
Korea
Latvia
Russia
Country
0
0.2
0.4
0.6
0.8
1
value
Indicator
Health



Finland
Switzerland
Netherlands
Sweden
Norway
Austria
Australia
Ireland
Czech Republic
United Kingdom
Estonia
Spain
Slovenia
Costa Rica
Chile
Japan
Hungary
Greece
Korea
Russia
Türkiye
Country
0
0.2
0.4
0.6
0.8
1
Indicator
Life satisfaction_ov

value

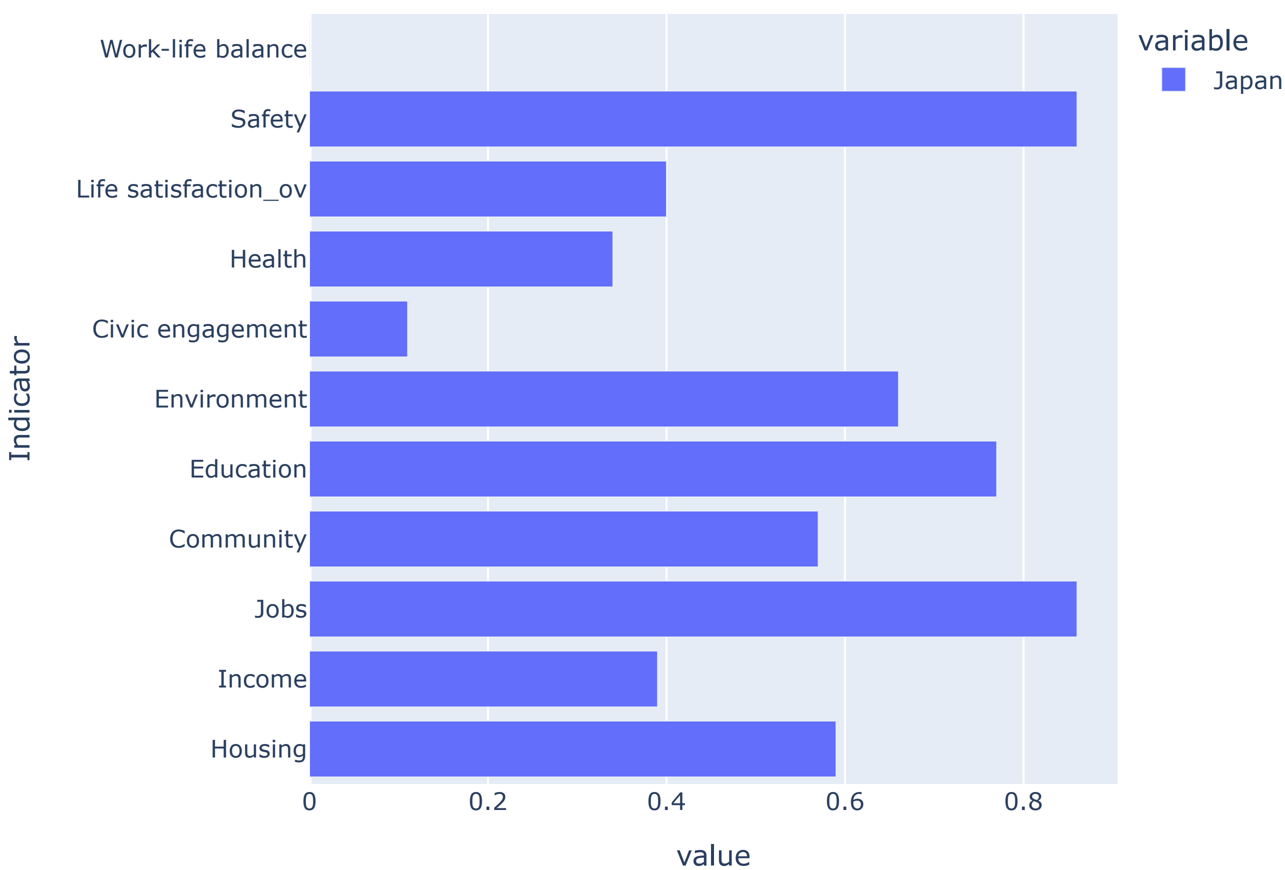

Norway
Luxembourg
Switzerland
Denmark
Netherlands
Korea
United Kingdom
Japan
Estonia
Canada
Slovak Republic
France
Hungary
Greece
Latvia
New Zealand
Türkiye
Belgium
Costa Rica
Brazil
Mexico
Country
0
0.2
0.4
0.6
0.8
1
value
Indicator
Safety

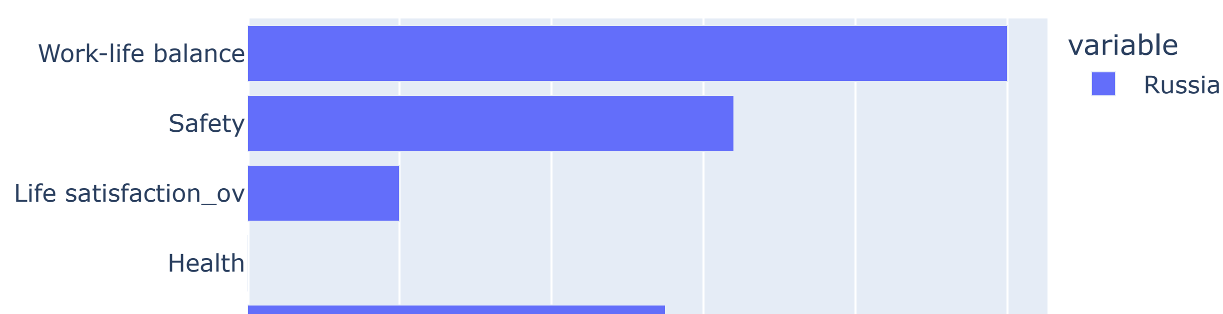

Russia
Netherlands
Lithuania
Norway
Latvia
Spain
Canada
Finland
Poland
Czech Republic
Greece
Korea
Portugal
Brazil
Chile
United Kingdom
Australia
New Zealand
Costa Rica
Türkiye
Japan
Country
0
0.2
0.4
0.6
0.8
1
Indicator
Work-life balance

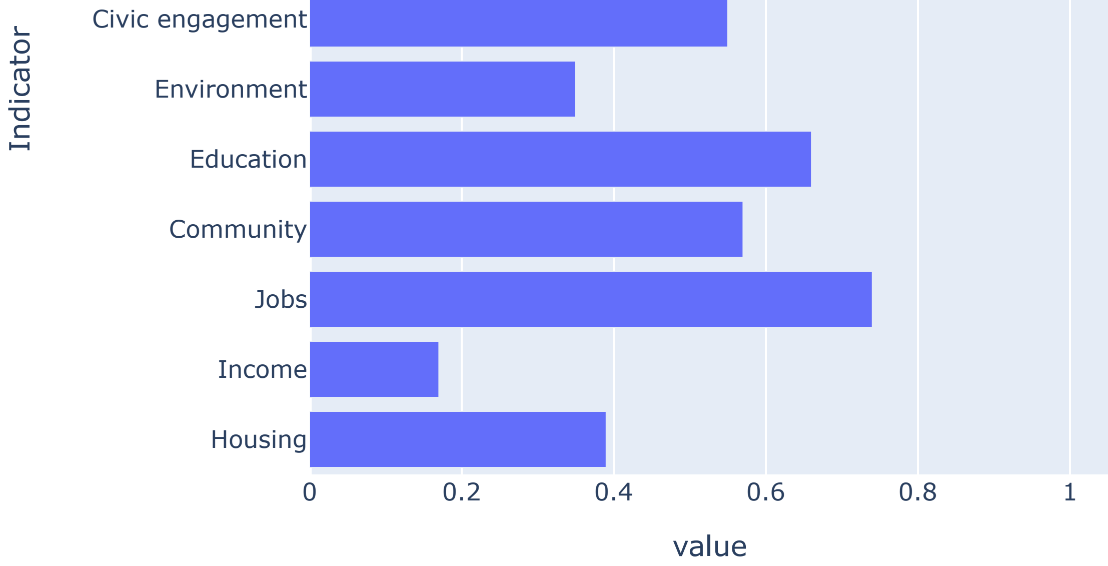
value

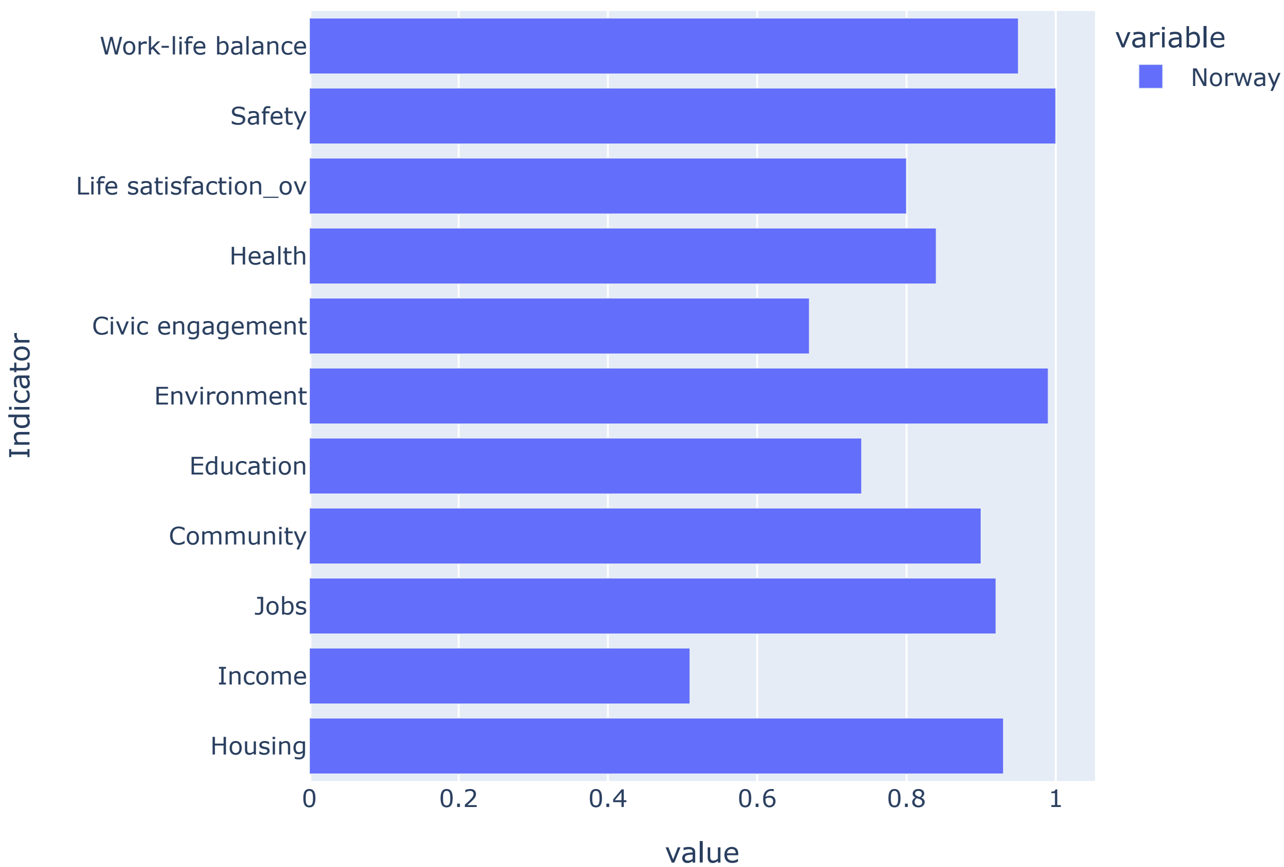
Examining specific countries, such as Japan, Russia, and Norway, it is observed that Japan ranks low due to low scores in 'Civic engagement', 'Health', 'Life satisfaction', and 'Work-life balance'. In Russia, low scores can be attributed to 'Housing', 'Environment', 'Health', and 'Life satisfaction'. Norway, as the leader of our rating, has high scores in all features, although 'Income' and 'Civic engagement' stand out slightly. We can identify areas where different countries need to work particularly hard.

In [27]:
```
fig_j = px.bar(Rating_fin.loc['Japan', Overal_features], orientation='h')
fig_r = px.bar(Rating_fin.loc['Russia', Overal_features], orientation='h')
fig_n = px.bar(Rating_fin.loc['Norway', Overal_features], orientation='h')
fig_j.show()
fig_r.show()
fig_n.show()
```



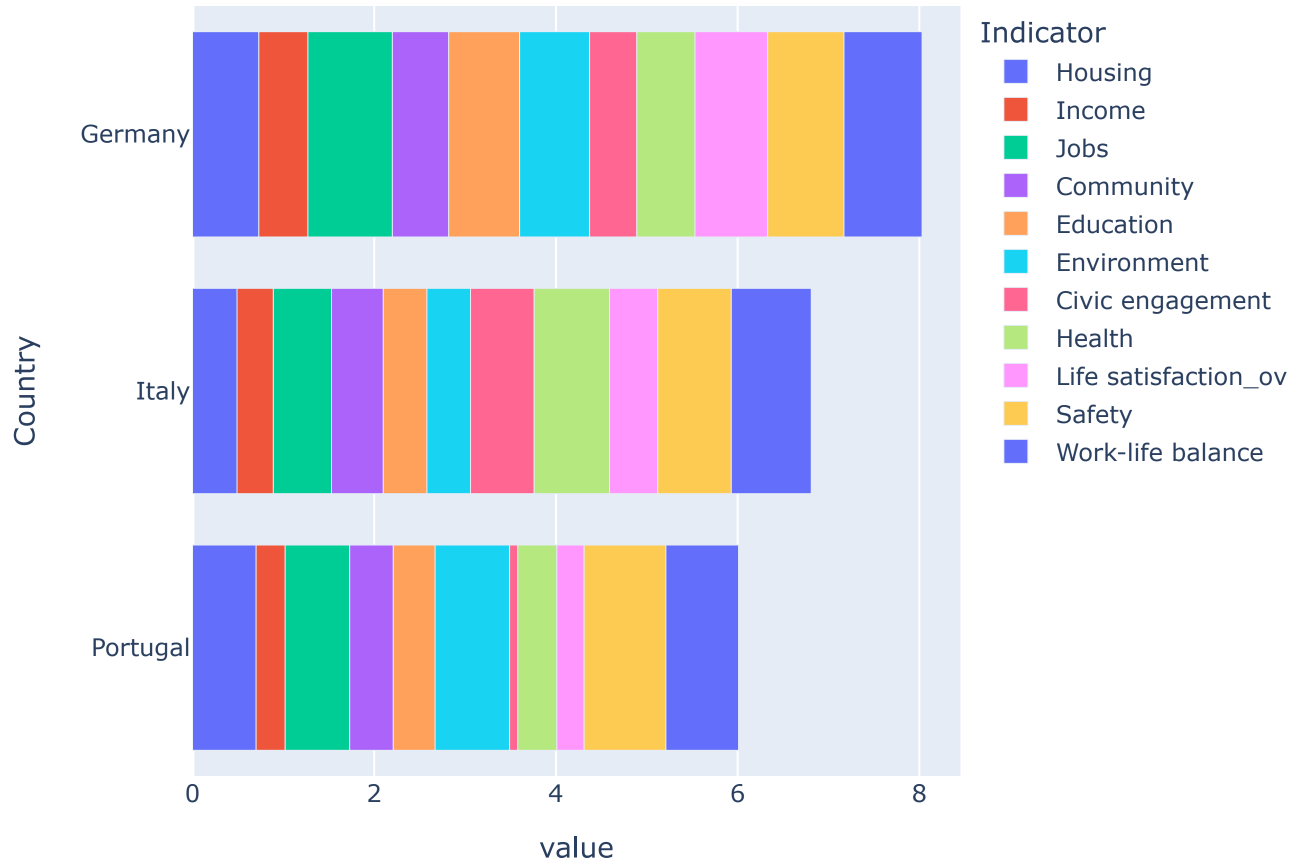






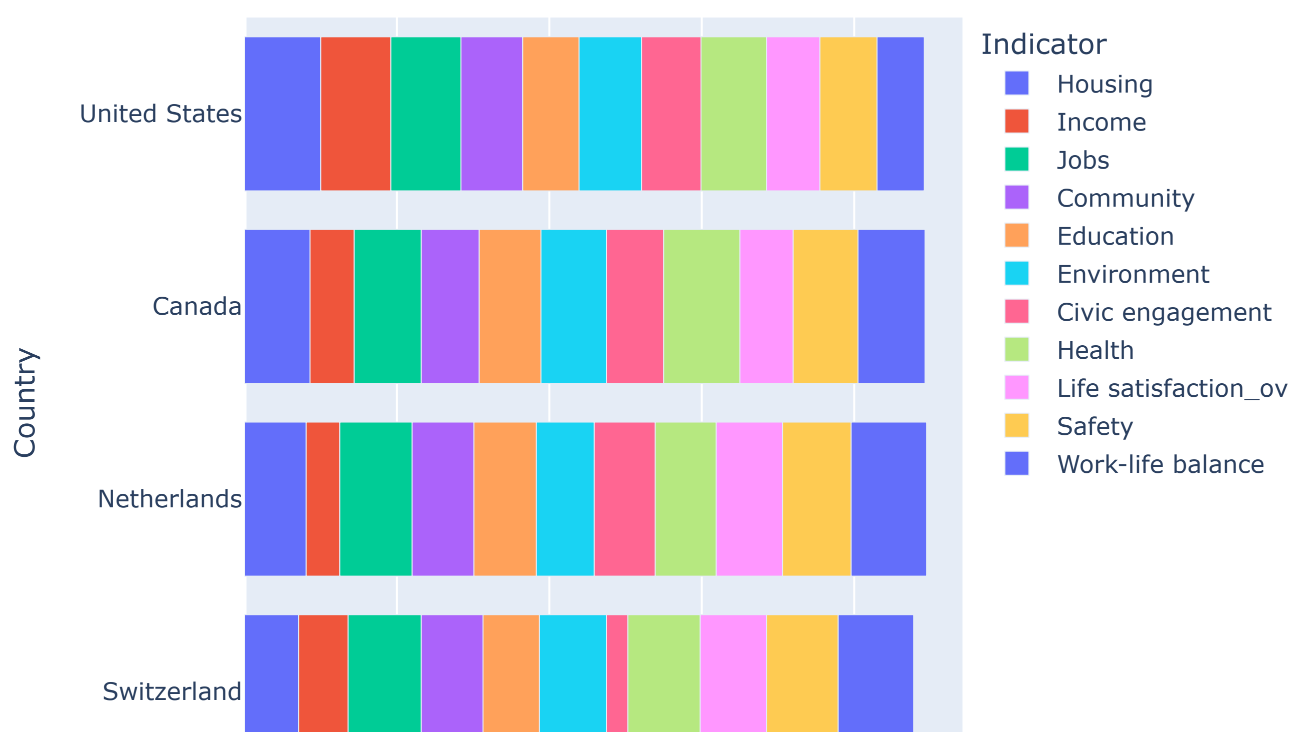
To identify specific areas for improvement, we can compare neighboring countries with moderately similar levels of development, for example, Germany, Portugal, and Italy. It can be observed that Portugal lacks in indicators such as "Community" and "Civic engagement," while Italy needs to work on such indicators as "Environment" and "Work-life balance." In comparison to these countries, Germany shows high levels across all dimensions, with the lowest scores in the categories of "Income" and "Civic engagement."

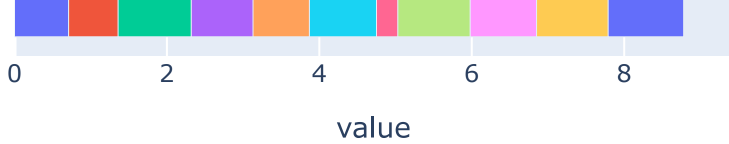
In [28]:
```
fig_eur = px.bar(Rating_fin.loc[['Portugal', 'Italy', 'Germany'], Overal_features], orie
fig_eur.show()
```



A similar analysis can be conducted among the top-ranked countries in our rating (Switzerland, Netherlands, Canada, United States). The author suggests that you draw your own conclusions based on the analysis.

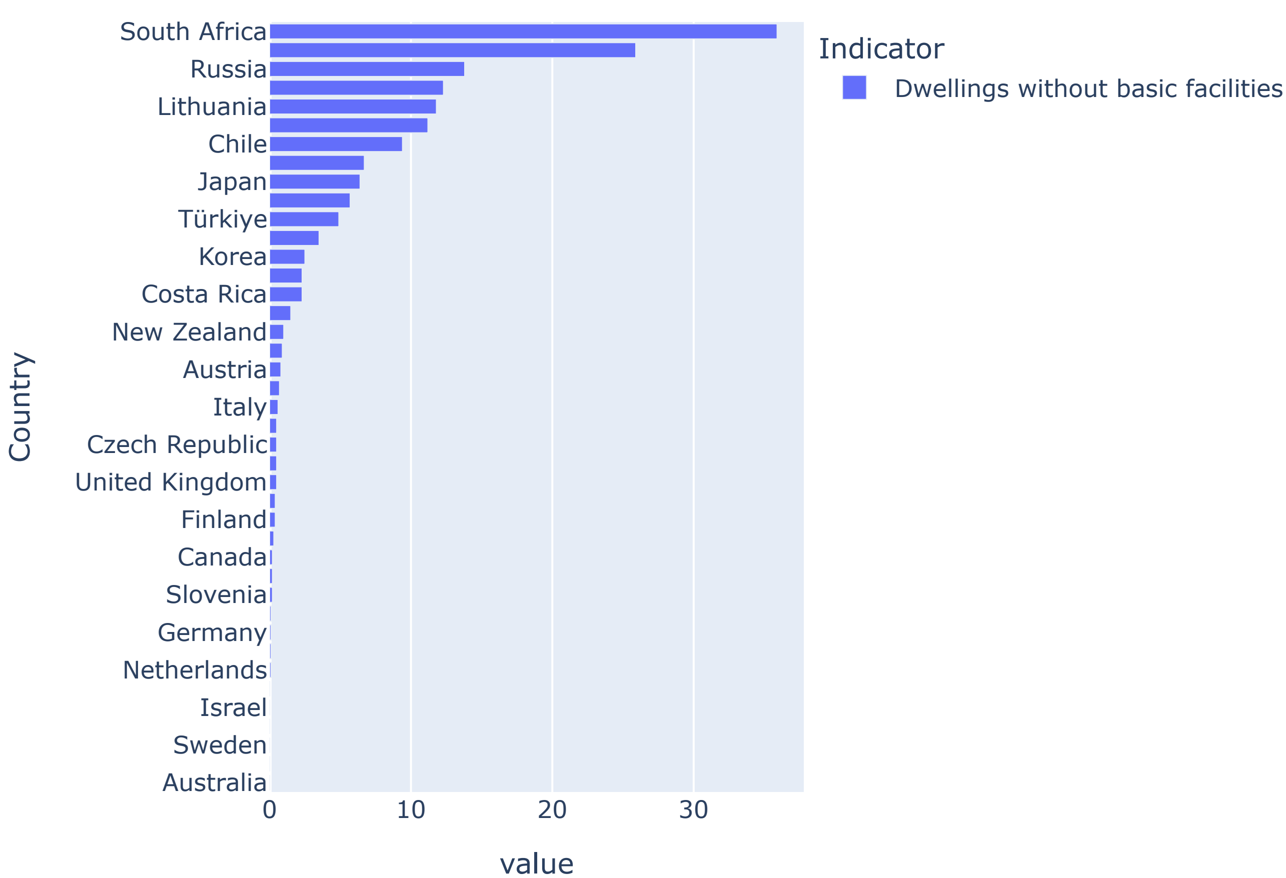
In [29]:
```
fig_top = px.bar(Rating_fin.loc[['Switzerland', 'Netherlands', 'Canada', 'United States'
fig_top.show()
```

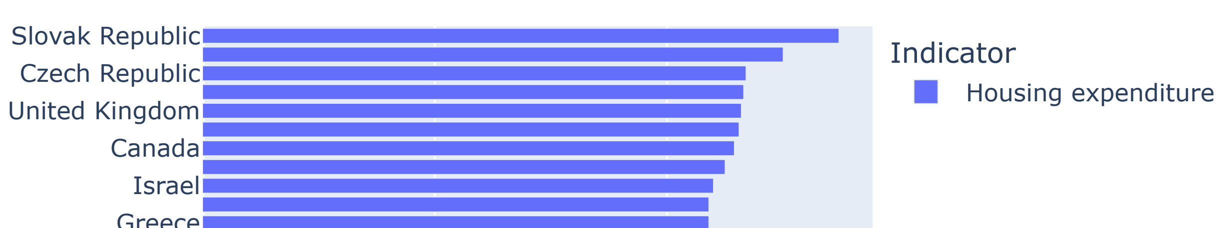


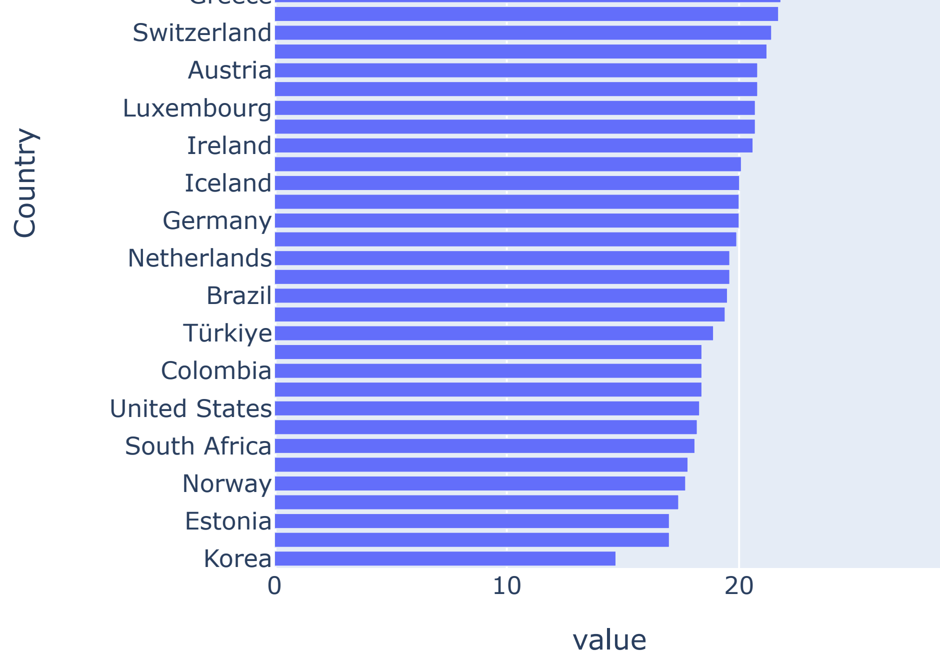


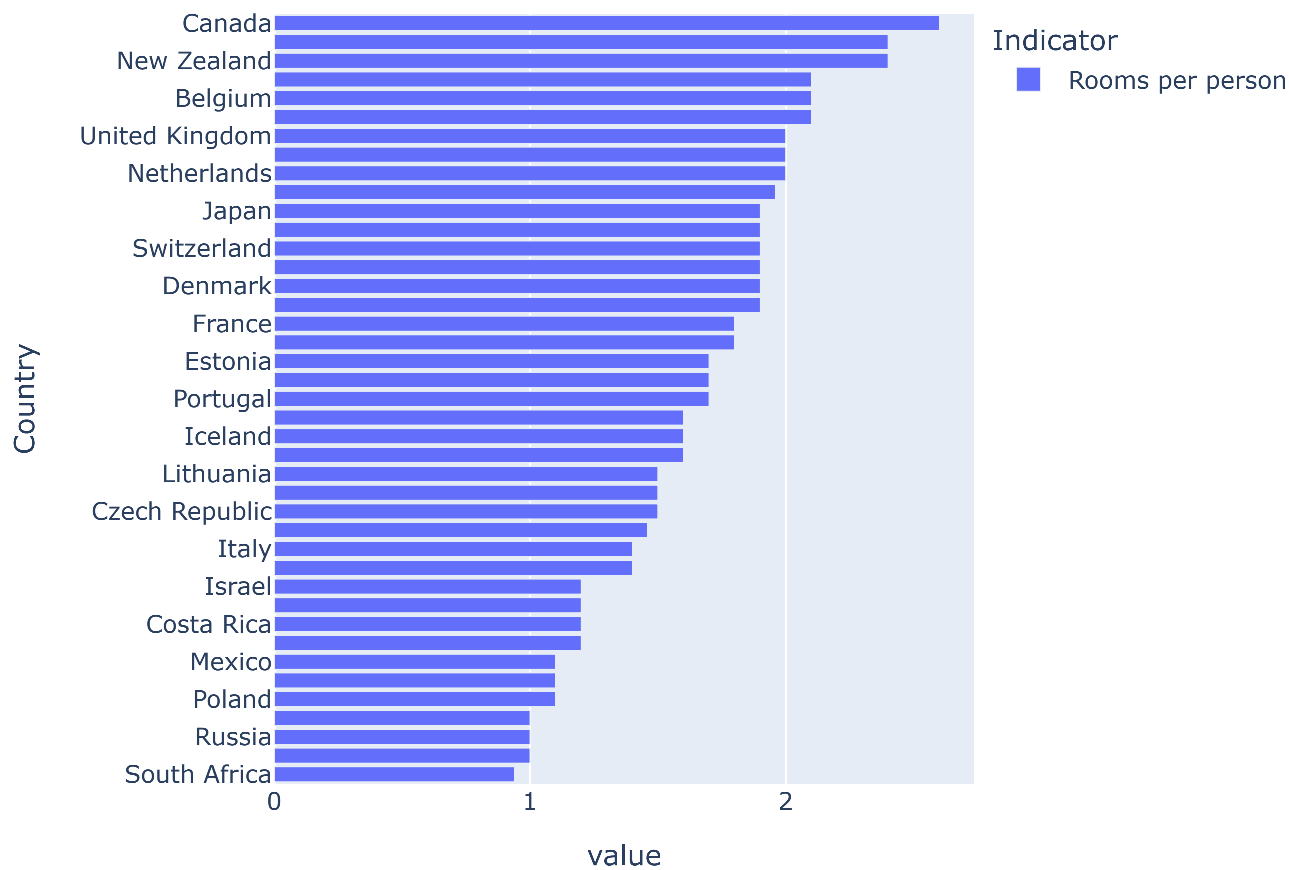
To better understand possible conclusions, we can investigate specific features by building graphs to visualize them.

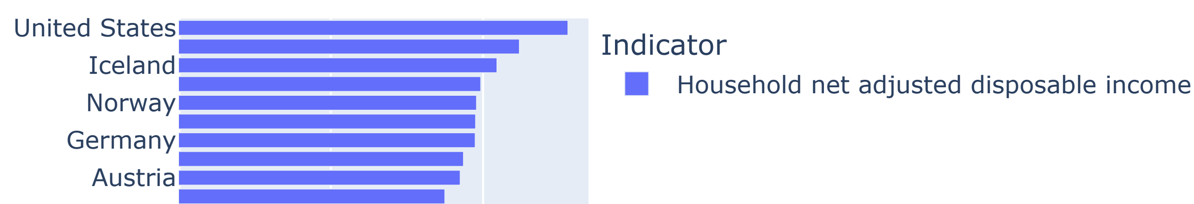
```
In [30]: Features = ['Dwellings without basic facilities', 'Housing expenditure', 'Rooms per pers
                    'Household net adjusted disposable income', 'Household net wealth', 'Labour ma
                    'Employment rate', 'Long-term unemployment rate', 'Personal earnings','Quality
                    'Educational attainment', 'Student skills', 'Years in education', 'Air polluti
                    'Stakeholder engagement for developing regulations', 'Voter turnout', 'Life ex
                    'Self-reported health', 'Life satisfaction', 'Feeling safe walking alone at ni
                    'Employees working very long hours']
         for i in Features:
             fig_bar = px.bar(Rating_fin[[i]].sort_values(i), orientation='h')
             fig_bar.show()
```

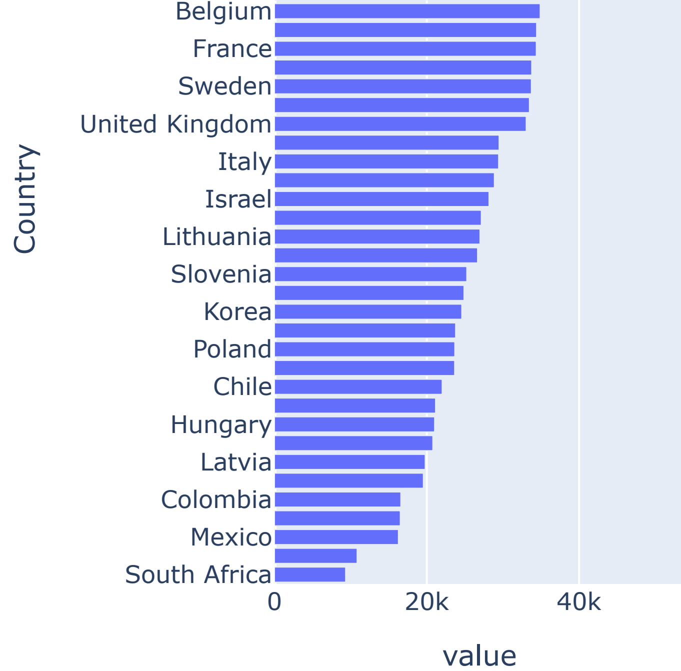


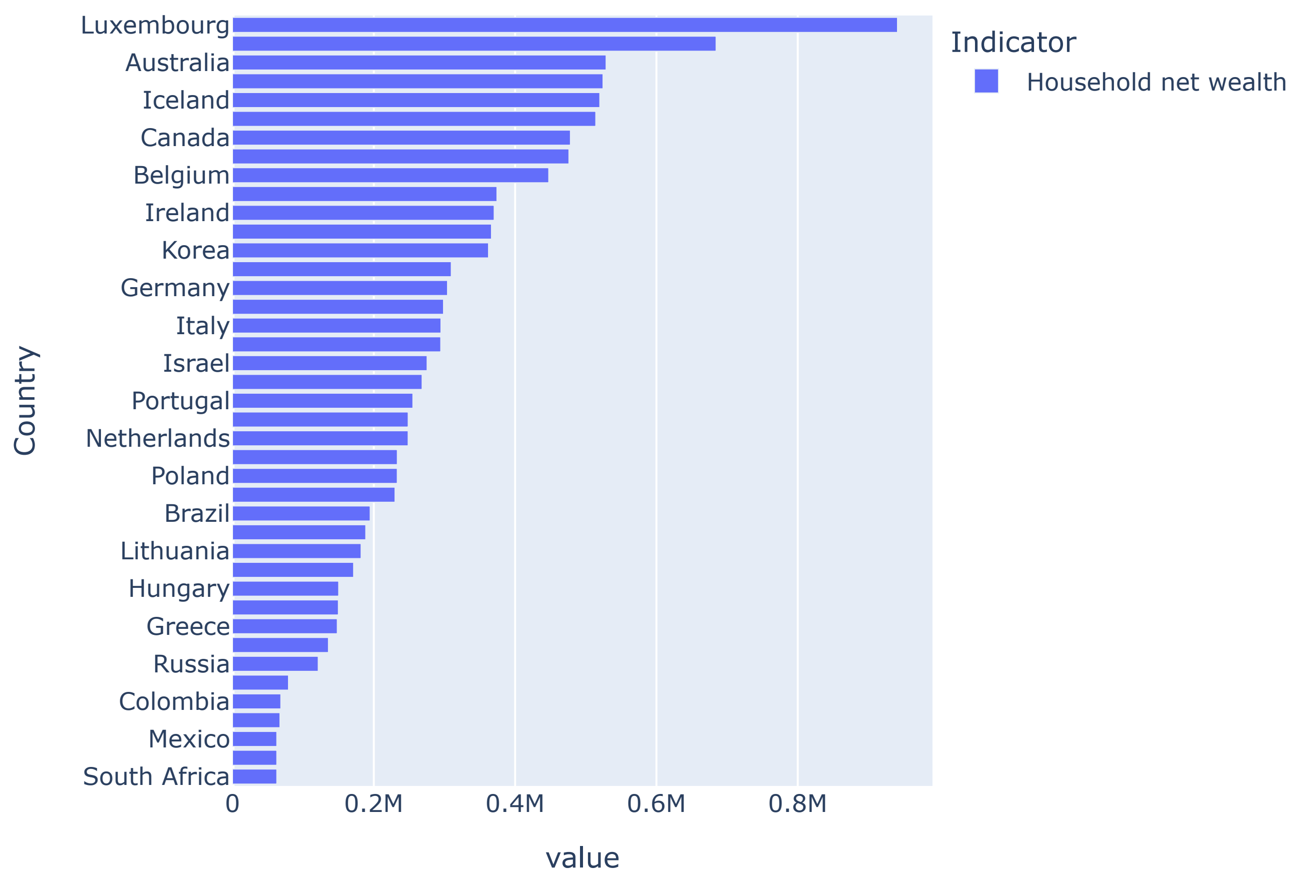


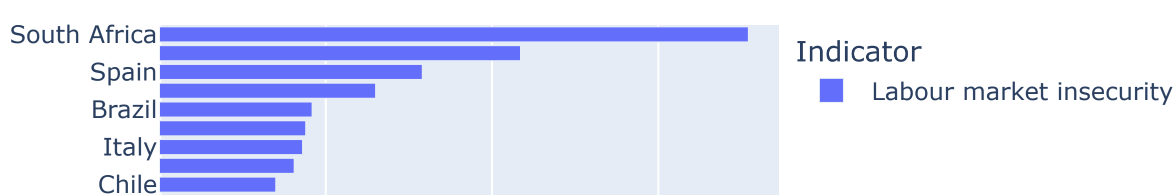

Switzerland
Austria
Luxembourg
Ireland
Iceland
Germany
Netherlands
Brazil
Türkiye
Colombia
United States
South Africa
Norway
Estonia
Korea
Country
0
10
20
value

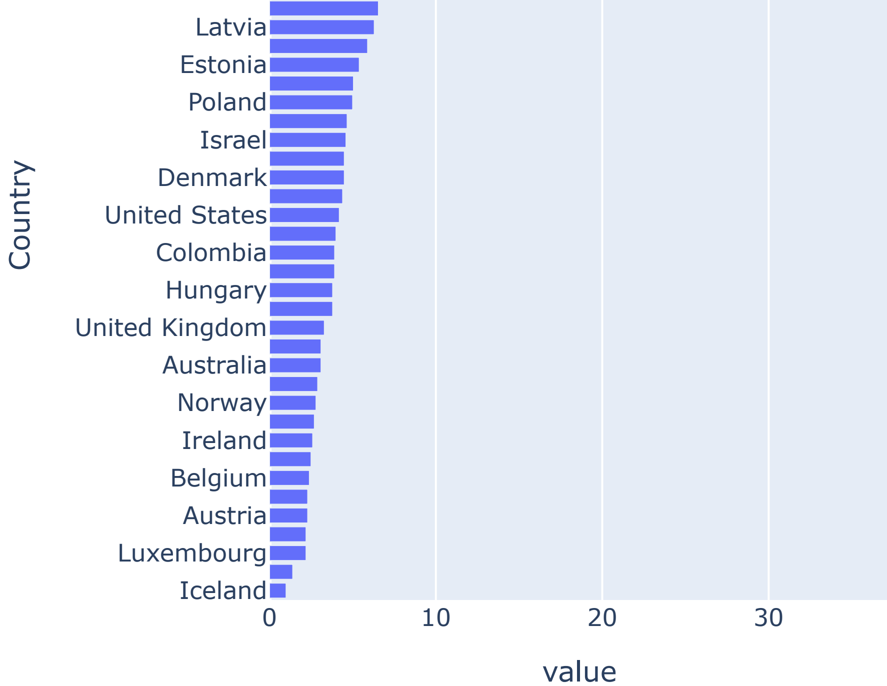

Canada
New Zealand
Belgium
United Kingdom
Netherlands
Japan
Switzerland
Denmark
France
Estonia
Portugal
Iceland
Lithuania
Czech Republic
Italy
Israel
Costa Rica
Mexico
Poland
Russia
South Africa
Country
0
1
2
value
Indicator
Rooms per person

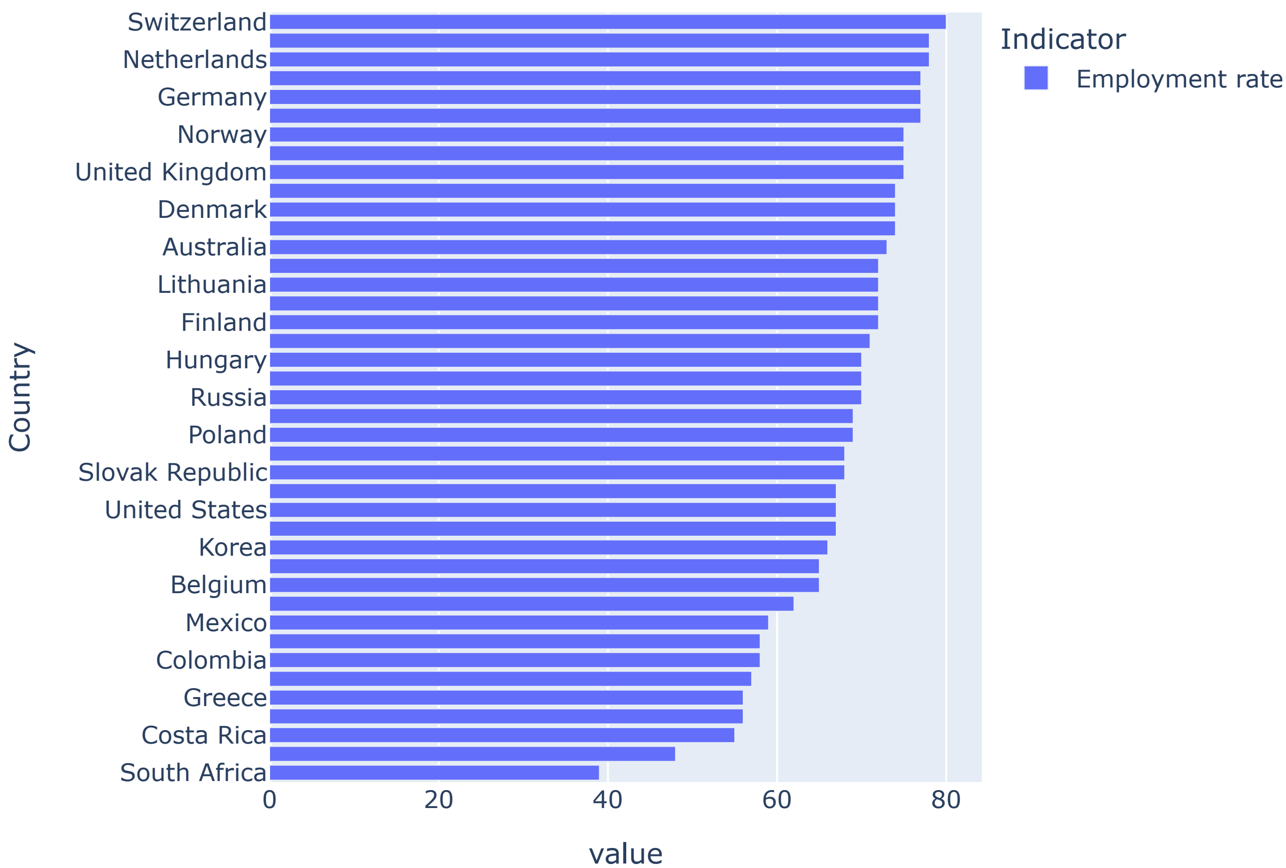

United States
Iceland
Norway
Germany
Austria
Indicator
Household net adjusted disposable income

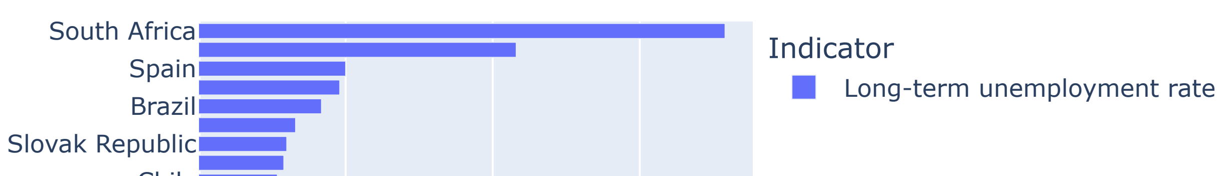

Belgium
France
Sweden
United Kingdom
Italy
Israel
Lithuania
Slovenia
Korea
Poland
Chile
Hungary
Latvia
Colombia
Mexico
South Africa
Country
0
20k
40k
value

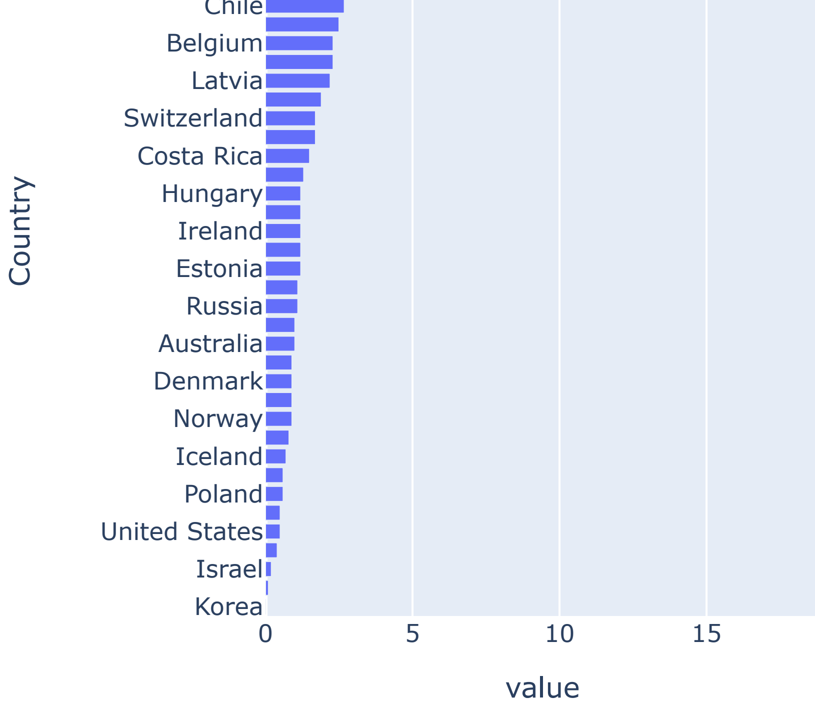

Luxembourg
Australia
Iceland
Canada
Belgium
Ireland
Korea
Germany
Italy
Israel
Portugal
Netherlands
Poland
Brazil
Lithuania
Hungary
Greece
Russia
Colombia
Mexico
South Africa
Country
0
0.2M
0.4M
0.6M
0.8M
value
Indicator
Household net wealth

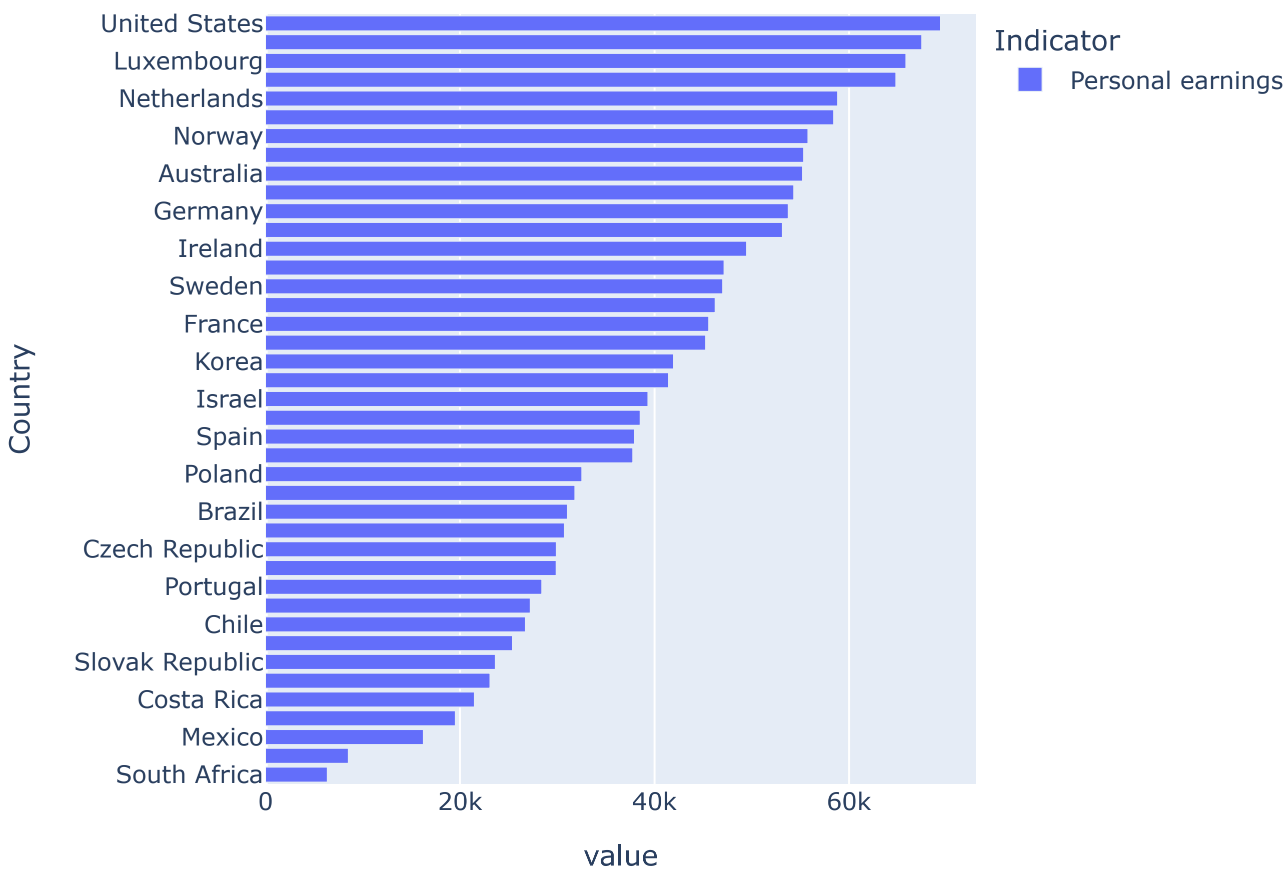

South Africa
Spain
Brazil
Italy
Chile
Indicator
Labour market insecurity


Latvia
Estonia
Poland
Israel
Denmark
United States
Colombia
Hungary
United Kingdom
Australia
Norway
Ireland
Belgium
Austria
Luxembourg
Iceland
Country
0
10
20
30
value

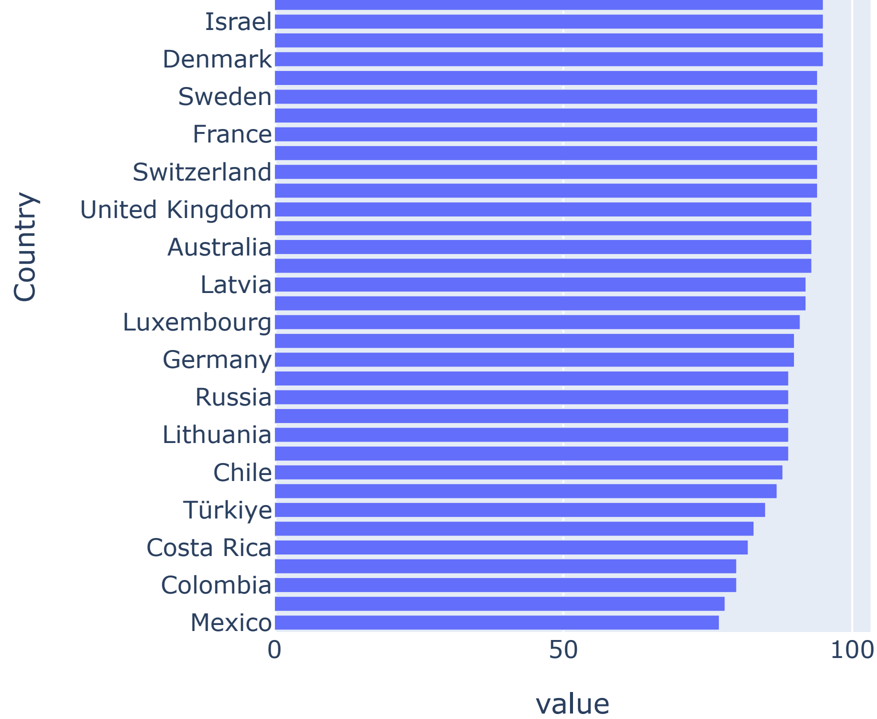

Switzerland
Netherlands
Germany
Norway
United Kingdom
Denmark
Australia
Lithuania
Finland
Hungary
Russia
Poland
Slovak Republic
United States
Korea
Belgium
Mexico
Colombia
Greece
Costa Rica
South Africa
Country
0
20
40
60
80
value
Indicator
Employment rate

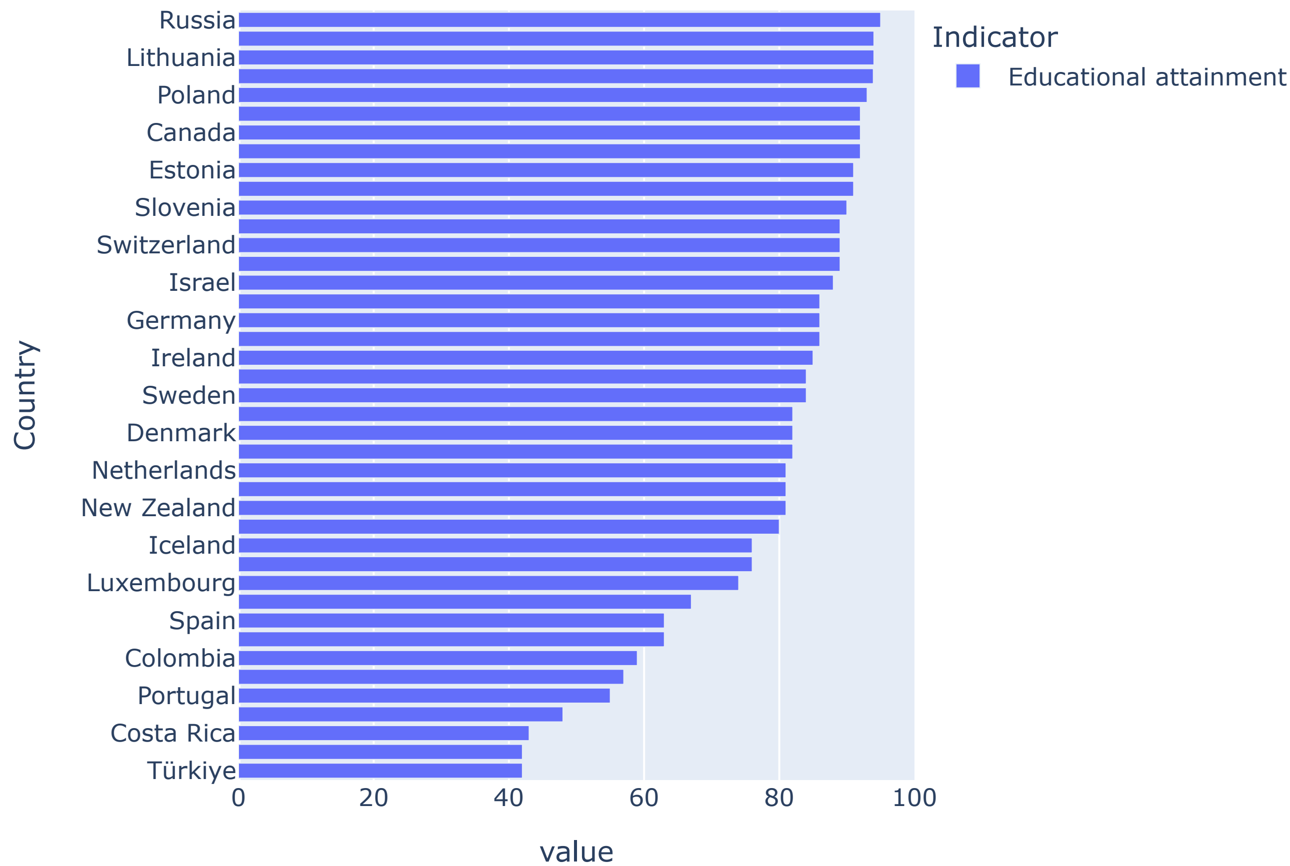

South Africa
Spain
Brazil
Slovak Republic
Indicator
Long-term unemployment rate


Chile
Belgium
Latvia
Switzerland
Costa Rica
Hungary
Ireland
Estonia
Russia
Australia
Denmark
Norway
Iceland
Poland
United States
Israel
Korea
Country
0
5
10
15
value

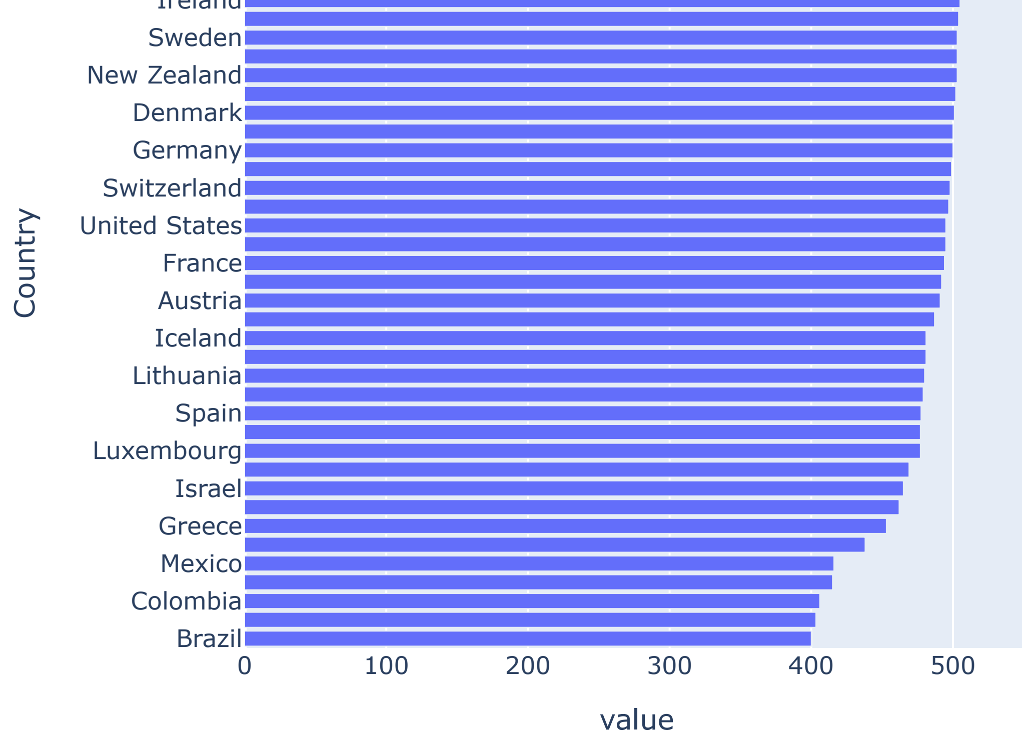

United States
Luxembourg
Netherlands
Norway
Australia
Germany
Ireland
Sweden
France
Korea
Israel
Spain
Poland
Brazil
Czech Republic
Portugal
Chile
Slovak Republic
Costa Rica
Mexico
South Africa
Country
0
20k
40k
60k
value
Indicator
Personal earnings

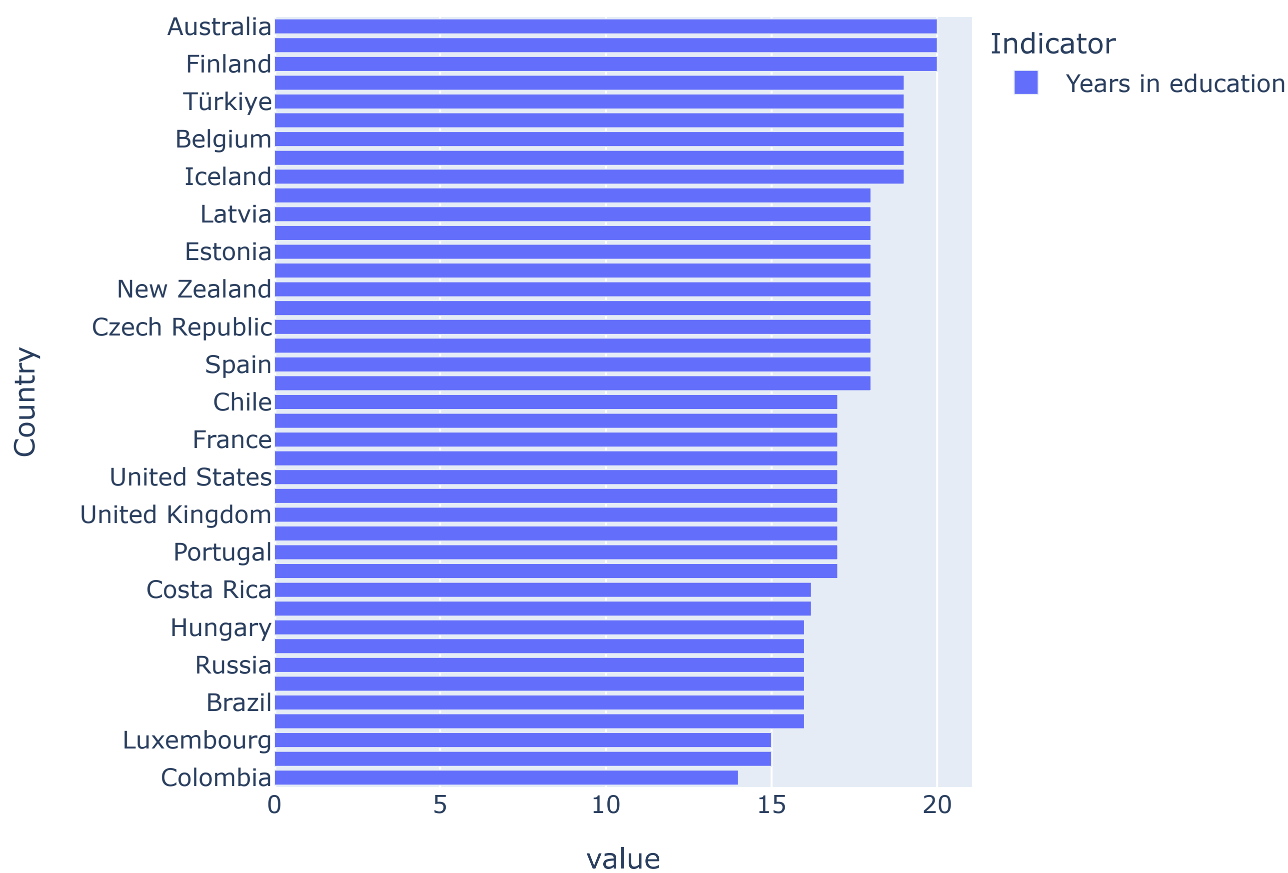

Iceland
Ireland
Norway
New Zealand
Indicator
Quality of support network


Israel
Denmark
Sweden
France
Switzerland
United Kingdom
Australia
Latvia
Luxembourg
Germany
Russia
Lithuania
Chile
Türkiye
Costa Rica
Colombia
Mexico
Country
0
50
100
value

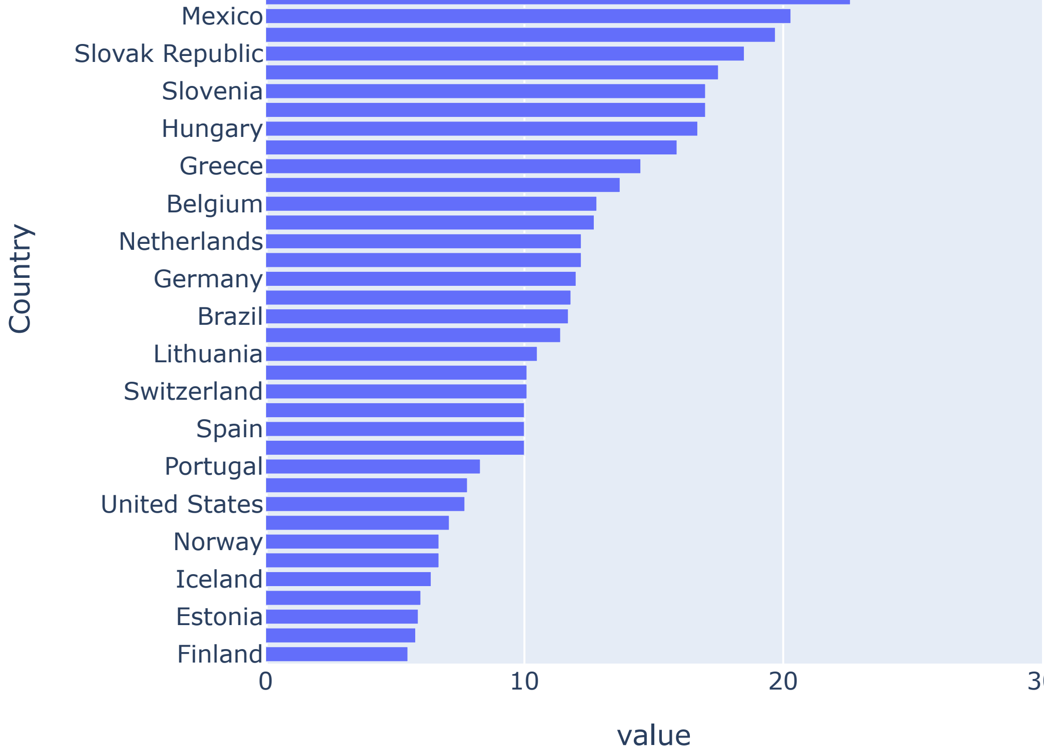

Russia
Lithuania
Poland
Canada
Estonia
Slovenia
Switzerland
Israel
Germany
Ireland
Sweden
Denmark
Netherlands
New Zealand
Iceland
Luxembourg
Spain
Colombia
Portugal
Costa Rica
Türkiye
Country
0
20
40
60
80
100
value
Indicator
Educational attainment

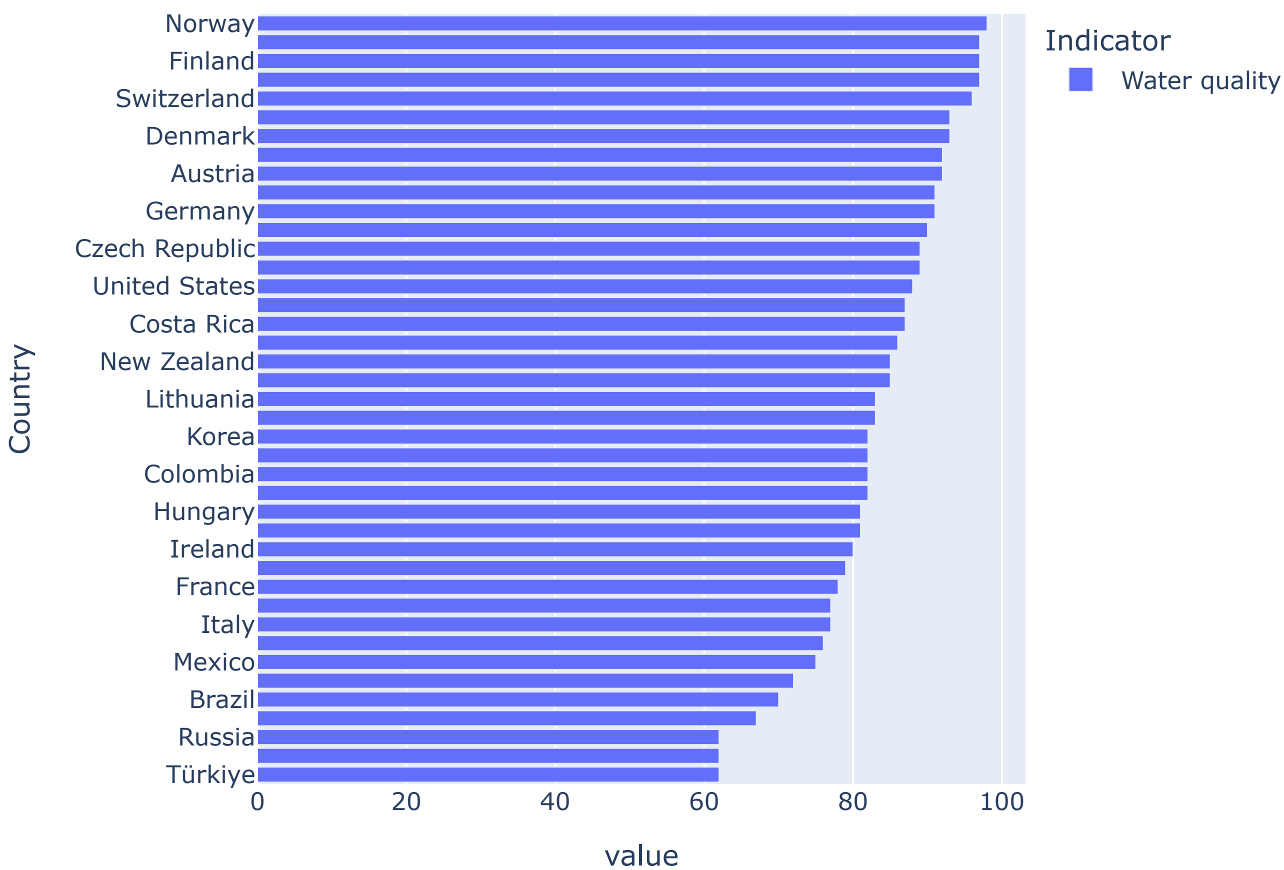

Estonia
Korea
Finland
Indicator
Student skills


Ireland
Sweden
New Zealand
Denmark
Germany
Switzerland
United States
France
Austria
Iceland
Lithuania
Spain
Luxembourg
Israel
Greece
Mexico
Colombia
Brazil
Country
0
100
200
300
400
500
value

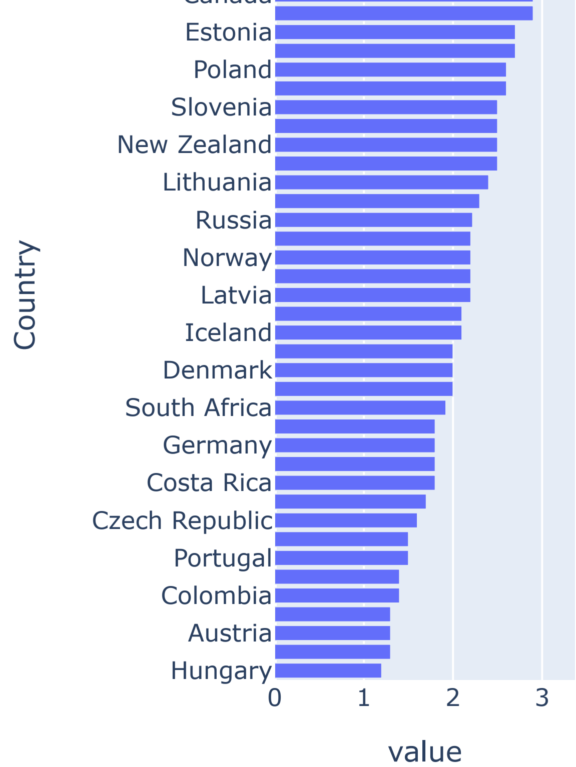

Australia
Finland
Türkiye
Belgium
Iceland
Latvia
Estonia
New Zealand
Czech Republic
Spain
Chile
France
United States
United Kingdom
Portugal
Costa Rica
Hungary
Russia
Brazil
Luxembourg
Colombia
Country
0
5
10
15
20
value
Indicator
Years in education

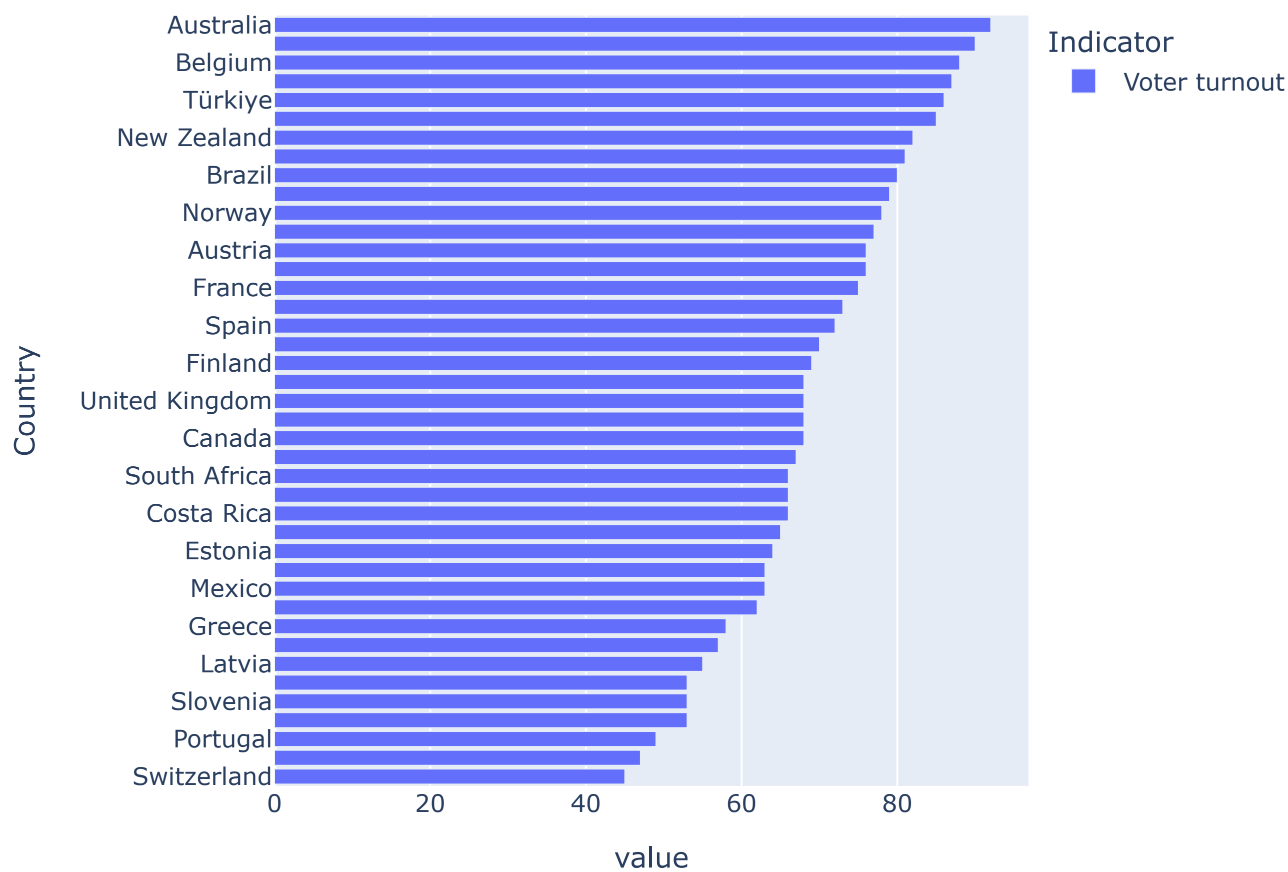

South Africa
Türkiye
Poland
Indicator
Air pollution


Mexico
Slovak Republic
Slovenia
Hungary
Greece
Belgium
Netherlands
Germany
Brazil
Lithuania
Switzerland
Spain
Portugal
United States
Norway
Iceland
Estonia
Finland
Country
0
10
20
30
value

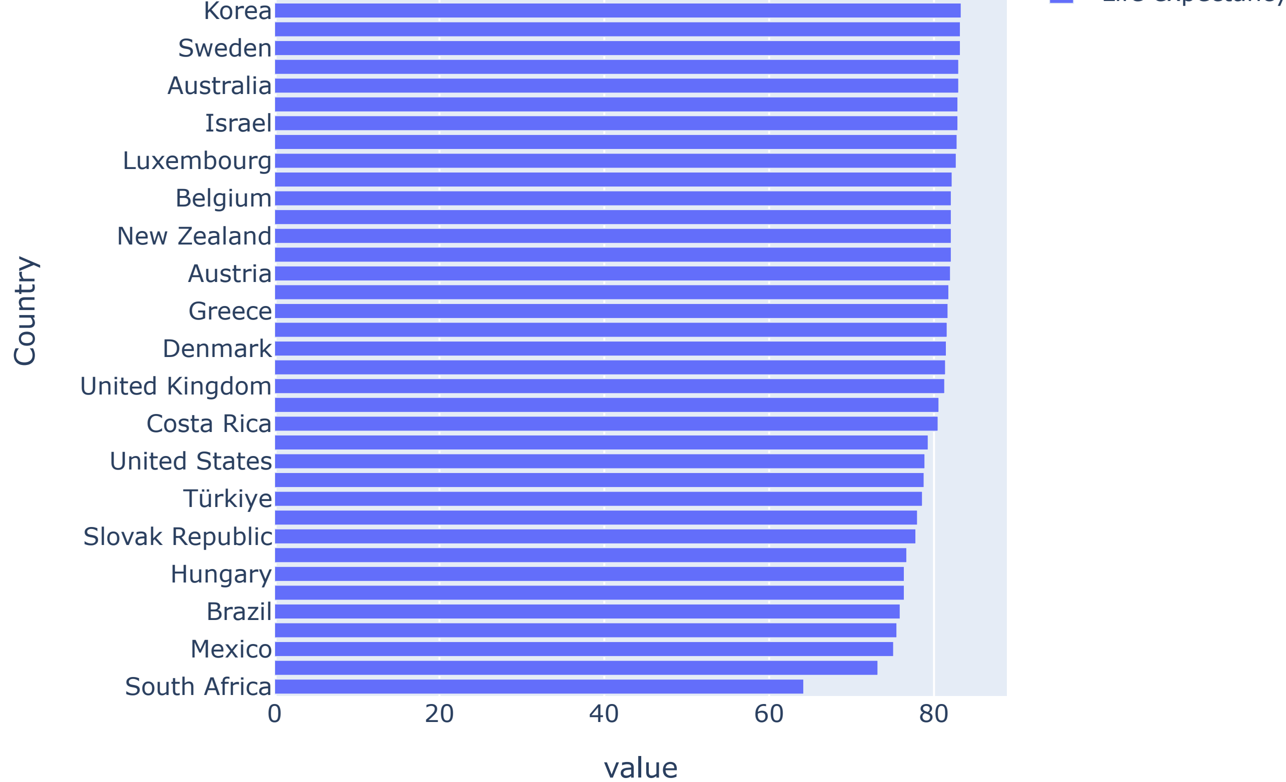

Norway
Finland
Switzerland
Denmark
Austria
Germany
Czech Republic
United States
Costa Rica
New Zealand
Lithuania
Korea
Colombia
Hungary
Ireland
France
Italy
Mexico
Brazil
Russia
Türkiye
Country
0
20
40
60
80
100
value
Indicator
Water quality

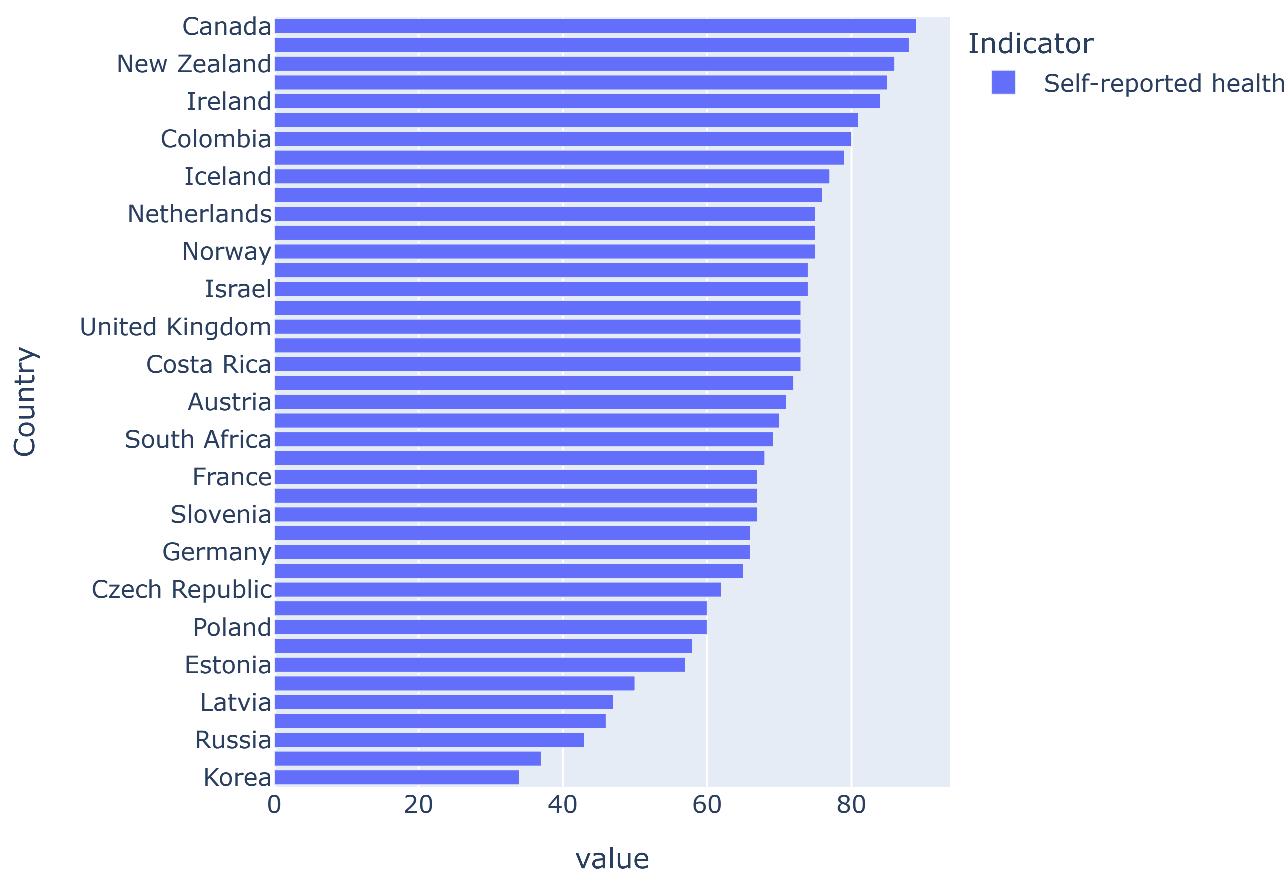

Mexico
United Kingdom
Canada
Indicator
Stakeholder engagement for developing regulations

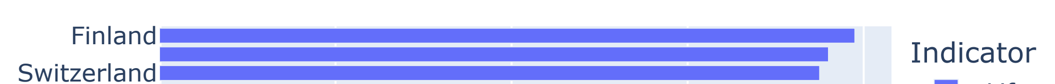

Estonia
Poland
Slovenia
New Zealand
Lithuania
Russia
Norway
Latvia
Iceland
Denmark
South Africa
Germany
Costa Rica
Czech Republic
Portugal
Colombia
Austria
Hungary
Country
0
1
2
3
value

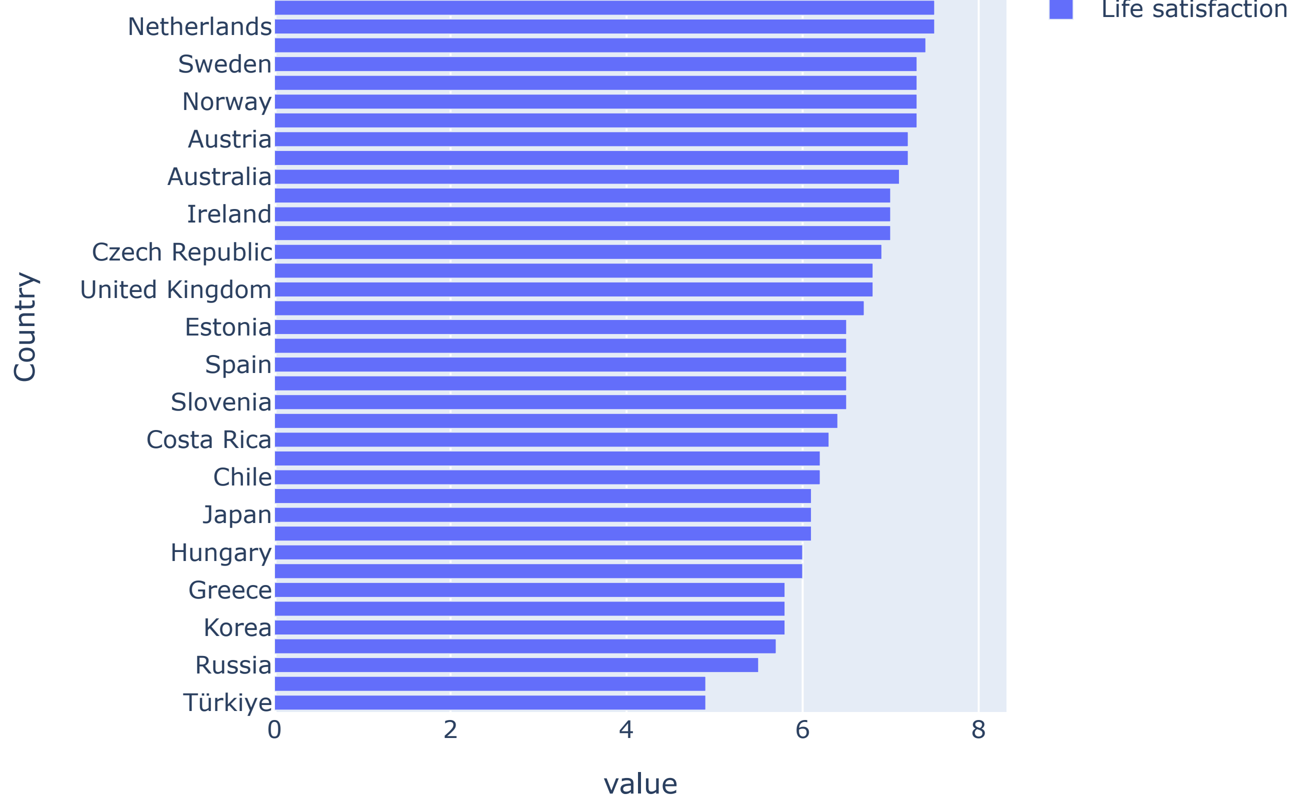

Australia
Belgium
Türkiye
New Zealand
Brazil
Norway
Austria
France
Spain
Finland
United Kingdom
Canada
South Africa
Costa Rica
Estonia
Mexico
Greece
Latvia
Slovenia
Portugal
Switzerland
Country
0
20
40
60
80
value
Indicator
Voter turnout

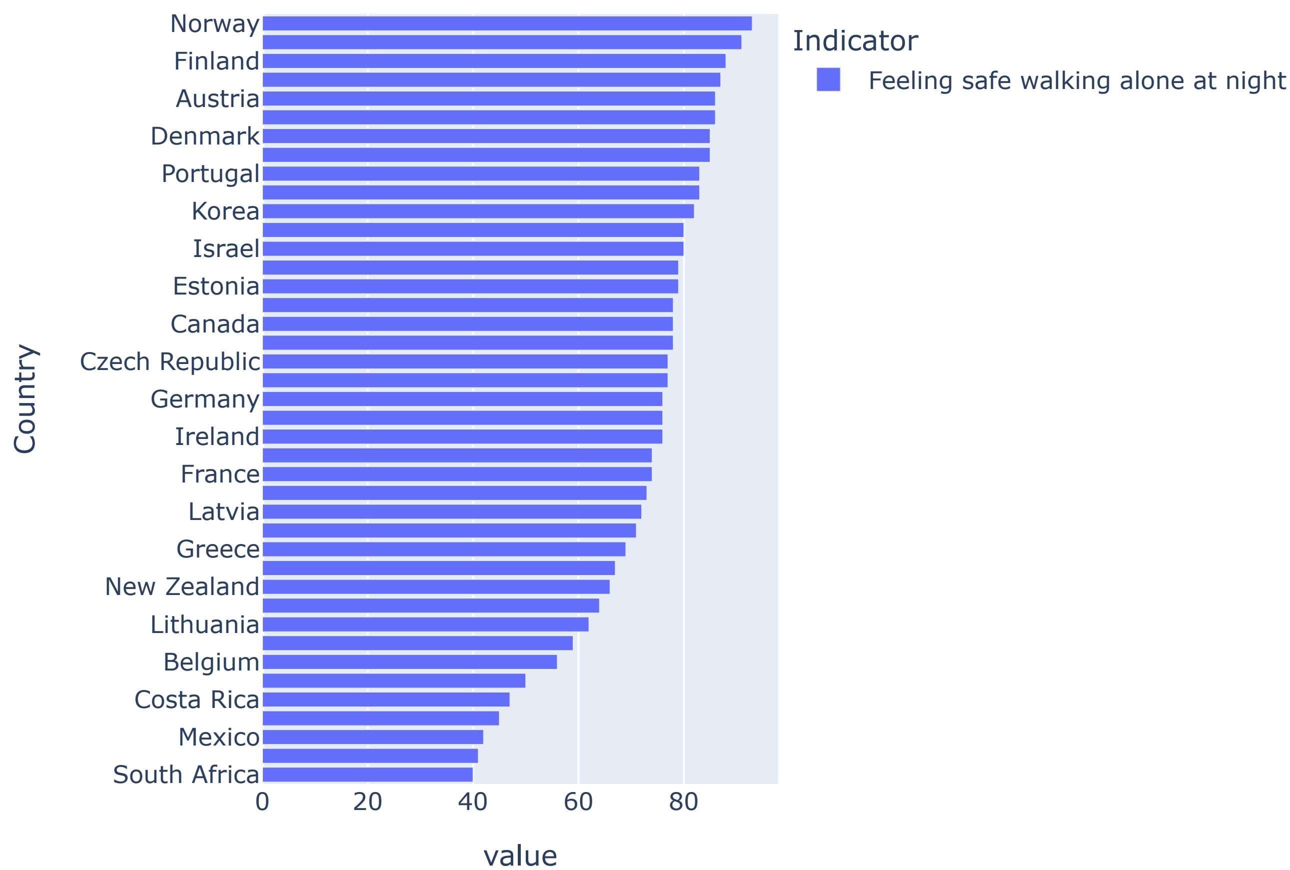

Japan
Spain
Indicator


Korea
Sweden
Australia
Israel
Luxembourg
Belgium
New Zealand
Austria
Greece
Denmark
United Kingdom
Costa Rica
United States
Türkiye
Slovak Republic
Hungary
Brazil
Mexico
South Africa
Country
0
20
40
60
80
value

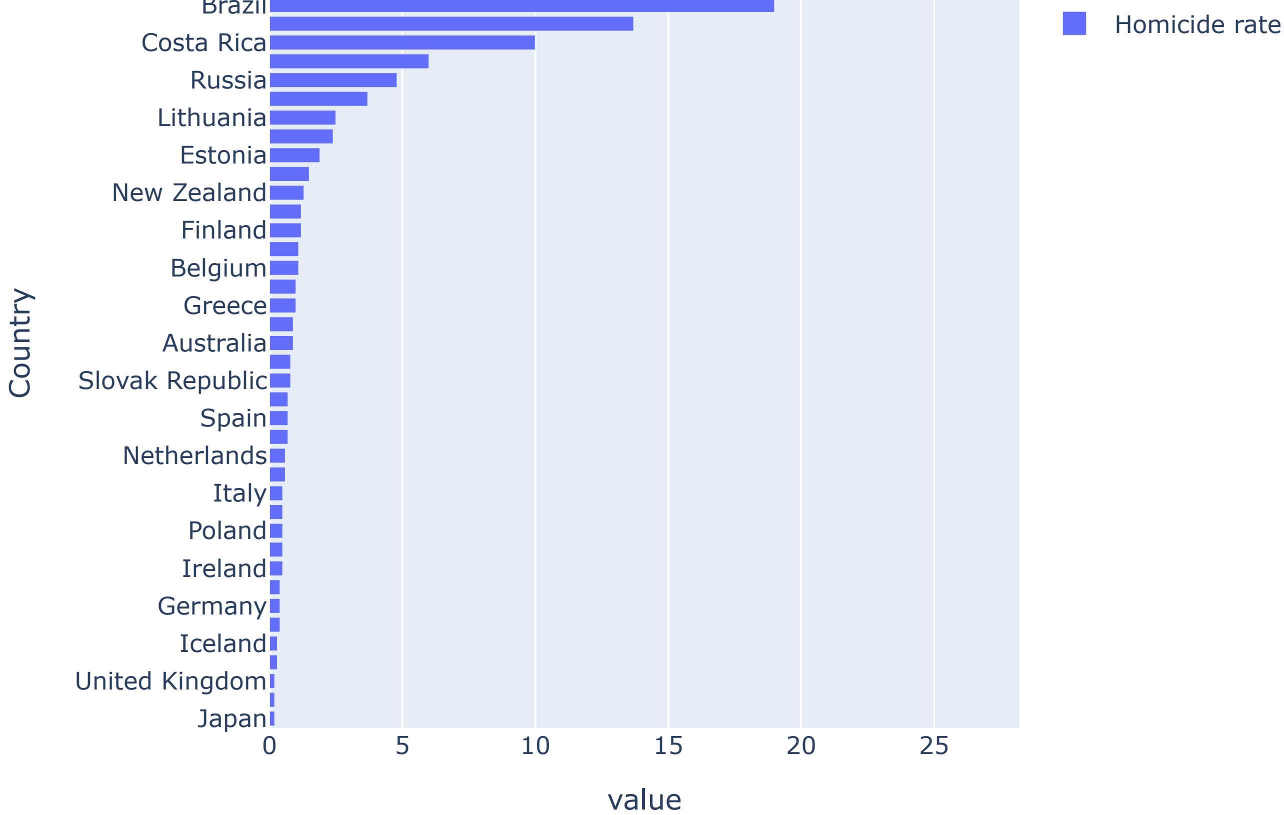

Canada
New Zealand
Ireland
Colombia
Iceland
Netherlands
Norway
Israel
United Kingdom
Costa Rica
Austria
South Africa
France
Slovenia
Germany
Czech Republic
Poland
Estonia
Latvia
Russia
Korea
Country
0
20
40
60
80
value
Indicator
Self-reported health

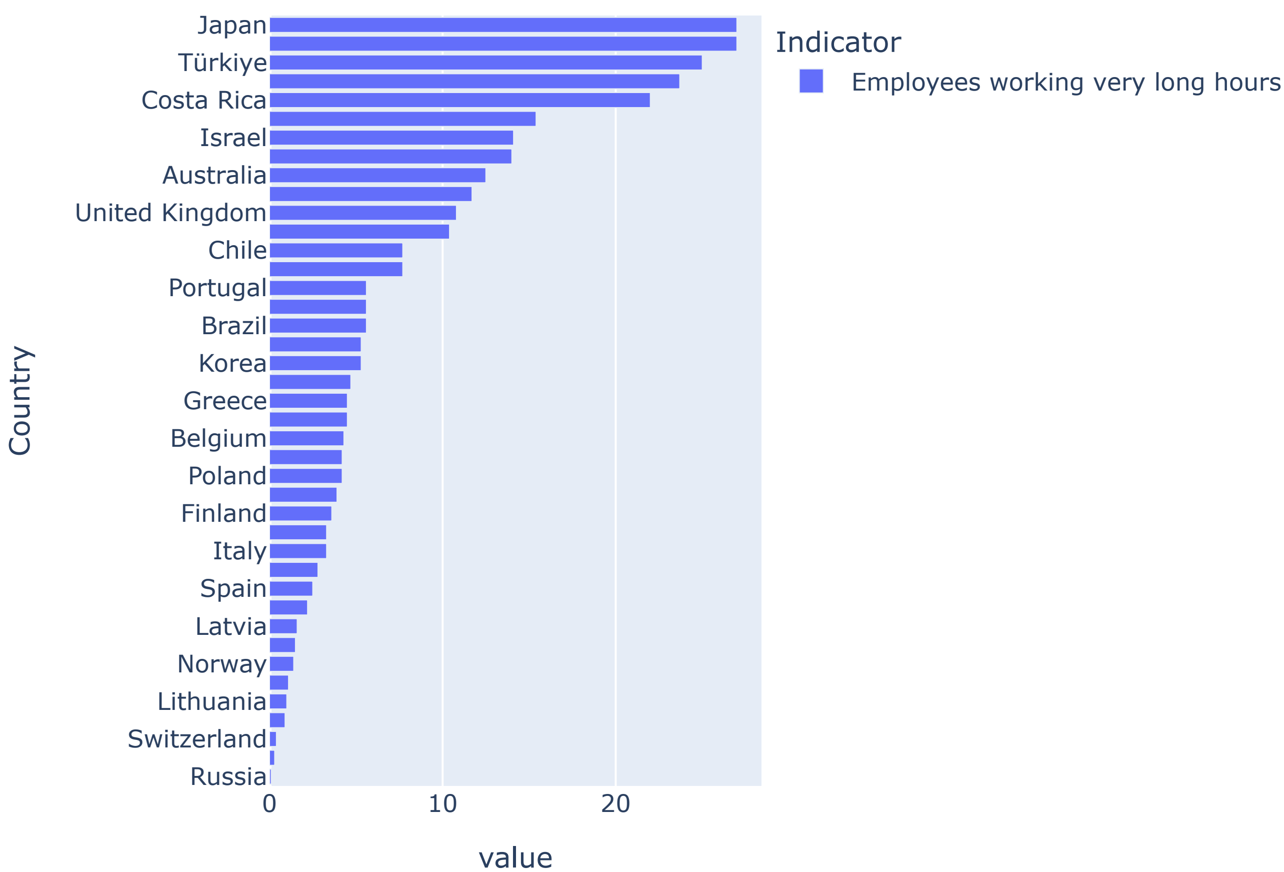

Finland
Switzerland
Indicator


Life satisfaction
Netherlands
Sweden
Norway
Austria
Australia
Ireland
Czech Republic
United Kingdom
Estonia
Spain
Slovenia
Costa Rica
Chile
Japan
Hungary
Greece
Korea
Russia
Türkiye
Country
0
2
4
6
8
value


Indicator
Feeling safe walking alone at night
Norway
Finland
Austria
Denmark
Portugal
Korea
Israel
Estonia
Canada
Czech Republic
Germany
Ireland
France
Latvia
Greece
New Zealand
Lithuania
Belgium
Costa Rica
Mexico
South Africa
Country
0
20
40
60
80
value


Mexico
Indicator





Finally, we will write the final dataset to a CSV file for its further use in Power BI for creating a dashboard.

In [31]:
```
#Rating_fin.to_csv('RatingFin.csv')
```

The research conducted within this notebook is complete, and the dashboard for convenient exploration of the project results can be viewed in the webpage at ___ or in the PowerPoint presentation at _.

Thank you for your time and attention!