ResearchProject

December 6, 2024

0.0.1 IMPORT LIBRARIES

[2]:
```
# Data source: https://www.kaggle.com/datasets/unsdsn/world-happiness/data
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sb
import scipy.stats as stats
import numpy as np
```

0.0.2 LOAD DATA

df1 represents 2018.csv, df2 represents 2019.csv

[8]:
```
df1 = pd.read_csv("2018.csv")
df2 = pd.read_csv("2019.csv")
```

[10]:
```
df1
```

[10]:
```
     Overall rank         Country or region  Score  GDP per capita  \
0               1                   Finland  7.632           1.305
1               2                    Norway  7.594           1.456
2               3                   Denmark  7.555           1.351
3               4                   Iceland  7.495           1.343
4               5               Switzerland  7.487           1.420
..            …                       …    …                …
151           152                     Yemen  3.355           0.442
152           153                  Tanzania  3.303           0.455
153           154               South Sudan  3.254           0.337
154           155  Central African Republic  3.083           0.024
155           156                   Burundi  2.905           0.091

     Social support  Healthy life expectancy  Freedom to make life choices  \
0             1.592                    0.874                         0.681
1             1.582                    0.861                         0.686
2             1.590                    0.868                         0.683
3             1.644                    0.914                         0.677
4             1.549                    0.927                         0.660
..              …                      …                           …
151           1.073                    0.343                         0.244
```

```
152           0.991                    0.381                         0.481
153           0.608                    0.177                         0.112
154           0.000                    0.010                         0.305
155           0.627                    0.145                         0.065

     Generosity  Perceptions of corruption
0         0.202                      0.393
1         0.286                      0.340
2         0.284                      0.408
3         0.353                      0.138
4         0.256                      0.357
..          …                         …
151       0.083                      0.064
152       0.270                      0.097
153       0.224                      0.106
154       0.218                      0.038
155       0.149                      0.076

[156 rows x 9 columns]
```

[12]:
```
df2
```

[12]:
```
     Overall rank         Country or region  Score  GDP per capita  \
0               1                   Finland  7.769           1.340
1               2                   Denmark  7.600           1.383
2               3                    Norway  7.554           1.488
3               4                   Iceland  7.494           1.380
4               5               Netherlands  7.488           1.396
..            …                       …     …             …
151           152                    Rwanda  3.334           0.359
152           153                  Tanzania  3.231           0.476
153           154               Afghanistan  3.203           0.350
154           155  Central African Republic  3.083           0.026
155           156               South Sudan  2.853           0.306

     Social support  Healthy life expectancy  Freedom to make life choices  \
0             1.587                    0.986                         0.596
1             1.573                    0.996                         0.592
2             1.582                    1.028                         0.603
3             1.624                    1.026                         0.591
4             1.522                    0.999                         0.557
..              …                      …                            …
151           0.711                    0.614                         0.555
152           0.885                    0.499                         0.417
153           0.517                    0.361                         0.000
154           0.000                    0.105                         0.225
155           0.575                    0.295                         0.010
```

```
     Generosity  Perceptions of corruption
0         0.153                      0.393
1         0.252                      0.410
2         0.271                      0.341
3         0.354                      0.118
4         0.322                      0.298
..          …                          …
151       0.217                      0.411
152       0.276                      0.147
153       0.158                      0.025
154       0.235                      0.035
155       0.202                      0.091

[156 rows x 9 columns]
```

0.0.3 DATA PREPROCESSING

[15]:

```
print(df1.isna().values.any())
print(df2.isna().values.any())
```

```
True
False
```

Missing values in df1: determine which column it is

[18]:

```
for i in df1.columns:
    print(f"{i}:{df1[i].isna().values.any()}")
```

```
Overall rank:False
Country or region:False
Score:False
GDP per capita:False
Social support:False
Healthy life expectancy:False
Freedom to make life choices:False
Generosity:False
Perceptions of corruption:True
```

Missing values in perceptions of corruption: replace NA value with mean of column

[21]:

```
x = np.mean(df1['Perceptions of corruption'])
df1['Perceptions of corruption'] = df1['Perceptions of corruption'].fillna(x)
```

0.0.4 DATA EXPLORATION

[24]:

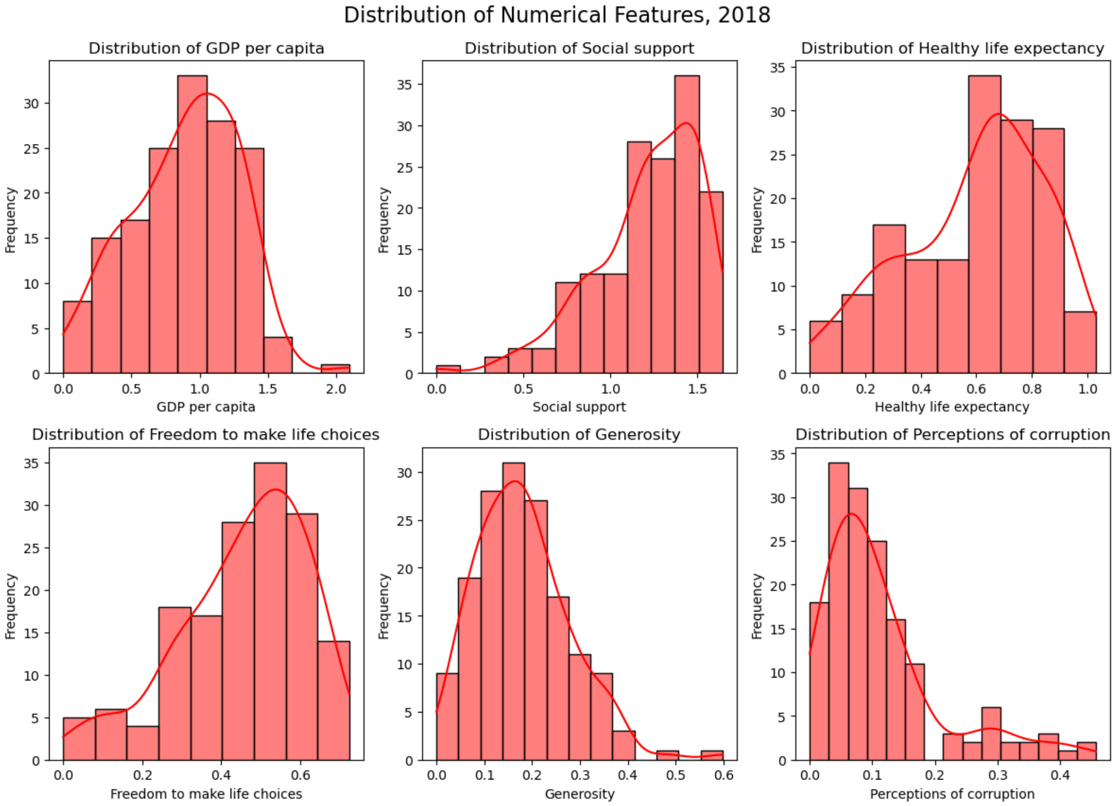
```
columns = ['GDP per capita', 'Social support',
       'Healthy life expectancy', 'Freedom to make life choices', 'Generosity',
       'Perceptions of corruption']
```

```
x = range(1,7)
```

2018

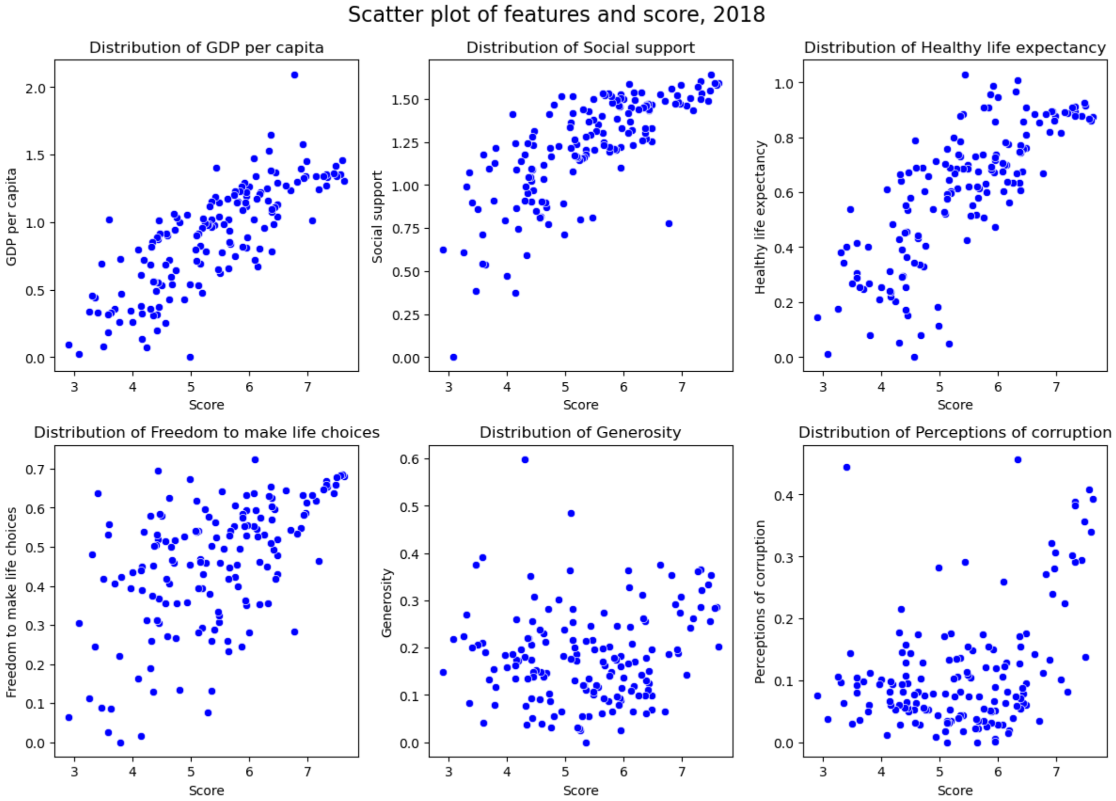
[27]:

```
plt.figure(figsize=(12, 12))
pairs = zip(columns, x)
for column, i in pairs:
    plt.subplot(int(len(columns) / 3 + 1), 3, i)
    sb.histplot(df1[column], color='red', kde=True)
    plt.ylabel("Frequency")
    plt.xlabel(column)
    plt.title(f"Distribution of {column}")
plt.tight_layout()
plt.suptitle("Distribution of Numerical Features, 2018", y=1.02, fontsize=16)
plt.show()
```



[29]:

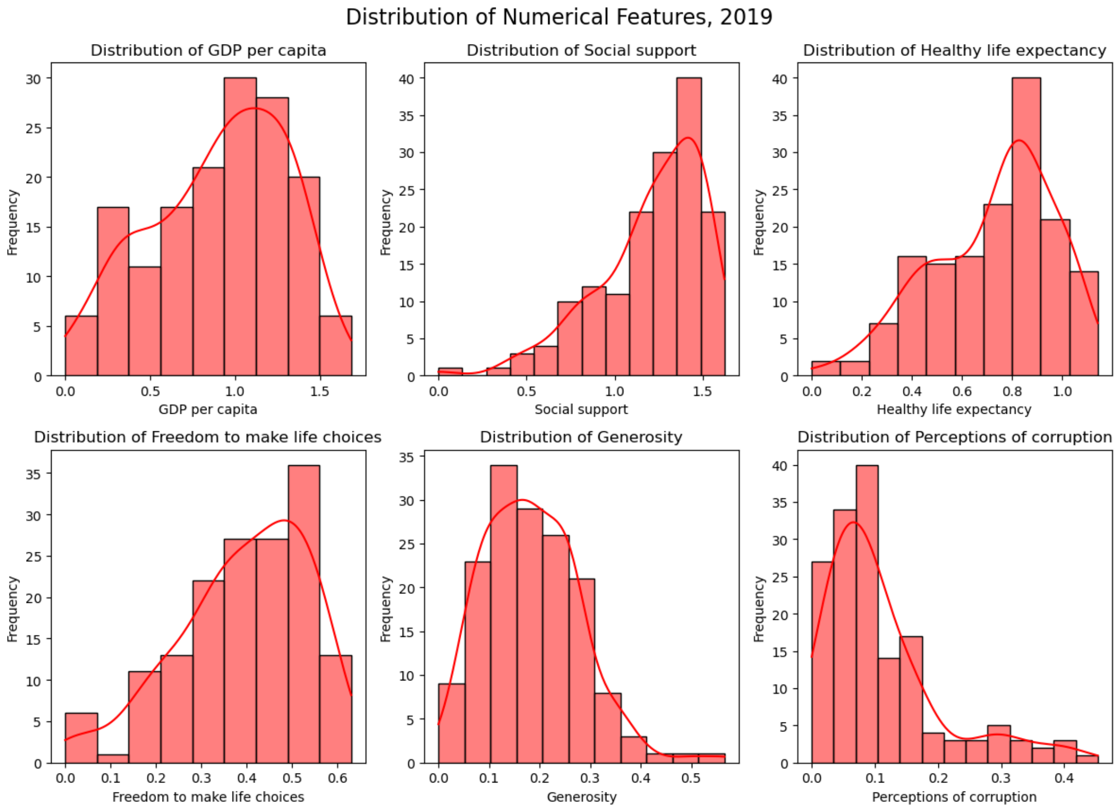
```
plt.figure(figsize=(12, 12))
pairs = zip(columns, x)
for column, i in pairs:
    plt.subplot(int(len(columns) / 3 + 1), 3, i)
    sb.scatterplot(x = df1['Score'], y = df1[column], color='blue')
    plt.ylabel(column)
    plt.xlabel("Score")
    plt.title(f"Distribution of {column}")
plt.tight_layout()
plt.suptitle("Scatter plot of features and score, 2018", y=1.02, fontsize=16)
plt.show()
```



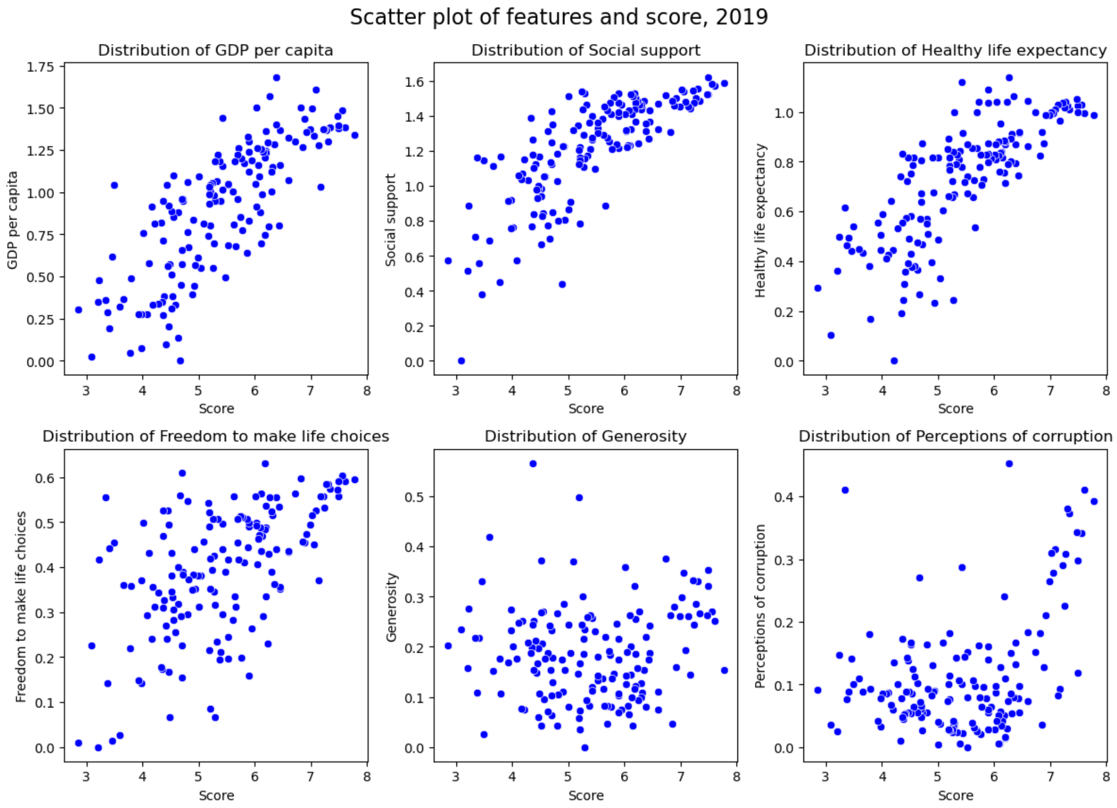
2019

[32]:

```
plt.figure(figsize=(12, 12))
pairs = zip(columns, x)
for column, i in pairs:
    plt.subplot(int(len(columns) / 3 + 1), 3, i)
    sb.histplot(df2[column], kde=True, color='red')
    plt.ylabel("Frequency")
    plt.xlabel(column)
    plt.title(f"Distribution of {column}")
plt.tight_layout()
plt.suptitle("Distribution of Numerical Features, 2019", y=1.02, fontsize=16)
plt.show()
```



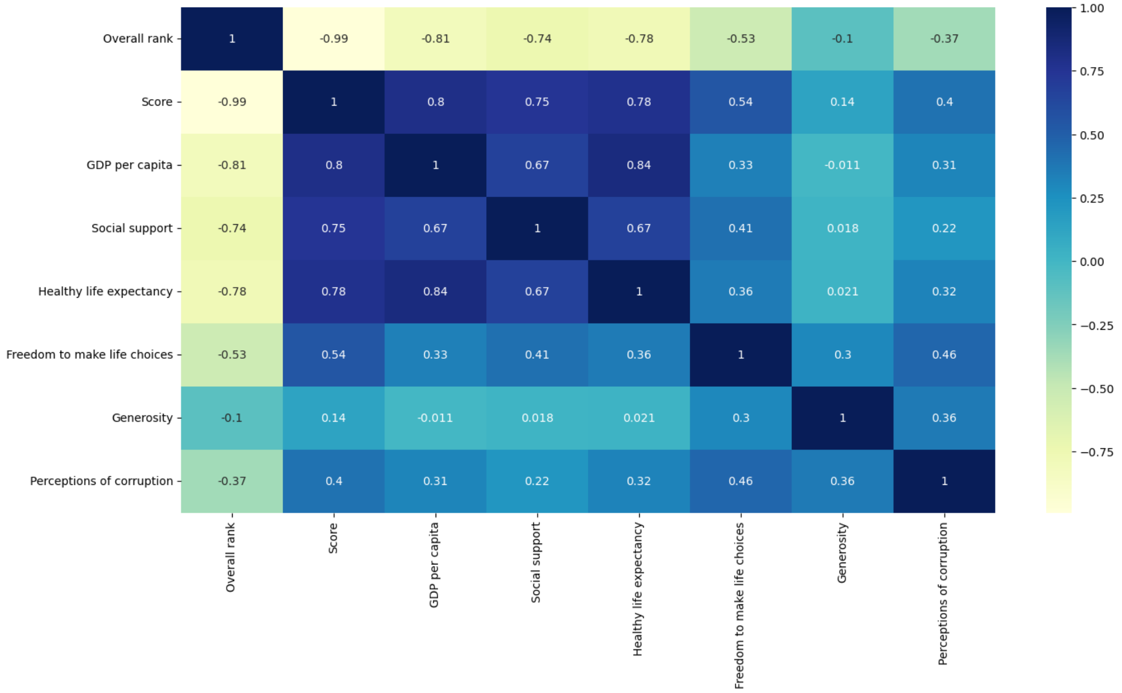
[34]:
```
plt.figure(figsize=(12, 12))
pairs = zip(columns, x)
for column, i in pairs:
    plt.subplot(len(columns) // 3 + 1, 3, i)
    sb.scatterplot(x = df2['Score'], y = df2[column], color='blue')
    plt.title(f"Distribution of {column}")
    plt.xlabel("Score")
    plt.ylabel(column)
plt.tight_layout()
plt.suptitle("Scatter plot of features and score, 2019", y=1.02, fontsize=16)
plt.show()
```



0.0.5 DATA ANALYSIS

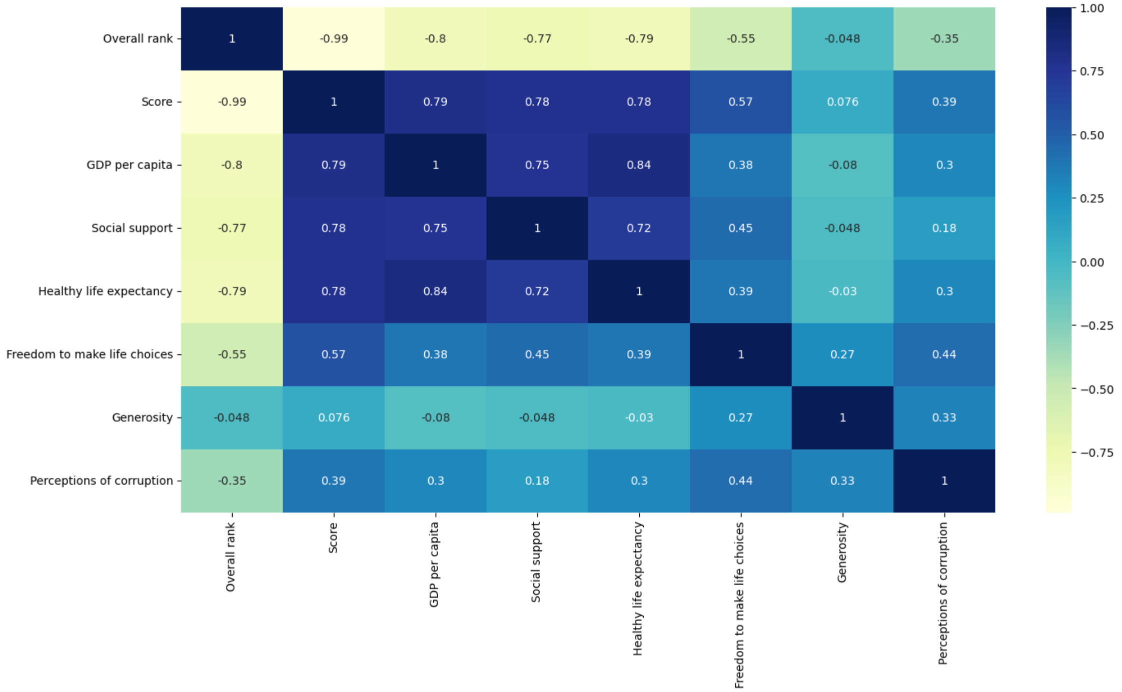
2018

[38]:
```
plt.figure(figsize=(16,8))
dataplot = sb.heatmap(df1.drop(columns = ['Country or region'], inplace =␣
  ↪False).corr(numeric_only=True), cmap="YlGnBu", annot=True)
```



2019

```
[41]: plt.figure(figsize=(16,8))
dataplot = sb.heatmap(df2.drop(columns = ['Country or region'], inplace =␣
    ↪False).corr(numeric_only=True), cmap="YlGnBu", annot=True)
```



0.0.6 EVALUATION

[44]:

```
pval2018 = []
pval2019 = []
corr2018 = []
corr2019 = []
for i in columns:
    corr = df1['Score'].corr(df1[i])
    r, p_value = stats.pearsonr(df1['Score'], df1[i])
    pval2018.append(p_value)
    corr2018.append(corr)

for i in columns:
    corr = df2['Score'].corr(df2[i])
    r, p_value = stats.pearsonr(df2['Score'], df2[i])
    pval2019.append(p_value)
    corr2019.append(corr)

dd = {'Feature': columns, 'correlation 2018': corr2018, 'p-value 2018':␣
 ↪pval2018, 'correlation 2019': corr2019, 'p-value 2019': pval2019}
eval = pd.DataFrame(data = dd)
eval
```

[44]:

```
                        Feature  correlation 2018  p-value 2018  \
0                GDP per capita          0.802124  2.626646e-36
1                Social support          0.745760  5.878287e-29
2       Healthy life expectancy          0.775814  1.307391e-32
3  Freedom to make life choices          0.544280  2.074589e-13
4                    Generosity          0.135825  9.090351e-02
5     Perceptions of corruption          0.403234  1.796884e-07

   correlation 2019  p-value 2019
0          0.793883  4.315481e-35
1          0.777058  8.975120e-33
2          0.779883  3.785454e-33
3          0.566742  1.237924e-14
4          0.075824  3.468195e-01
5          0.385613  6.654011e-07
```