PRA2

June 8, 2021

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

1 Descripció

El "World Happiness Report" [1] és una enquesta realitzada per tal de mesurar l'estat mundial i nacional de felicitat. Ha estat realitzant informes i recollint dades al respecte des de l'any 2012 fins a l'actualitat, recol · lectant informació en pràcticament tots els països del món.

Les puntuacions utilitzades en les diferents categories del "World Happiness Report" fan servir les dades del "Gallup World Poll" [2], que enquesta aproximadament 1.000 persones per país cara a cara o per telèfon. Aquestes puntuacions es basen en les respostes donades a la categoria de "life evaluation", mesurades segons la "Cantril's Ladder of Life Scale" [3]. Per cada país i any, es mostra una puntuació de felicitat ("Happiness Score") que indica com de feliç se sent la població d'aquell país. Per entendre aquest indicador, s'utilitzen 6 atributs: economia, família, esperança de vida, llibertat, confiança en les institucions públiques i generoistat.

Sobretot en els països més desnvolupats, cada vegada es dóna més importància a la felicitat de la població. Evidentment són molt rellevants indicadors com els econòmics, socials o mediambientals, però de què serveix que aquests tinguin molt bones puntuacions si la gent d'un país no se sent feliç? És per això que cada vegada s'està debatent més el fet de tenir en compte un indicador com el de la felicitat. Països com Nova Zelanda han votat a favor de tenir en compte l'índex de felicitat de la població i de destinar una part del pressupost estatal en millorar-lo cada any [4].

Per altra banda, no falten crítiques en aquesta manera de valorar la felicitat d'una nació, fins i tot mostrant inconsistències entre diferents índexs de felicitat [5]. Una de les crítiques principals és el fet de valorar una puntuació de ben estar segons l'opinió subjectiva de la gent, que pot estar altament influenciada per molts factors socials i pot resultar difícil de comparar entre diferents països [6].

Nogensmenys, probablement és important analitzar i veure les conclusions que es poden extreure dels diferents índexs, sempre tenint en compte els seus desavantatges i inconvenients. En aquesta entrega ens centrarem en analitzar les dades extretes del "World Happiness Report" [7], però alertem que pot ser important contrastar aquesta informació amb altres fonts i índexs que valorin les seves variables i puntuacions de felicitat de maneres diferents.

1.1 Dades

Les dades que s'utilitzaran per a la realització d'aquest informe s'han descarregat d'un dels datasets públics de Kaggle [7]. Estan compostes per la mesura de diferents indicadors en bona part dels països del món entre els anys 2015 i 2019. Els atributs de les dades són els següents:

- **Country**: País en el qual s'han recollit les dades
- **Region**: Regió en la qual pertany el país.
- **Country or region**: País o regió en la qual pertany el país.
- **Economia**: Bastant-se en el PIB per càpita del país.
- **Family**: Sensació de suport i cohesió de la família.
- **Freedom**: Llibertat per prendre decisions sobre la vida personal.
- **Generosity**: Percepció de generositat general de la població.
- **Health**: Esperança de vida.
- **Trust**: Percepció de confiança o corrupció en el govern.
- **Happines Score**: Puntuació de felicitat calculada a partir dels altres indicadors.
- **Standard Error**: Error estàndard de la puntuació de felicitat.
- **Happiness Rank**: Posició general del país en quan a la puntuació de felicitat.
- **Dystopia Residual**: Comparació de cada país amb un país imaginari (Dystopia) on hi ha la població més infeliç del planeta.
- **Lower Confidence Interval**: Marca l'interval de confiança inferior del 95% dels "ladder scores".
- **Upper Confidence Interval**: Marca l'interval de confiança superior del 95% dels "ladder scores".

1.2 Objectius

A partir de l'anàlisi de les dades de felicitat, s'intentarà donar resposta a diferents preguntes:

- Quins dels sis factors esmentats tenen més rellevància a l'hora de determinar la felicitat general d'un país?
- És possible predir amb precisió la puntuació de felicitat d'un país a partir d'aquests sis factors?
- Quins són els països que més han incrementat la seva felicitat entre l'any 2015 i 2019 i en quina mesura? Quins són en els que més ha decrementat? Hi ha cap raó evident segons els sis indicadors utilitzats?
- Com es comparen les puntuacions de felicitat entre les diferents regions del planeta?
- Com es posiciona Espanya en cadascun dels sis factors i la puntuació de felicitat a nivell europeu? I a nivell mundial?

2 Integració i selecció

Com que tenim un fitxer separat per cada any de recollida de dades, definim un diccionari per guardar-hi cadascun d'aquests datasets.

L'objectiu d'aquest apartat serà unificar totes aquestes dades en un sol dataset uniforme i preparat per la posterior neteja i anàlisi.

[2]:
```
# diccionari on guardarem els datasets per any
data = {}
```

2.1 Lectura de dades

Carreguem cadascuna de les 5 fonts de dades dins del diccionari:

[3]:
```
for file in sorted(os.listdir('data/')):
    # any de les dades (és el mateix nom del fitxer)
    year = file.split('.')[0]
    # guardem les dades per any al diccionari
    data[year] = pd.read_csv(f'data/{file}')
```

2.2 Estandarització de columnes

Com que hem carregat diferents fonts de dades, anem a comprovar que els noms de columnes coincideixen entre els diferents datasets. Per fer-ho, mirem tots els noms únics de cadascuna de les fonts:

[4]:
```
unique_cols = np.unique([col for df in data.values() for col in df.columns])
unique_cols.reshape(-1,1)
```

[4]:
```
array([['Country'],
       ['Country or region'],
       ['Dystopia Residual'],
       ['Dystopia.Residual'],
       ['Economy (GDP per Capita)'],
       ['Economy..GDP.per.Capita.'],
       ['Family'],
       ['Freedom'],
       ['Freedom to make life choices'],
       ['GDP per capita'],
       ['Generosity'],
       ['Happiness Rank'],
       ['Happiness Score'],
       ['Happiness.Rank'],
       ['Happiness.Score'],
       ['Health (Life Expectancy)'],
       ['Health..Life.Expectancy.'],
       ['Healthy life expectancy'],
       ['Lower Confidence Interval'],
       ['Overall rank'],
       ['Perceptions of corruption'],
       ['Region'],
       ['Score'],
       ['Standard Error'],
       ['Trust (Government Corruption)'],
       ['Trust..Government.Corruption.'],
```

```
        ['Upper Confidence Interval'],
        ['Whisker.high'],
        ['Whisker.low']], dtype='<U29')
```

Veiem que els noms canvien substancialment entre els diferents fitxers. Per exemple, a partir de 2017 la columna "Economy (GDP per Capita)" passa a anomenar-se "Economy..GDP.per.Capita". També, hi ha variables que s'utilitzen en alguns datasets i en d'altres no. Per exemple, el 2015 té atributs com "Standard Error" que ja no s'utilitzen més en la resta d'anys.

No ens preocupa el fet de tenir algunes variables diferents, ja que ens encarregarem de tractar aquests valors nuls en la neteja posterior de valors buits. Ara bé, seria convenient estandariditzar els diferents noms de variables que tenen el mateix significat.

Per fer-ho, definim un diccionari i una funció de traducció per tal de normalitzar tots els noms amb el mateix significat. Per tal de fer-ho més estàndard, agafem com a convenció que les diferents paraules estaran separades per un espai i cada paraula començarà amb lletra majúscula.

[5]:
```
# els atributs que continguin la clau del diccionari com a nom, es traduïran␣
↪segons indiqui el valor
translations = {
    'dystopia': 'Dystopia Residual',
    'economy': 'Economy',
    'gdp': 'Economy',
    'freedom': 'Freedom',
    'rank': 'Happiness Rank',
    'score': 'Happiness Score',
    'health': 'Health',
    'trust': 'Trust',
    'corruption': 'Trust',
    'whisker.high': 'Upper Confidence Interval',
    'whisker.low': 'Lower Confidence Interval',
}
```

[6]:
```
# traducció dels noms de columnes
for _, df in data.items():
    transl = {}
    for contained_col, final_col in translations.items():
        for df_col in df.columns:
            if contained_col in df_col.lower():
                transl[df_col] = final_col

    # canvia els noms de les columnes
    df.rename(columns=transl, inplace=True)
```

[7]:
```
# estandarització del format dels noms
for _, df in data.items():
    transl = {}
    for col in df.columns:
```

```
        # canviem punts per espais i comencem cada paraula en majúscules
        transl[col] = col.replace('.', ' ').title()

    # canvia els noms de les columnes
    df.rename(columns=transl, inplace=True)
```

Per últim, comprovem que els noms finals de les columnes s'ha estandaritzat correctament i no tenim significats repetits amb diferents noms:

[8]:
```
unique_cols = np.unique([col for df in data.values() for col in df.columns])
unique_cols.reshape(-1,1)
```

[8]:
```
array([['Country'],
       ['Country Or Region'],
       ['Dystopia Residual'],
       ['Economy'],
       ['Family'],
       ['Freedom'],
       ['Generosity'],
       ['Happiness Rank'],
       ['Happiness Score'],
       ['Health'],
       ['Lower Confidence Interval'],
       ['Region'],
       ['Standard Error'],
       ['Trust'],
       ['Upper Confidence Interval']], dtype='<U25')
```

2.3 Integració dels anys

Abans d'agrupar les dades, ens interessa mantenir la informació dels anys als quals pertanyen. Per això, creem una nova columna en cadascun dels datasets que indiqui l'any en que es van recollir les dades:

[9]:
```
for year, df in data.items():
    df['Year'] = int(year)
```

2.4 Agrupació de dades

Prosseguim a ajuntar totes les dades en un únic dataset. Veiem que ens queden un total de 782 files (unes 155 ciutats per cada any entre 2015 i 2019) i 16 columnes. També, ja podem observar que hi ha alguns valors buits (NaN), però això es tractarà a l'apartat 3.

[10]:
```
df = pd.concat(data.values()).reset_index(drop=True)
df.head()
```

[10]:

	Country	Region	Happiness Rank	Happiness Score	\
0	Switzerland	Western Europe	1	7.587	

```
1      Iceland  Western Europe                2            7.561
2      Denmark  Western Europe                3            7.527
3       Norway  Western Europe                4            7.522
4       Canada   North America                5            7.427

   Standard Error  Economy   Family   Health  Freedom    Trust  Generosity  \
0         0.03411  1.39651  1.34951  0.94143  0.66557  0.41978     0.29678
1         0.04884  1.30232  1.40223  0.94784  0.62877  0.14145     0.43630
2         0.03328  1.32548  1.36058  0.87464  0.64938  0.48357     0.34139
3         0.03880  1.45900  1.33095  0.88521  0.66973  0.36503     0.34699
4         0.03553  1.32629  1.32261  0.90563  0.63297  0.32957     0.45811

   Dystopia Residual  Year  Lower Confidence Interval  \
0            2.51738  2015                        NaN
1            2.70201  2015                        NaN
2            2.49204  2015                        NaN
3            2.46531  2015                        NaN
4            2.45176  2015                        NaN

   Upper Confidence Interval Country Or Region
0                        NaN                NaN
1                        NaN                NaN
2                        NaN                NaN
3                        NaN                NaN
4                        NaN                NaN
```

[11]:
```
df.shape
```

[11]:
```
(782, 16)
```

2.5 Reordenació de columnes

Per tal que visualment sigui més còmode entendre i interpretar les dades en forma tabular, reordenem les columnes per tal de tenir les que més ens interessen al principi (esquerra) de la taula, aquestes són: Country, Region, Year, Happiness Rank i Happiness Score.

Els sis indicadors estaran a la part central de la taula. Al final hi posem aquells atributs que probablement no utilitzarem tant durant l'anàlisi, aquests són: Dystopia Residual, Standard Error, Lower and Upper Confidence Interval.

[12]:
```
# columnes que volem al principi
first_cols = ['Country', 'Region', 'Country Or Region', 'Year', 'Happiness␣
 ↪Rank', 'Happiness Score']
last_cols = ['Dystopia Residual', 'Standard Error',
             'Lower Confidence Interval', 'Upper Confidence Interval']
# resta de columnes
mid_cols = list(set(df.columns) - set(first_cols) - set(last_cols))
```

```
df = df[first_cols + mid_cols + last_cols]
df.head()
```

[12]:

	Country	Region	Country Or Region	Year	Happiness Rank	Happiness Score	Health	Trust	Family	Economy	Freedom	Generosity	Dystopia Residual	Standard Error	Lower Confidence Interval	Upper Confidence Interval
0	Switzerland	Western Europe	NaN	2015	1	7.587	0.94143	0.41978	1.34951	1.39651	0.66557	0.29678	2.51738	0.03411	NaN	NaN
1	Iceland	Western Europe	NaN	2015	2	7.561	0.94784	0.14145	1.40223	1.30232	0.62877	0.43630	2.70201	0.04884	NaN	NaN
2	Denmark	Western Europe	NaN	2015	3	7.527	0.87464	0.48357	1.36058	1.32548	0.64938	0.34139	2.49204	0.03328	NaN	NaN
3	Norway	Western Europe	NaN	2015	4	7.522	0.88521	0.36503	1.33095	1.45900	0.66973	0.34699	2.46531	0.03880	NaN	NaN
4	Canada	North America	NaN	2015	5	7.427	0.90563	0.32957	1.32261	1.32629	0.63297	0.45811	2.45176	0.03553	NaN	NaN

2.6 Ordenació de files

Per tal de seguir un criteri en l'ordre de les files, escollim fer-ho en les següents columnes i en ordre de prioritat: Year, Happiness Score i Country.

[13]:
```
df.sort_values(['Year', 'Happiness Score', 'Country'], ascending=(True, False,␣
↪True), inplace=True)
df.head()
```

[13]:

	Country	Region	Country Or Region	Year	Happiness Rank	Happiness Score	Health	Trust	Family	Economy	Freedom	Generosity	Dystopia Residual	Standard Error	Lower Confidence Interval	Upper Confidence Interval
0	Switzerland	Western Europe	NaN	2015	1	7.587	0.94143	0.41978	1.34951	1.39651	0.66557	0.29678	2.51738	0.03411	NaN	NaN
1	Iceland	Western Europe	NaN	2015	2	7.561	0.94784	0.14145	1.40223	1.30232	0.62877	0.43630	2.70201	0.04884	NaN	NaN
2	Denmark	Western Europe	NaN	2015	3	7.527	0.87464	0.48357	1.36058	1.32548	0.64938	0.34139	2.49204	0.03328	NaN	NaN
3	Norway	Western Europe	NaN	2015	4	7.522	0.88521	0.36503	1.33095	1.45900	0.66973	0.34699	2.46531	0.03880	NaN	NaN
4	Canada	North America	NaN	2015	5	7.427	0.90563	0.32957	1.32261	1.32629	0.63297	0.45811	2.45176	0.03553	NaN	NaN

```
   Happiness Score   Health    Trust   Family  Economy  Freedom  Generosity  \
0            7.587  0.94143  0.41978  1.34951  1.39651  0.66557     0.29678
1            7.561  0.94784  0.14145  1.40223  1.30232  0.62877     0.43630
2            7.527  0.87464  0.48357  1.36058  1.32548  0.64938     0.34139
3            7.522  0.88521  0.36503  1.33095  1.45900  0.66973     0.34699
4            7.427  0.90563  0.32957  1.32261  1.32629  0.63297     0.45811

   Dystopia Residual  Standard Error  Lower Confidence Interval  \
0            2.51738         0.03411                        NaN
1            2.70201         0.04884                        NaN
2            2.49204         0.03328                        NaN
3            2.46531         0.03880                        NaN
4            2.45176         0.03553                        NaN

   Upper Confidence Interval
0                        NaN
1                        NaN
2                        NaN
3                        NaN
4                        NaN
```

3 Neteja

Al llarg d'aquest apartat es tractaran els atributs i elements que continguin valors buits, zeros o extrems.

3.1 Buits

A continuació es mostren totes les columnes que contenen com a mínim un valor buit. En els posteriors subapartats les tractem, ja que sovint requereixen solucions diferents.

[293]:
```
# columnes que contenen valors buits
sorted(df.columns[df.isna().any()].tolist())
```

[293]:
```
['Country',
 'Country Or Region',
 'Dystopia Residual',
 'Lower Confidence Interval',
 'Region',
 'Standard Error',
 'Trust',
 'Upper Confidence Interval']
```

3.1.1 Standard Error

Observem que la columna "Standard Error" només es va registrar l'any 2015, implicant que el 80% dels elements del dataset tenen aquest atribut buit. Degut a la falta d'utilitat que això comporta, es decideix eliminar.

[17]:
```
print(f'Elements "Standard Error" buits: {round(len(df[df["Standard Error"].
↪isna()])*100 / len(df))}%')
```

```
Elements "Standard Error" buits: 80%
```

[18]:
```
# anys que contenen elements buits de Standard Error
df[df['Standard Error'].isna()]['Year'].unique()
```

[18]:
```
array([2016, 2017, 2018, 2019])
```

[19]:
```
df.drop('Standard Error', axis=1, inplace=True)
```

3.1.2 Intervals de confiança

Observem que les columnes "Lower Confidence Interval" i "Upper Confidence Interval" només es va registrar l'any 2016 i 2017, implicant que el 60% dels elements del dataset tenen aquest atribut buit. Degut a la falta d'utilitat que això comporta, es decideix eliminar.

[20]:
```
lower_conf_nas = len(df[df["Lower Confidence Interval"].isna()])*100 / len(df)
print(f'Elements "Lower Confidence Interval" buits: {round(lower_conf_nas)}%')

upper_conf_nas = len(df[df["Upper Confidence Interval"].isna()])*100 / len(df)
print(f'Elements "Upper Confidence Interval" buits: {round(upper_conf_nas)}%')
```

```
Elements "Lower Confidence Interval" buits: 60%
Elements "Upper Confidence Interval" buits: 60%
```

[22]:
```
# anys que contenen elements buits de Lower Confidence Interval
df[df['Lower Confidence Interval'].isna()]['Year'].unique()
```

[22]:
```
array([2015, 2018, 2019])
```

[23]:
```
# anys que contenen elements buits de Upper Confidence Interval
df[df['Upper Confidence Interval'].isna()]['Year'].unique()
```

[23]:
```
array([2015, 2018, 2019])
```

[24]:
```
df.drop(['Lower Confidence Interval', 'Upper Confidence Interval'], axis=1,␣
↪inplace=True)
```

3.1.3 Dystopia Residual

Observem que la columna "Dystopia Residual" està buida en un 40% dels elements, ja que no es va registrar durant els anys 2018 i 2019. Tot i que aquest atribut pot resultar interessant, com que un dels objectius principals és estudiar l'evolució al llarg dels anys de l'índex de felicitat, l'eliminem perquè durant els últims anys no es va generar le mesura i no ens permetria fer aquesta comparació.

[25]:
```
print(f'Elements "Dystopia Residual" buits: {round(len(df[df["Dystopia␣
↪Residual"].isna()])*100 / len(df))}%')
```

```
Elements "Dystopia Residual" buits: 40%
```

[26]:
```
# anys que contenen elements buits de Standard Error
df[df['Dystopia Residual'].isna()]['Year'].unique()
```

[26]:
```
array([2018, 2019])
```

[27]:
```
df.drop('Dystopia Residual', axis=1, inplace=True)
```

3.1.4 Country

Sabem que hi ha una clara relació entre les columnes "Country", "Region" i "Country Or Region". Sembla ser que els últims anys s'utilitzava la columna general "Country Or Region", mentre que durant els primers es preferia tenir-ho per separat.

Per tal d'assolir els objectius plantejats, en aquest cas es prefereix tenir les dues variables per separat, així que a continuació s'integra la columna "Country Or Region" dins de les altres dues: "Country" o "Region", segons el cas. Això ho fem iterant sobre les dades que no tenen buida la columna "Country Or Region" i assignant el seu valor a una de les dues columnes "Country" o "Region", segons si aquell valor ja estava inclòs prèviament en alguna de les dues variables.

[28]:
```
df[['Country', 'Region', 'Country Or Region']]
```

[28]:

	Country	Region	Country Or Region
0	Switzerland	Western Europe	NaN
1	Iceland	Western Europe	NaN
2	Denmark	Western Europe	NaN
3	Norway	Western Europe	NaN
4	Canada	North America	NaN
..	…	…	…
777	NaN	NaN	Rwanda
778	NaN	NaN	Tanzania
779	NaN	NaN	Afghanistan
780	NaN	NaN	Central African Republic
781	NaN	NaN	South Sudan

[782 rows x 3 columns]

[29]:
```
# assignació de Regió o Country a partir de la columna "Country Or Region"

countries_unique = df['Country'].unique()
regions_unique = df['Region'].unique()

for idx, ctry_or_reg in df[~df['Country Or Region'].isna()]['Country Or␣
↪Region'].items():
    if ctry_or_reg in countries_unique and pd.isnull(df.loc[idx, 'Country']):
        df.loc[idx, 'Country'] = ctry_or_reg
    elif ctry_or_reg in regions_unique and pd.isnull(df.loc[idx, 'Region']):
        df.loc[idx, 'Region'] = ctry_or_reg
```

Ara que ja hem passat els països o regions a les seves respectives columnes separades, obsevem que hi ha 6 instàncies que no s'han pogut classificar correctament. Això és perquè els seus respectius "Country Or Region" no estaven inclosos en cap de les dues columnes ("Country" o "Region") i no hem pogut comprovar a quina pertanyen. Com que són pocs elements, acabem de completar-ho a mà.

Sabem que Trinidad & Tobago, North macedonia i Gambia són països, mentre que Northern Cyprus és una regió que pertany a la República de Xipre (només Turquia ho reconeix com a país), però per algun motiu en aquest estudi ho han considerat com a país en altres entrades, així que també ho considerarem com a tal.

[30]:
```
df[df['Country'].isna()]
```

[30]:

	Country	Region	Country Or Region	Year	Happiness Rank	Happiness Score	\
507	NaN	NaN	Trinidad & Tobago	2018	38	6.192	
527	NaN	NaN	Northern Cyprus	2018	58	5.835	
664	NaN	NaN	Trinidad & Tobago	2019	39	6.192	
689	NaN	NaN	Northern Cyprus	2019	64	5.718	
709	NaN	NaN	North Macedonia	2019	84	5.274	
745	NaN	NaN	Gambia	2019	120	4.516	

	Health	Trust	Family	Economy	Freedom	Generosity
507	0.564	0.019	1.492	1.223	0.575	0.171
527	0.909	0.154	1.211	1.229	0.495	0.179
664	0.713	0.016	1.477	1.231	0.489	0.185
689	1.042	0.162	1.252	1.263	0.417	0.191
709	0.838	0.034	1.294	0.983	0.345	0.185
745	0.428	0.167	0.939	0.308	0.382	0.269

[31]:
```
df.loc[df['Country'].isna(), 'Country'] = df.loc[df['Country'].isna(), 'Country␣
↪Or Region']
```

Un cop feta aquesta integració de l'atribut "Country Or Region" a les columnes "Country" i "Region", podem eliminar-la del dataset, ja que la informació que conté és redundant.

[32]:
```
df.drop('Country Or Region', axis=1, inplace=True)
```

3.1.5 Region

Tot i la transformació realtizada anteriorment, veiem que la columna "Region" segueix tenint molts elements buits, consistint en un total del 60% de les dades. Això és perquè sembla que només es va registrar aquest atribut el 2015 i 2016:

[33]:
```
print(f'Elements "Region" buits: {round(len(df[df["Region"].isna()])*100 /␣
↪len(df))}%')
```

```
Elements "Region" buits: 60%
```

[34]:
```
# anys que contenen elements buits a Region
df[df['Region'].isna()]['Year'].unique()
```

[34]:
```
array([2017, 2018, 2019])
```

Com que és un atribut que ens pot resultar útil d'analitzar per veure diferències entre diferents regions, es decideix mantenir-lo i omplir els elements en els quals és buit. Per fer-ho, només hem de buscar a quina regió s'havia assignat aquell país en anys anteriors. En cas que no s'hagués registrat aquesta informació anteriorment i no puguem determinar a quina regió pertany, li assignarem el valor "NI" (No Identificat) i ho tractarem posteriorment en aquest apartat.

[35]:
```
def fill_region_by_group_country(group):
    # funció que retorna la regió a la qual pertany el país del grup segons␣
↪s'hagués classificat anteriorment
    non_na_region = group[~group['Region'].isna()]
    if non_na_region.empty:
        return group['Region'].fillna('NI')
    return group['Region'].fillna(group[~group['Region'].isna()].
↪iloc[0]['Region'])
```

[36]:
```
impute_region = df.groupby('Country', as_index=False)\
                  .apply(fill_region_by_group_country)\
                  .reset_index(level=0, drop=True)
```

[37]:
```
df.loc[impute_region.index, 'Region'] = impute_region.values
```

Després de fer l'assignació de regions, veiem que no hi ha gaires elements que no hagin pogut quedar identificats, així que podem completar a mà les regions d'aquests:

[38]:
```
df[df['Region'] == 'NI']
```

[38]:

	Country	Region	Year	Happiness Rank	Happiness Score	\
347	Taiwan Province of China	NI	2017	33	6.422	
385	Hong Kong S.A.R., China	NI	2017	71	5.472	
507	Trinidad & Tobago	NI	2018	38	6.192	
527	Northern Cyprus	NI	2018	58	5.835	
664	Trinidad & Tobago	NI	2019	39	6.192	
689	Northern Cyprus	NI	2019	64	5.718	

```
709           North Macedonia     NI  2019              84           5.274
745                    Gambia     NI  2019             120           4.516

       Health     Trust    Family   Economy   Freedom  Generosity
347  0.793984  0.063829  1.384565  1.433627  0.361467    0.258360
385  0.943062  0.293934  1.262791  1.551675  0.490969    0.374466
507  0.564000  0.019000  1.492000  1.223000  0.575000    0.171000
527  0.909000  0.154000  1.211000  1.229000  0.495000    0.179000
664  0.713000  0.016000  1.477000  1.231000  0.489000    0.185000
689  1.042000  0.162000  1.252000  1.263000  0.417000    0.191000
709  0.838000  0.034000  1.294000  0.983000  0.345000    0.185000
745  0.428000  0.167000  0.939000  0.308000  0.382000    0.269000
```

[39]:
```
# regió de Taiwan
df.loc[df['Country'] == 'Taiwan Province of China', 'Region'] = 'Eastern Asia'

# regió de Hong Kong
df.loc[df['Country'] == 'Hong Kong S.A.R., China', 'Region'] = 'Eastern Asia'

# regió de Trinidad & Tobago
df.loc[df['Country'] == 'Trinidad & Tobago', 'Region'] = 'Latin America and␣
 ↪Caribbean'

# regió de Northern Cyprus
df.loc[df['Country'] == 'Northern Cyprus', 'Region'] = 'Western Europe'

# regió de North Macedonia
df.loc[df['Country'] == 'North Macedonia', 'Region'] = 'Central and Eastern␣
 ↪Europe'

# regió de Gambia
df.loc[df['Country'] == 'Gambia', 'Region'] = 'Sub-Saharan Africa'
```

3.1.6 Trust

A continuació observem que només tenim un element amb "Trust" buit, consistent en l'entrada de UAE de l'any 2018. Veiem també que els nivells de confiança en el govern del país anaven descendint substancialment any rere any, tenint el 2015 un valor de 0,39 i decreixent any rere any fins arribar a un valor de 0,18 l'any 2019.

Podria no ser bo haver d'eliminar aquesta entrada i perdre altra informació útil degut a un únic valor buit, per això es decideix fer un càlcul per aquest valor. Probablement es podria assumir que el ritme descendent que mostrava la dada es va complir també el 2018 i podríem calcular el valor mig entre l'any anterior (2017) i el posterior (2019). Això faria que el seu valor fos de 0,182 + (0,324 - 0,182) / 2 = 0,253.

Tot i que aquest càlcul podria ser bo, el més correcte estadísticament és, probablement, fer una mitjana dels veïns més propers en relació a les altres variables, suposant que aquestes són capaces

de “predir” un valor aproximat per a la dada buida.

[43]:
```
df[df['Trust'].isna()]
```

[43]:
```
                  Country                              Region  Year  \
489  United Arab Emirates  Middle East and Northern Africa  2018

     Happiness Rank  Happiness Score  Health  Trust  Family  Economy  Freedom  \
489              20            6.774    0.67    NaN   0.776    2.096    0.284

     Generosity
489       0.186
```

[44]:
```
df[df['Country'] == 'United Arab Emirates'][['Year', 'Trust']]
```

[44]:
```
     Year    Trust
19   2015  0.38583
185  2016  0.35561
335  2017  0.32449
489  2018      NaN
646  2019  0.18200
```

Per fer una imputació segons el valor dels veïns més propers, s'utilitza kNN (KNNImputer). Agafem 20 veïns i ponderem la mitjana segons la distància d'aquests als valors de UAE de 2018, o sigui que com més propers els valors, més pes tindran en la mitjana.

Després d'executar la imputació de dades buides, veiem que s'obté un valor de “Trust” de 0,32 per a l'any 2018, fet que concorda amb la observació de valor decreixent que havíem vist abans, encara que potser hauríem esperat un valor més baix per tal que no hi hagués un salt tan gran entre 2018 i 2019. Tot i això, el valor sembla completament realista i el donarem per vàlid.

[84]:
```
from sklearn.impute import KNNImputer

imputer = KNNImputer(n_neighbors=20, weights='distance')
# entrenem i transformem les dades i agafem la fila corresponent a UAE 2018
imputed_row = imputer.fit_transform(df[['Family', 'Health', 'Generosity',␣
 ↪'Freedom', 'Trust', 'Economy']])[489]
```

[85]:
```
print('Valor de "Trust" imputat:', imputed_row[-2])
```

```
Valor de "Trust" imputat: 0.3203249585035027
```

[86]:
```
# assignem el valor imputat
df.loc[df['Trust'].isna(), 'Trust'] = imputed_row[-2]
```

3.1.7 Comprovació final

Comprovem que després de tota la neteja no queda ni una sola columna que contingui valors buits dins el nostre dataset:

[88]:
```
# columnes que contenen valors buits
sorted(df.columns[df.isna().any()].tolist())
```

[88]:
```
[]
```

3.2 Zeros

A continuació veiem el nombre d'elements amb un valor de zero que tenen les diferents columnes. Podria o no tractar-se d'errors, ja que encara que zero és un valor molt baix, és totalment factible que un país rebi aquesta puntuació, depenent del que hagin contestat els enquestats.

S'ha verificat un a un cadascun dels casos i, en tots ells, sembla un valor totalment vàlid. Per exemple, veiem més avall una evolució dels valors de "Trust" a Bòsnia; tal com podem observar, generalment té valors molt baixos, així que no és estrany que en alguns d'aquests anys (2016, 2017 i 2018) tingués valors de zero, a part que sabem que és una regió que ha tingut molts problemes en aquest i d'altres sentits. Una cosa semblant passa amb Somalia i l'economia del país, que rep puntuacions de zero el 2016, 2018 i 2019, i una puntuació molt baixa l'any 2017, tal i com podem veure més avall.

Amb la comprovació individual i manual que s'ha fet de cadascun d'aquests casos, el raonament és similar. Així doncs, deixem intactes tots els valors de zero que trobem en el dataset, ja que semblen ser dades plenament vàlides.

[54]:
```
(df == 0).sum(axis=0).sort_values(ascending=False)
```

[54]:
```
Trust              6
Freedom            5
Family             5
Generosity         5
Economy            5
Health             5
Happiness Score    0
Happiness Rank     0
Year               0
Region             0
Country            0
dtype: int64
```

[89]:
```
# evolució de "Trust" a Bòsnia i Herzegovina
df[df['Country'] == 'Bosnia and Herzegovina'][['Country', 'Year', 'Trust']]
```

[89]:

	Country	Year	Trust
95	Bosnia and Herzegovina	2015	0.00227
244	Bosnia and Herzegovina	2016	0.00000
404	Bosnia and Herzegovina	2017	0.00000
562	Bosnia and Herzegovina	2018	0.00000
703	Bosnia and Herzegovina	2019	0.00600

[90]:
```
df[df['Country'] == 'Somalia'][['Country', 'Year', 'Economy']]
```

[90]:

	Country	Year	Economy
233	Somalia	2016	0.000000
407	Somalia	2017	0.022643
567	Somalia	2018	0.000000
737	Somalia	2019	0.000000

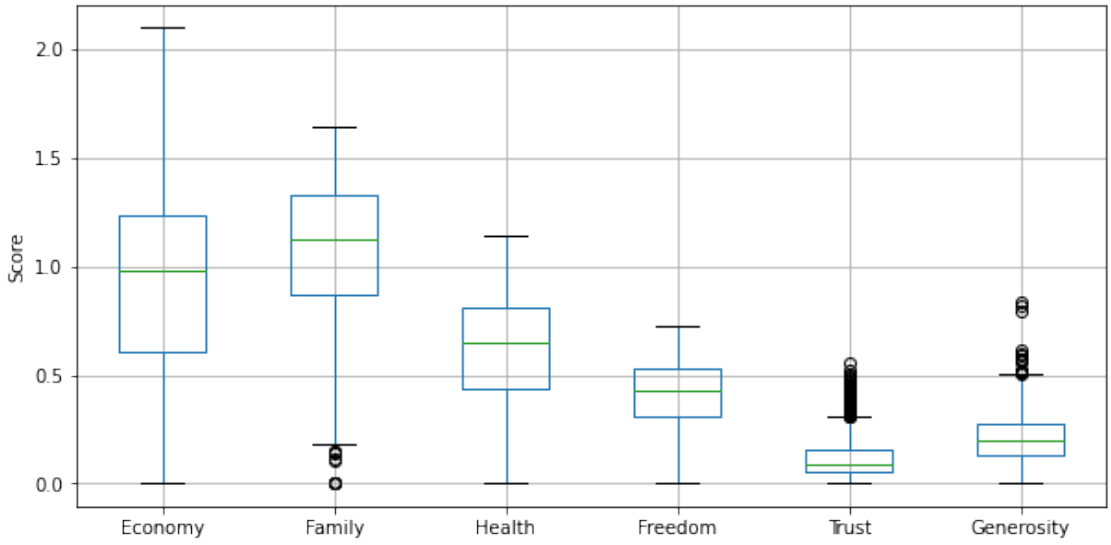
3.3 Extrems

Pel que fa a casos extrems, en els següents boxplots s'observen les respectives distribucions de les diferents variables. Veiem que, tot i senyalar-se alguns punts com a "valors llunyans" (a més de 1,5 vegades els corresponents extrems de l'amplitud interquartílica [8]), aquests no semblen estar fora dels rangs esperats. Per exemple, veiem alguns valors més alts en "Trust" o "Generosity", però tots tenen complet sentit i no semblen gens inversemblatns. El mateix passa amb els valors baixos senyalats a "Family", alguns d'ells ja s'han debatut a l'apartat 3.2. Tant la resta d'atributs com el "Happiness Score" semblen també perfectament dins de rangs vàlids.

Així doncs, en aquest apartat no es fa cap tractament específic sobre valors extrems, ja que es considera que no hi ha errors i que tots formen part de la mostra.

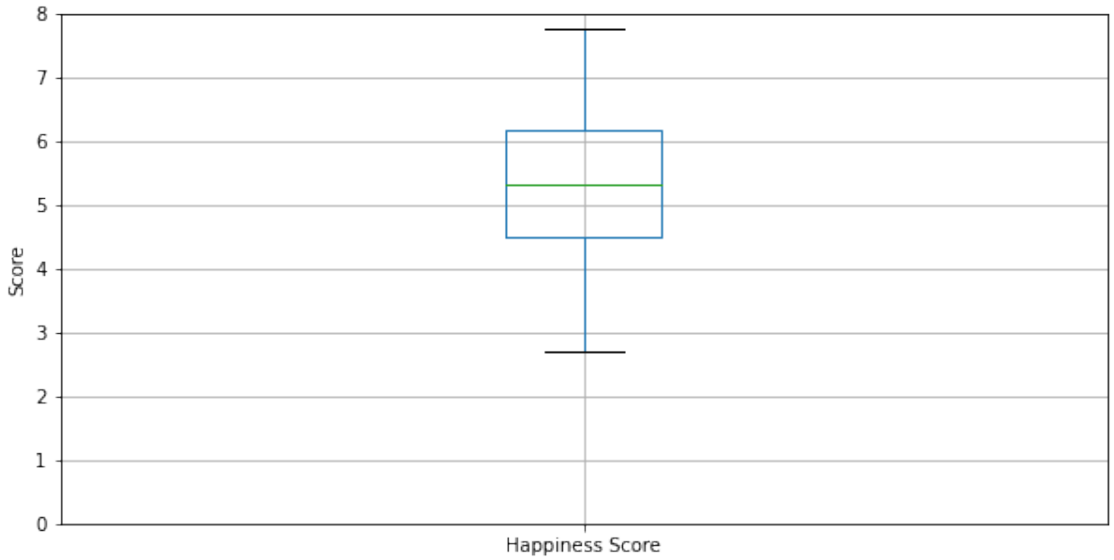
[91]:
```
plt.rcParams['figure.figsize'] = [10, 5]

df[['Economy', 'Family', 'Health', 'Freedom', 'Trust', 'Generosity']].boxplot()
plt.ylabel('Score')
plt.show()
```



[92]:
```
df[['Happiness Score']].boxplot()
plt.ylabel('Score')
```

```
plt.ylim(0, 8)
plt.show()
```



3.4 Exportació de dades

Un cop realitzada tota la integració, selecció i neteja de dades, procedim a guardar-los en un nou fitxer CSV anomenat "world_happiness_2015-2019". Observem que finalment tenim les 782 files que ja havíem observat inicialment, però hem passat de 16 a 11 columnes.

[98]:
```
df.shape
```

[98]:
```
(782, 11)
```

[94]:
```
df.to_csv('data/world_happiness_2015-2019.csv', index=False)
```

4 Anàlisi

[97]:
```
# càrrega del fitxer de dades net
df = pd.read_csv('data/world_happiness_2015-2019.csv')
df.head()
```

[97]:

	Country	Region	Year	Happiness Rank	Happiness Score	\
0	Switzerland	Western Europe	2015	1	7.587	
1	Iceland	Western Europe	2015	2	7.561	
2	Denmark	Western Europe	2015	3	7.527	
3	Norway	Western Europe	2015	4	7.522	
4	Canada	North America	2015	5	7.427	

	Health	Trust	Family	Economy	Freedom	Generosity
0	0.94143	0.41978	1.34951	1.39651	0.66557	0.29678
1	0.94784	0.14145	1.40223	1.30232	0.62877	0.43630
2	0.87464	0.48357	1.36058	1.32548	0.64938	0.34139
3	0.88521	0.36503	1.33095	1.45900	0.66973	0.34699
4	0.90563	0.32957	1.32261	1.32629	0.63297	0.45811

4.1 Selecció de grups i planificació

Per tal de donar resposta a les preguntes plantejades a l'apartat 1.2, es creen els següents grups per tal d'analitzar-los amb més comoditat:

- **region_gb_2019**: Puntuacions de felicitat del 2019 agrupades per regions.
- **happiness_diff_2015_2019**: Càlcul de la diferència de valor de felicitat entre els anys 2015 i 2019 en tots els països.
- **western_europe_2019**: Dades de la regió Western Europe del 2019.
- **world_2019**: Dades de tots els països del món del 2019.
- **world_lt_2019**: Dades de tots els països del món anteriors a 2019 (no inclòs).
- **spain_df**: Dades d'Espanya en tots els anys.

[ ]:
```
# puntuacions de felicitat del 2019 agrupades per regions
region_gb_2019 = df[df['Year'] == 2019].groupby('Region')
region_happiness_2019 = {region: group['Happiness Score'] for region, group in␣
↪region_gb_2019}
```

[113]:
```
def happiness_diff(group, year1, year2):
    # càlcul de la diferència de valor de felicitat entre year2 i year1
    years = group['Year'].unique()
    if not (year1 in years and year2 in years):
        return None
    is_2015 = group['Year'] == year1
    is_2019 = group['Year'] == year2
    return group[is_2019].iloc[0]['Happiness Score'] - group[is_2015].
↪iloc[0]['Happiness Score']

# diferència de valor de felicitat entre els anys 2015 i 2019
happiness_diff_2015_2019 = df.groupby('Country').apply(lambda g:␣
↪happiness_diff(g, 2015, 2019))
```

[115]:
```
# dades western europe 2019
western_europe_2019 = df[(df['Year'] == 2019) & (df['Region'] == 'Western␣
↪Europe')]
```

[116]:
```
# dades mundials 2019
world_2019 = df[df['Year'] == 2019]
# dades mundials abans del 2019
world_lt_2019 = df[df['Year'] < 2019]
```

[212]:
```
# dades d'Espanya en tots els anys.
spain_df = df[df['Country'] == 'Spain']
```

4.2 Normalitat

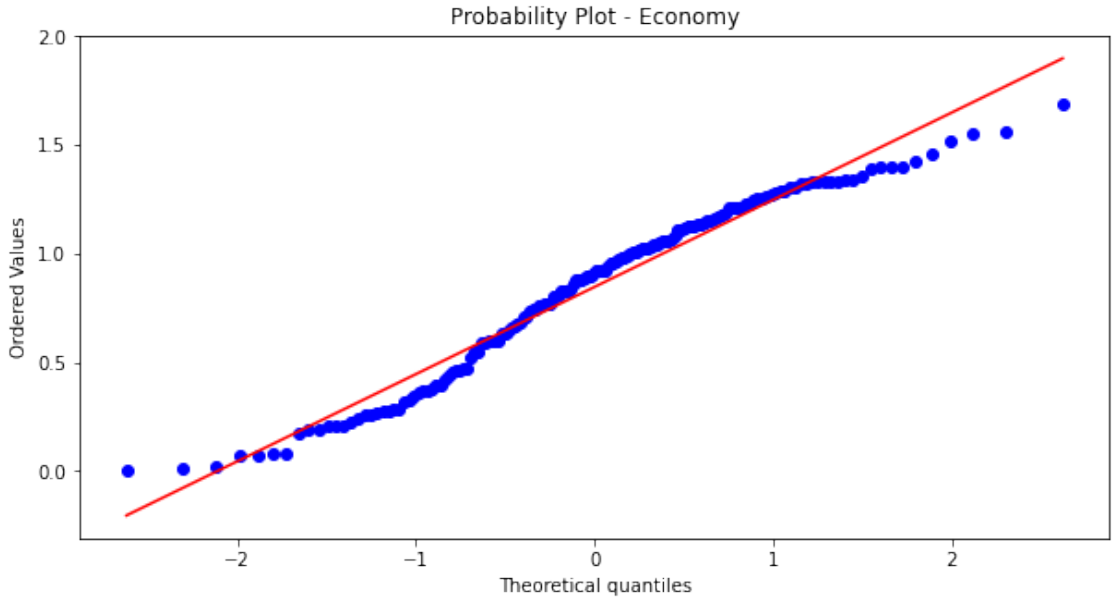
[84]:
```
import scipy.stats as stats
```

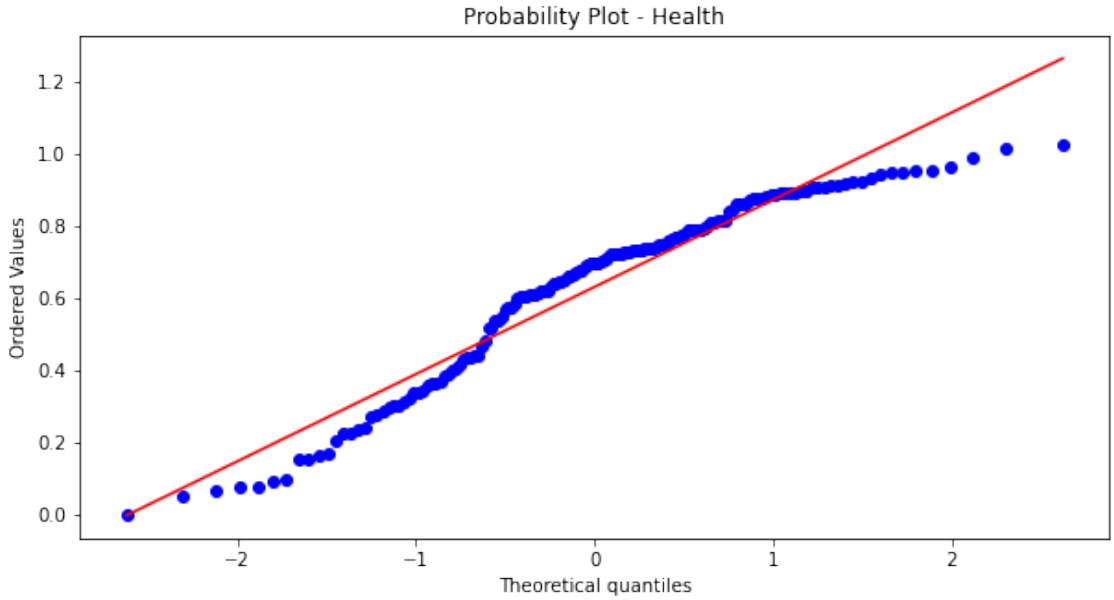
Per tal de tenir una idea visual sobre la normalitat de les dades, fem un QQ plot del 2015 sobre tots els indicadors. Posteriorment ja farem els tests estadístics pertinents per comprovar la seva normalitat.

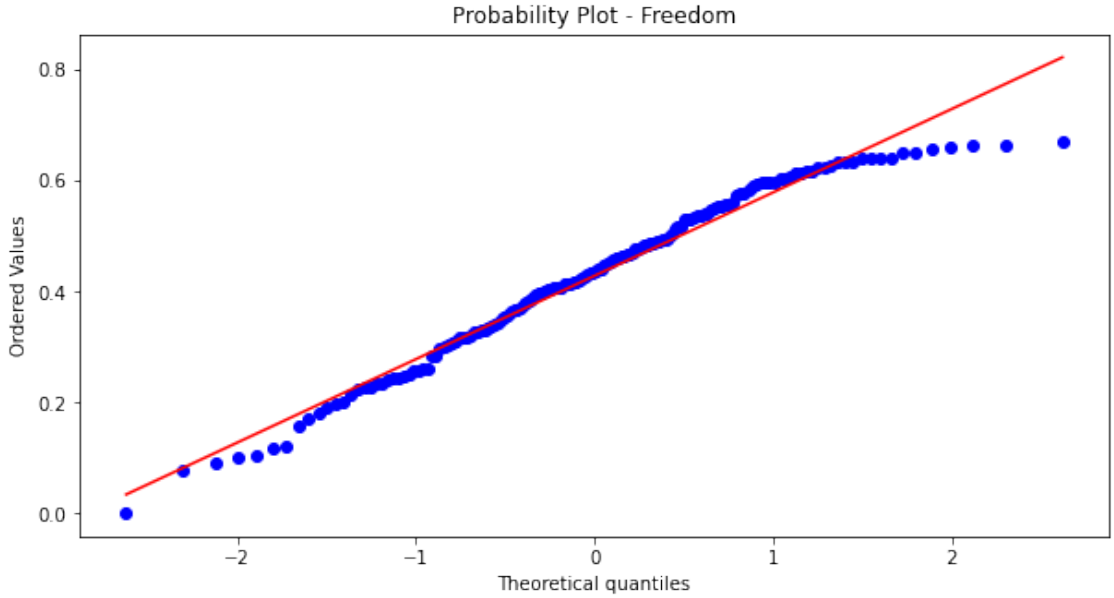
Veient les desviacions que tenim de les dades quan ho comparem amb les que serien propies d'una normal (línia vermella), podem dir que sembla poc probable que es tractin de distribucions normals. Per extensió, probablement la resta d'anys posteriors a 2015 mostrin patrons similars.

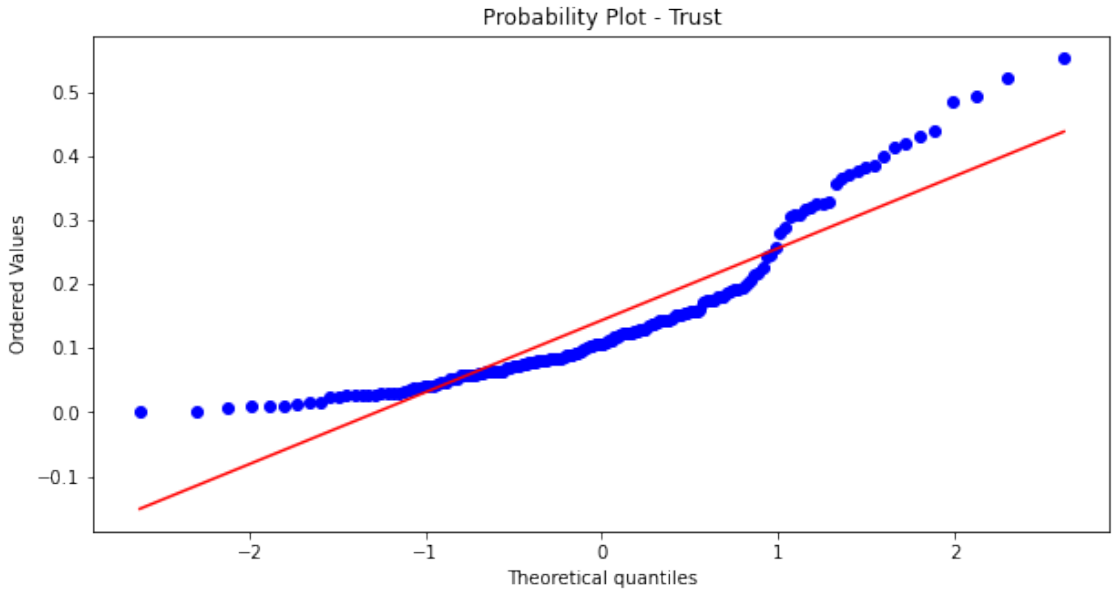
[85]:
```
ft_cols = ['Economy', 'Health', 'Freedom', 'Trust', 'Generosity', 'Family']
```

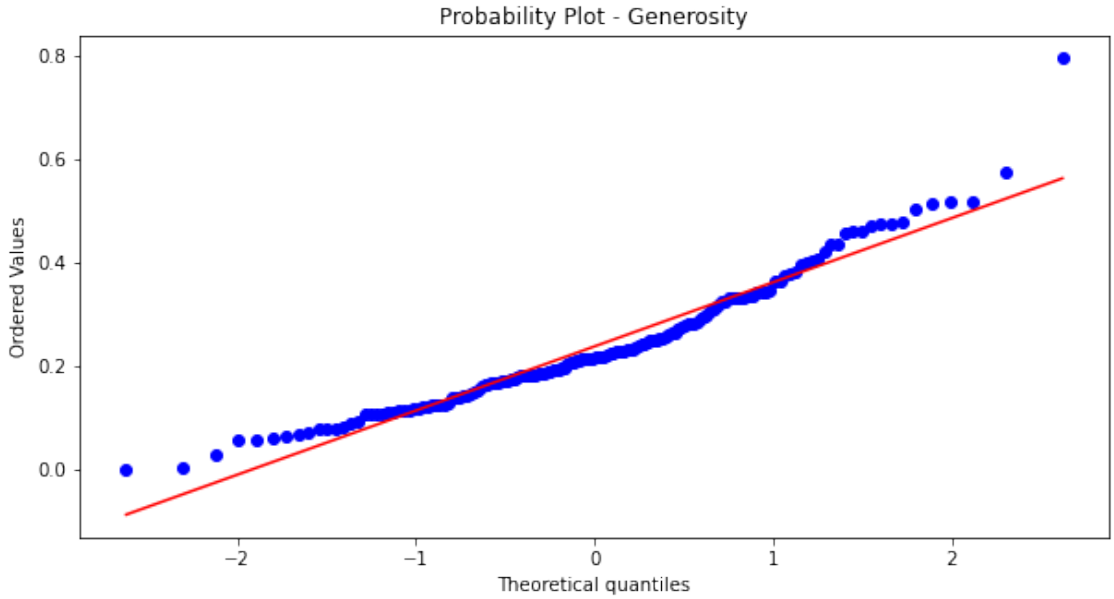
[86]:
```
for col in ft_cols:
    stats.probplot(df[df['Year'] == 2015][col], dist='norm', plot=plt)
    plt.title(f'Probability Plot - {col}')
    plt.show()
```

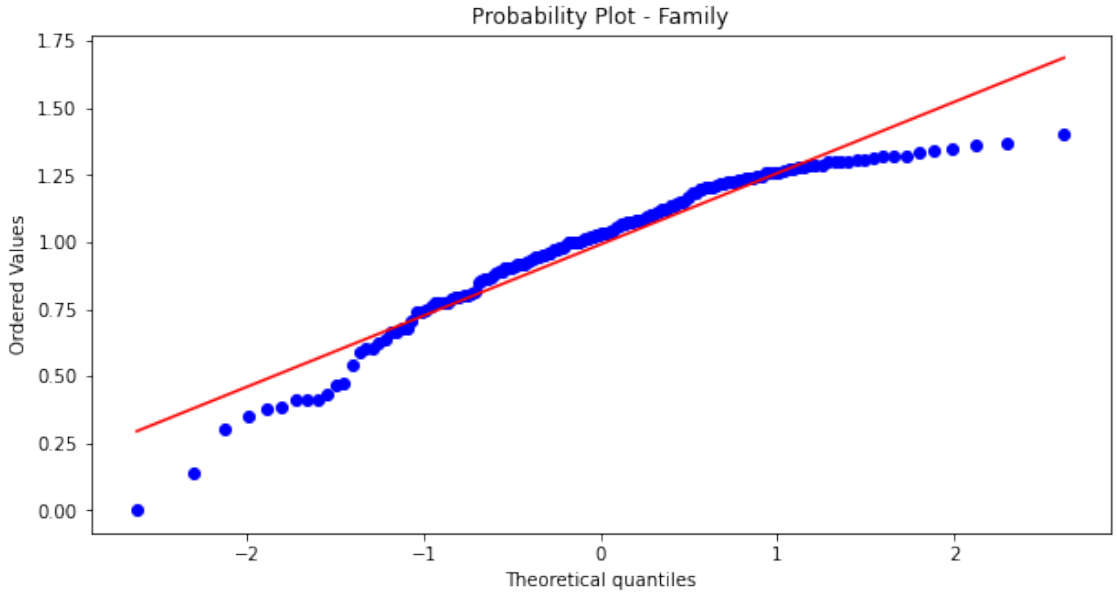



Probability Plot - Health
Ordered Values
Theoretical quantiles
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−2
−1
0
1
2


Probability Plot - Freedom
Ordered Values
Theoretical quantiles
0.8
0.6
0.4
0.2
0.0
−2
−1
0
1
2


Probability Plot - Trust
Ordered Values
Theoretical quantiles
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−2
−1
0
1
2


Probability Plot - Generosity
Ordered Values
Theoretical quantiles
0.8
0.6
0.4
0.2
0.0
−2
−1
0
1
2



Per a testejar la normalitat dels diferents atributs en tots els anys, utilitzem el test de Shapiro-Wilk, ja que és un test no-paramètric (així ens estalviem de fer les comprovacions pertinents de pertanyença a una distribució concreta abans de fer el test) i és un dels més potents i acurats en donar resultats de significància segons unes simulacions de MonteCarlo [9].

La hipòtesi nul · la del test és que la població està normalment distribuïda.

Per a fer els tests, ens centrarem en evaluar cadascun dels anys per separat per comprovar si les variables estan o no distribuïdes normalment en aquell any concret.

A continuació es mostren els p-values obtinguts per cada any i atribut. S'han arrodonit els valors a 3 decimals, així que si els p-values són molt petits es mostren com a 0.

Tal i com podem comprovar, tots els valors són molt baixos, implicant que rebutgem la hipòtesi nul · la a favor de l'alternativa, que diu que les dades no provenen de distribucions normalment distribuïdes.

[87]:
```
from scipy.stats import shapiro
```

[88]:
```
print('Shapiro-Wilk Test')
print()
for year in df['Year'].unique():
    print(year)
    print('----------------------')
    for col in ft_cols:
        _, p = shapiro(df[df['Year'] == year][col])
        print(f'{col} --> p-value:{round(p, 3)}')
    print()
```

```
Shapiro-Wilk Test
```

```
2015
----------------------
Economy --> p-value:0.0
Health --> p-value:0.0
Freedom --> p-value:0.002
Trust --> p-value:0.0
Generosity --> p-value:0.0
Family --> p-value:0.0

2016
----------------------
Economy --> p-value:0.001
Health --> p-value:0.0
Freedom --> p-value:0.0
Trust --> p-value:0.0
Generosity --> p-value:0.0
Family --> p-value:0.0

2017
----------------------
Economy --> p-value:0.002
Health --> p-value:0.0
Freedom --> p-value:0.0
Trust --> p-value:0.0
Generosity --> p-value:0.0
Family --> p-value:0.0

2018
----------------------
Economy --> p-value:0.015
Health --> p-value:0.0
Freedom --> p-value:0.0
Trust --> p-value:0.0
Generosity --> p-value:0.0
Family --> p-value:0.0

2019
----------------------
Economy --> p-value:0.001
Health --> p-value:0.0
Freedom --> p-value:0.0
Trust --> p-value:0.0
Generosity --> p-value:0.0
Family --> p-value:0.0
```

4.3 Homogeneïtat de la variància

Per comprovar la homogeneïtat de la variància utilitzem el test de Levene, que és un dels més utilitzats en aquest àmbit. Com a mesures de centre utilitzem les medianes, ja que tenim distribucions asimètriques i així els resultats del test seran més robusts.

Comprovem la homogeneïtat de variàncies entre les puntuacions de felicitat de les diferents regions del planeta amb les dades de 2019.

Veiem que obtenim un p-value de 0,13 que, segons aquest test, implica que no sembla especialment probable que les diferències del mostreig siguin de poblacions amb variàncies diferents, és a dir, sembla que provenen de poblacions amb variàncies iguals.

[90]:
```
from scipy.stats import levene
```

[91]:
```
levene(*region_happiness_2019.values(), center='median')
```

[91]:
```
LeveneResult(statistic=1.552963242230361, pvalue=0.134824421577903)
```

4.4 Proves estadístiques

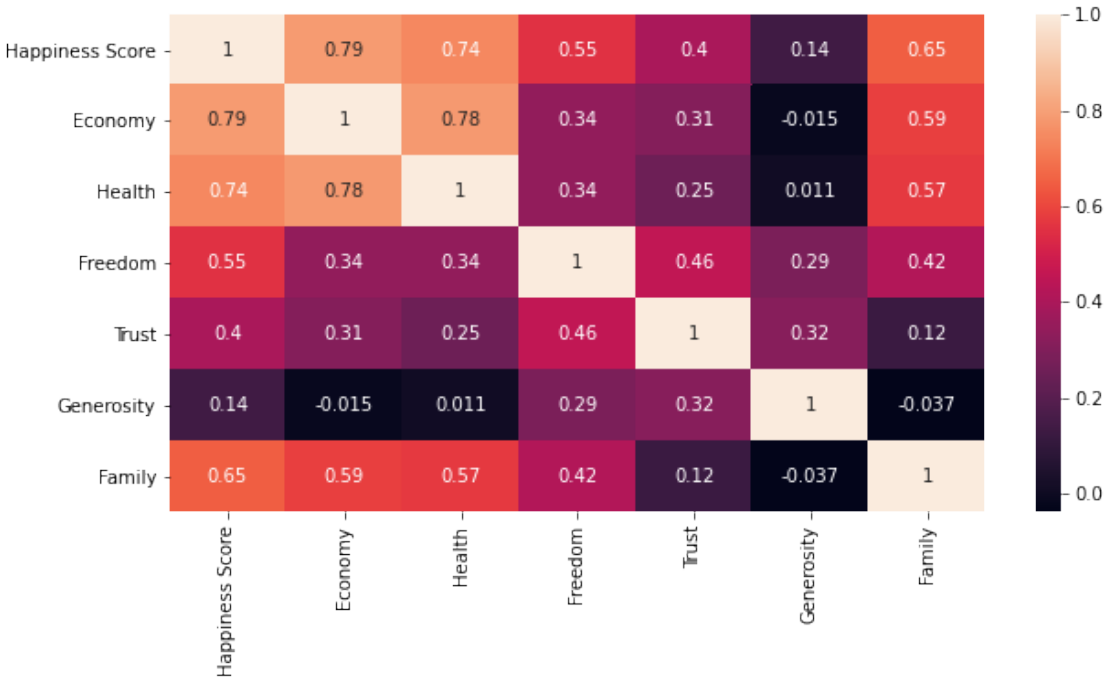
4.4.1 Correlació

Per entendre la relació que poden tenir les diferents variables amb la puntuació de felicitat o entre elles, en mostrem la matriu de correlació amb comparacions de correlació de Pearson dos a dos.

Podem comprovar que hi ha dos atributs amb una correlació relativament alta amb el "Happiness Score", essent aquests l'economia i la salut, amb uns valors de correlació de 0,79 i 0,74 respectivament. Té sentit pensar que aquells països amb una millor economia o un millor sistema de salut que et permet viure més anys també tindran uns habitants més feliços, així que no és d'estranyar haver trobat aquesta relació.
En aqueta línia, veiem també que l'economia i la salut tenen una bona correlació entre elles, amb un valor de 0,78, probablement indicant que països amb una millor economia també disposaran de millors sistemes i medis per cuidar la salut de la població.

Per altra banda, potser sorprèn la poca correlació que tenen en general les variables "Generosity" i "Trust", tan amb el "Happiness Score" com amb la resta d'atributs. Per exemple, es podria pensar que com més confien els països en els seus governs també se senten més feliços o anb més llibertat, fets que no semblen ser el cas (tot i que es requeriria un anàlisi més profund). També, veiem com la correlació de "Generosity" amb "Economy", "Health" o "Family" és pràcticament inexistent, fet que sorprèn en certa manera perquè potser s'esperaria que països amb més cohesió familiar també tinguin una major percepció de generositat, per exemple.

[102]:
```
corr_cols = ['Happiness Score', 'Economy', 'Health', 'Freedom', 'Trust',␣
↪'Generosity', 'Family']
sns.heatmap(df[corr_cols].corr(), annot=True)
plt.show()
```



4.4.2 Predicció i importància dels atributs

En aquest apartat ens centrarem a comprovar si és possible predir el "Happiness Score" a partir dels sis indicadors escollits. Per fer-ho, entrenarem els models amb les dades de 2015 a 2018 (conjunt d'entrenament), i a mesurar-ne la precisió amb les dades de 2019, que serà el nostre conjunt de test.

Comencem carregant les llibreries necessàries pel posterior anàlisi i a escollir les features utilitzades per integrar els conjunts d'entrenament i test.

[145]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error
from sklearn.linear_model import LinearRegression
from sklearn.decomposition import PCA
```

[144]:
```
ft_cols = ['Economy', 'Health', 'Freedom', 'Trust', 'Generosity', 'Family']
tg_col = 'Happiness Score'
X_train, y_train = world_lt_2019[ft_cols], world_lt_2019[tg_col]
X_test, y_test = world_2019[ft_cols], world_2019[tg_col]
```

A continuació s'entrena un model Random Forest, ja que és un mètode que permet fer regressions no lineals (si aquest fos el cas de les nostres dades). A més a més, és dels mètodes que ens dóna, generalment, més fiabilitat amb menys condicions en quan als seus hiperparàmetres (fer una cerca dels millor paràmetres del model implicaria introduir un conjunt de validació, afegint una complexitat que probablement no és necessària per a l'anàlisi exploratiu que s'està realitzant).

Així doncs, definim un model de Random Forest que tindrà els atributs per defecte, tals com 100 arbres estimadors o una profunditat màxima indefinida, ja que al tractar-se d'un Random Forest que s'entrenarà amb pocs atributs, probablement no arribi a sobreentrenar-se en excés.

Comparant les prediccions de "Happiness Score" del model amb les dades reals de l'any 2019, veiem que tenim un error absolut mitjà (MAE, per les seves sigles en anglès) de 0,5. Tenint en compte que els valors de la "Happiness Score" van de 3 a 8, tal i com es pot veure al boxplot de l'apartat 3.3, de bones a primeres el podem considerar un resultat força bo i, per tant, sembla que els sis indicadors escollits són prou útils per deduir la felicitat percebuda per la població d'un país.
Si ens fixem en tota la distribució d'errors absoluts, veiem que hi ha algun cas concret en el qual s'ha fallat per bastant, ja que el màxim ha estat amb un erro de més de 2 punts. No obstant, amb el 75% de les dades l'error ha estat de menys de 0,7 punts, fet que sembla força positiu en quan a la capacitat predictiva del model.

[159]:
```
# model Random Forest
rf = RandomForestRegressor(100, n_jobs=-1)
# entrenament
rf.fit(X_train, y_train)
# predicció del 2019
y_pred_rf = rf.predict(X_test)
# error absolut mitjà
print('MAE:', mean_absolute_error(y_test, y_pred_rf))
```

```
MAE: 0.4967255126834303
```

[202]:
```
# distribució dels errors absoluts
(abs(y_test - y_pred_rf)).describe()
```

[202]:
```
count    156.000000
mean       0.494673
std        0.388414
min        0.002950
25%        0.202016
50%        0.406960
75%        0.699007
max        2.159365
Name: Happiness Score, dtype: float64
```

Un dels motius pels quals abans hem decidit utilitzar un Random Forest és perquè ens permet realitzar regressions no lineals. Ara bé, és aquest el cas de les nostres dades?

Si provem de fer les mateixes prediccions amb un model de Regressió Lineal, veiem que el MAE obtingut ha estat de 0,42, significativament millor que el Random Forest utiltizat anteriorment. Així doncs, sembla que les nostres dades segueixen una tendència força lineal pel que fa a la predicció i explicació del "Happiness Score".

[160]:
```
# model Regressió lineal
lr = LinearRegression()
# entrenament
```

```
lr.fit(X_train, y_train)
# predicció del 2019
y_pred_lr = lr.predict(X_test)
# error absolut mitjà
print('MAE:', mean_absolute_error(y_test, y_pred_lr))
```

```
MAE: 0.42408836977003733
```

Com és que la regressió lineal ho ha fet força millor que el Random Forest si aquest últim també és capaç de realitzar regressions que siguin lineals (a part de no lineals)?

Com que tampoc tenim molts atributs, una possible explicació és que la colinealitat que han mostrat alguns dels atributs (veure apartat 4.4.1) l'està afectant negativament.
A continuació provem de realitzar un PCA amb 5 components (per no reduir massa el nombre de features), ja que això farà que els nous atributs (Principal Components) siguin ortogonals entre sí, eliminant tota colinealitat. A part, com que els Principal Components van en la direcció de màxima variància, probablement ajudi els arbres de decisió a trobar fàcilment el "split" que ofereix més guany d'informació, millorant-ne així el rendiment.

Veiem a continuació que del Random Forest entrenat i testejat amb la reducció PCA obté un MAE de 0,38, substancialment millor que el de la regressió lineal i, per tant, essent el model amb un millor rendiment dels 3 amb els quals hem experimentat.

[149]:
```
# apliquem un PCA amb 5 components
pca = PCA(n_components=5)
pca_train = pca.fit_transform(world_lt_2019[ft_cols])
pca_test= pca.fit_transform(world_2019[ft_cols])
```

[151]:
```
# entrenament Random Forest amb els components del PCA
rf.fit(pca_train, y_train)
# predicció del 2019
y_pred_rf = rf.predict(pca_test)
# error absolut mitjà
mean_absolute_error(y_test, y_pred_rf)
```

[151]:
```
0.3776901268478519
```

Com que és el model que realitza una millor predicció, què en podem dir de la importància que dóna als atributs per predir el "Happiness Score"? Quins són els que més contribueixen a explicar-lo? Aquesta és una de les preguntes plantejades als objectius.

Com que hem una reducció PCA, la deducció de la importància dels atributs no és tan directa, però tampoc és complicada.
A continuació veiem que el Principal Component amb més importància és el 1r, amb un valor de 0,75 i a molta distància respecte el segon, que és el PC3 amb una importància de 0,11. La resta de valors són força més baixos, amb importàncies d'entre 0,04 i 0,05.
Per altra banda, veiem seguidament que el PC1 explica un 69% de la variància de les dades, mentre que el PC2 n'explica un 15% i el PC3 un 6,9%.
Finalment, valorem la importància dels atributs en funció de les respectives direccions de màxima variància de cadascun dels Principal Components. Hem dit que el PC1 és de tros el més important,

a part d'explicar la major part de la variància. Aquest segueix sobretot la direcció de l'atribut "Economy", seguit a més distància de "Health" i, a molta més distància, de "Freedom". Pel que fa al PC3, el segon més important, apunta sobretot en la direcció de "Freedom", seguit per "Generosity" i "Trust"; encara que cal recordar que la importància d'aquest és força més moderada.

Així doncs, sembla que el factor més important per a la predicció del "Happiness Score" és l'economia del país, juntament amb l'esperança de vida i salut de la població i, en menys mesura, de la llibertat experimentada individualment.

[141]:
```
# importància dels "Principal Components" utiltizats com a features
ft_importances = pd.DataFrame({'importances': rf.feature_importances_},␣
↪index=[f'PC{i+1}' for i in range(5)])
ft_importances.sort_values('importances', ascending=False)
```

[141]:

	importances
PC1	0.753237
PC3	0.114793
PC5	0.046825
PC4	0.045546
PC2	0.039600

[143]:
```
# percentatge de variància explicada per cada Principal Component
pd.Series(pca.explained_variance_ratio_, index=[f'PC{i+1}' for i in range(5)])
```

[143]:
```
PC1    0.685864
PC2    0.154907
PC3    0.069160
PC4    0.042341
PC5    0.029888
dtype: float64
```

[139]:
```
# direccions de màxima variància per cadascuna de les features originals i els␣
↪Principal Components
pd.DataFrame(abs(pca.components_), columns=ft_cols, index=[f'PC{i+1}' for i in␣
↪range(5)])
```

[139]:

	Economy	Health	Freedom	Trust	Generosity	Family
PC1	0.756095	0.400637	0.136764	0.066273	0.005933	0.494651
PC2	0.461595	0.216278	0.164597	0.055463	0.006975	0.842577
PC3	0.150581	0.165261	0.679573	0.396344	0.555161	0.151335
PC4	0.421605	0.870212	0.136444	0.207694	0.056506	0.005852
PC5	0.046087	0.034282	0.569240	0.079805	0.808029	0.115707

4.4.3 Decision Tree

Tot i que hem pogut analitzar la importància dels diferents atributs a l'apartat anterior amb l'entrenament del Random Forest, no hem pogut deduir quin és el seu "flux" d'importància. Per exemple, per entendre millor el "Happines Score", seria interessant respondre a preguntes com: en

els països on hi ha una millor economia, és la confiança en el govern un factor més important que en els que l'economia no és tan forta?

Per aquest motiu, es decideix crear un arbre de decisió amb totes les dades (de 2015 a 2019), sense conjunt d'entrenament ni de test, ja que no ens interessa predir, sinó entendre aquesta relació entre les diferents separacions de valors dels atributs.

A continuació s'importen les llibreries i es crea el conjunt utiltizat per entrenar el model:

[187]:
```
from sklearn import tree
import graphviz
```

[188]:
```
ft_cols = ['Economy', 'Health', 'Freedom', 'Trust', 'Generosity', 'Family']
tg_col = 'Happiness Score'
X_train, y_train = df[ft_cols], df[tg_col]
```

S'utiltiza un arbre de decisió amb una profunditat màxima de 3 per tal que sigui més senzilla la seva interpretació.

Observem que, tal i com havíem vist amb el Random Forest, l'economia és el factor més important, ja que el trobem a l'arrel de l'arbre. Per a països amb una puntuació d'economia de més de 1,061 (aproximadament el 50%), el factor que més tenen en compte posteriorment és la llibertat i, per aquells amb una major sensació de llibertat (només un ~30% dels països), la confiança en el govern passa a ser el tercer factor decisiu.

Per altra banda, els països amb una economia menys potent valoren la salut com a segona opció després de l'economia. Per aquells amb millor salut d'aquest grup, el tercer factor és el suport faimiliar; mentre que en aquells amb menys salut l'economia torna a ser un element decisiu que sembla que determina la seva felicitat.

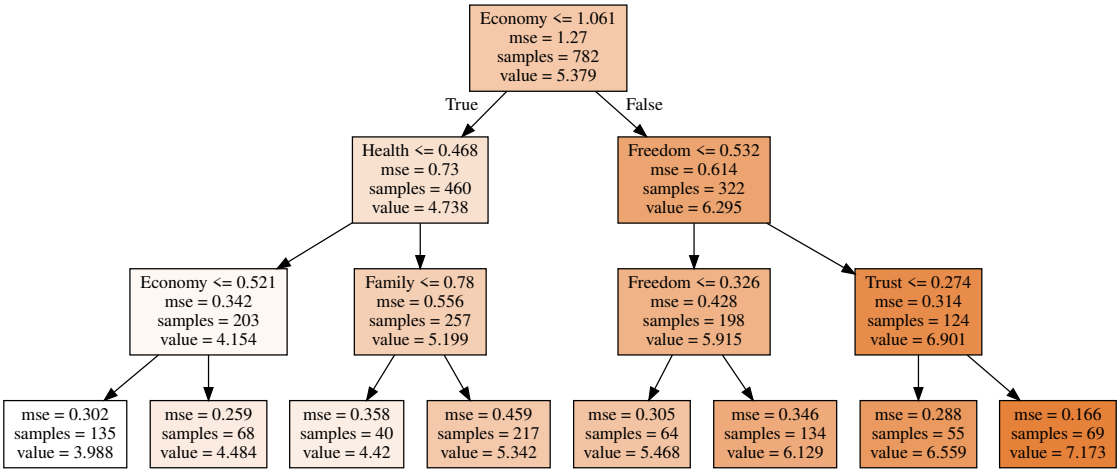
Una observació interessant és que la salut no forma part de la branca de decisió dels països amb més recursos. Una possible explicació és que, probablement, en aquests països ja es dóna per fet que hi ha un bon sistema de salut i, com que ja ve donat i tothom hi està acostumat, aquest factor passa a ser secundari en la felicitat de la població.

[199]:
```
dtree = tree.DecisionTreeRegressor(max_depth=3)
dtree.fit(X_train, y_train)

dot_data = tree.export_graphviz(dtree, out_file=None,
                                feature_names=ft_cols,
#                                 class_names=iris.target_names,
                                filled=True)

# Draw graph
graph = graphviz.Source(dot_data, format="png")
graph
```

[199]:



5 Representació

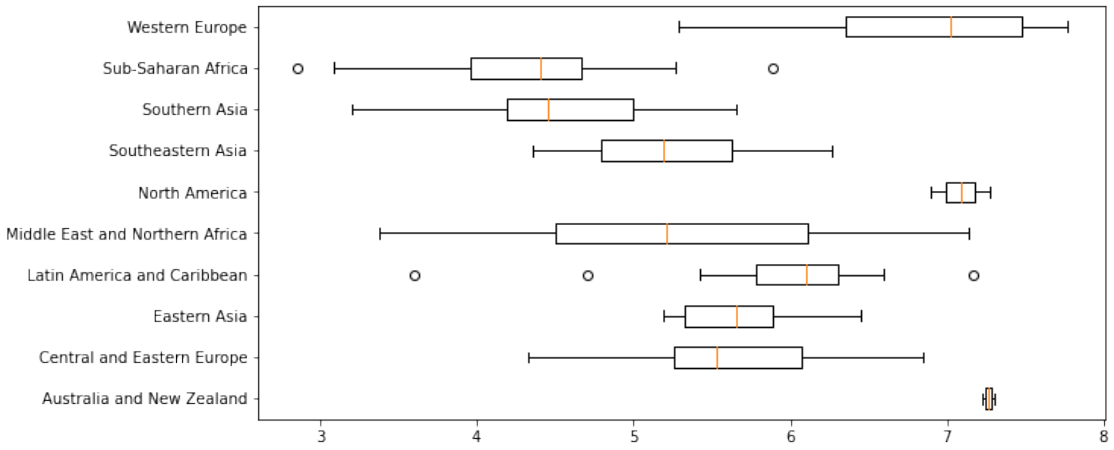
5.1 Regions

Per tal de comparar els ``Happiness Score'' de les diferents regions, realitzem un boxplot de les seves distribucions de puntuacions l'any 2019, que és l'últim pel qual tenim dades.

Podem observar que hi ha 3 regions amb, clarament, millors puntuacions de felicitat: ``Western Europe'', ``North America'' i ``Australia and New Zealand'', corresponent als països més desenvolupats. Tot i això, fins i tot dins d'aquestes hi ha força variació. A l'haver-hi pocs països, en la regió d'Austràlia i Nova Zelanda tots tenen unes puntuacions força altes. A ``Western Europe'' hi ha un rang molt més elevat, ja que països com Grècia o Portugal estan a la cua, mentre Finlàndia o Dinamarca estan al capdavant.
Les pitjors puntuacions les trobem al ``Sub-Saharan Africa'' i ``Southern Asia''.

``Middle East and Northern Africa'' és la que mostra un rang més variat de puntuacions, anant des de les més baixes (com Yemen) a les més altes (com Israel o UAE).

[92]:
```
plt.boxplot(region_happiness_2019.values(), labels=region_happiness_2019.
↪keys(), vert=False)
plt.show()
```

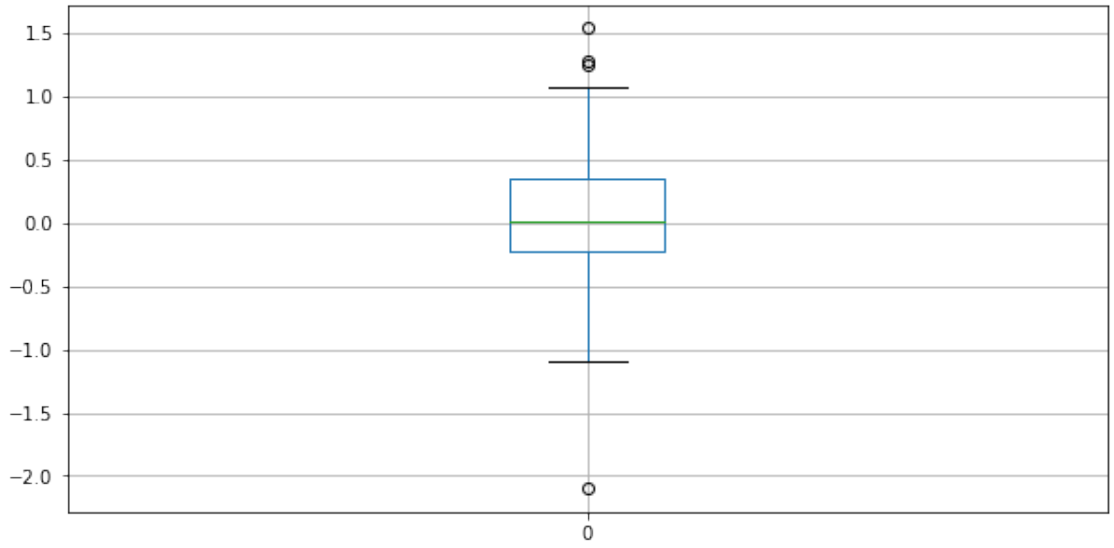


5.2 Diferències 2015-2019

En aquest apartat analitzem les diferències entre les felicitats dels països entre els anys 2015 i 2019.

En el següent boxplot veiem que hi ha un país on la felicitat arriba a decreixer 2 punts durant aquests 4 anys, mentre que a l'altre extrem trobem països que augmenten en 1,5 punts.

[93]:
```
happiness_diff_2015_2019.to_frame().boxplot()
plt.show()
```



Pel que fa als països més afortunats, veiem que l'encapçala Benin, seguit de la costa d'Ivory i Togo.

Pel que fa a Benin, veiem en el ``bar plot'' que la felicitat ja havia anat creixent any rere any des de 2015 fins a 2019.

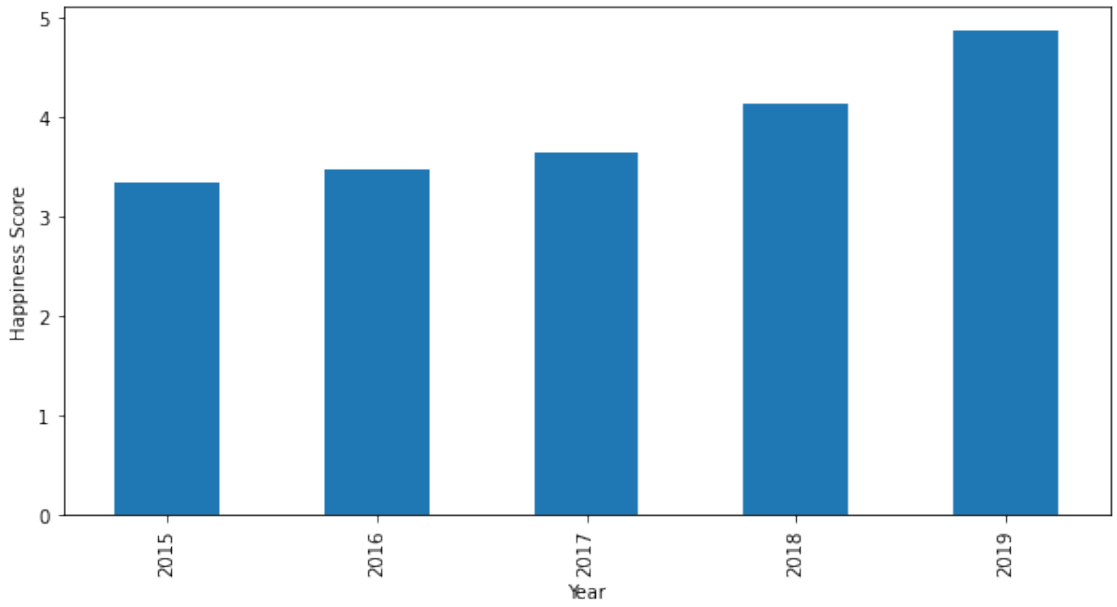
Hi ha cap dels sis factors que pugui explicar aquest augment de felicitat? N'hi ha algun que hagi augmentat significativament?
Doncs segons el plot que veiem a continuació, no sembla haver-hi cap factor que estigui influint significativament en aquest augment de felicitat.

[94]:
```
# better happiness diff
happiness_diff_2015_2019.sort_values(ascending=False).head()
```

[94]:
```
Country
Benin           1.543
Ivory Coast     1.289
Togo            1.246
Honduras        1.072
Burkina Faso    1.000
dtype: float64
```

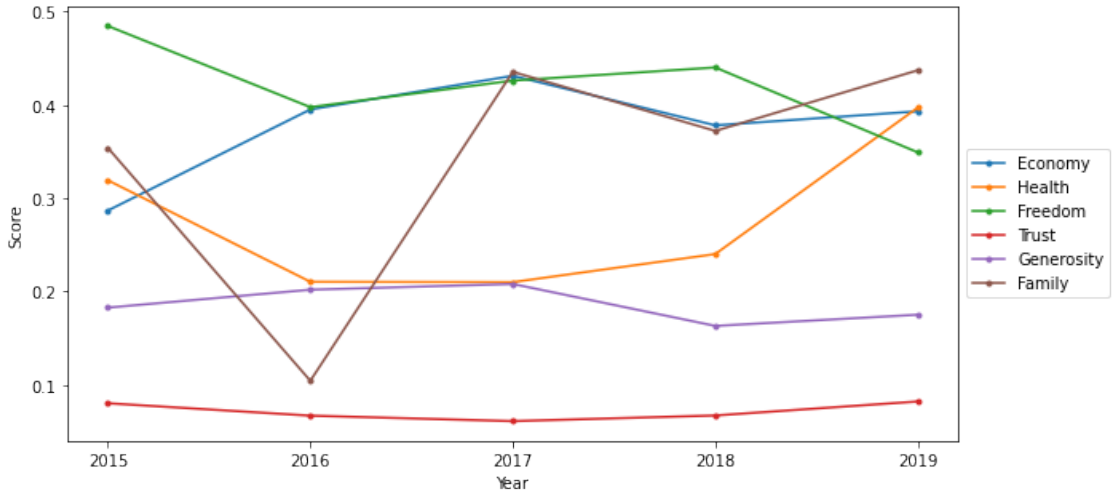
[210]:
```
benin_df = df[df['Country'] == 'Benin']

benin_df.set_index('Year')['Happiness Score'].plot.bar()
plt.ylabel('Happiness Score')
plt.show()
```



[191]:
```
for col in ft_cols:
    plt.plot(benin_df[col].values, label=col, marker='.')
```

```
plt.xticks(range(len(benin_df)), benin_df['Year'])
plt.xlabel('Year')
plt.ylabel('Score')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()
```



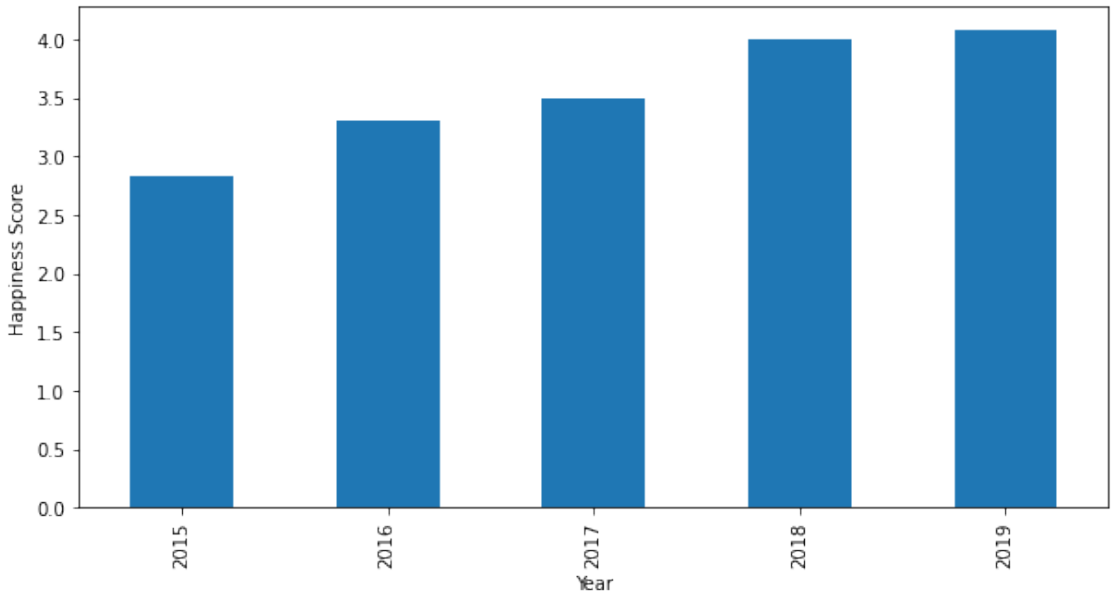
Podem observar una cosa similar amb Togo. Els nivells de felicitat augmenten any rere any des del 2015 fins al 2019, però si mirem l'evolució dels sis factors, no sembla que hi hagi res especial a part d'un increment de la puntuació de ``Family'', encara que aquest no sembla ser del tot suficient per explicar l'increment de felicitat.

El fet de no trobar cap relació evident entre els indicadors i el gran augment dels nivells de felicitat d'aquests països pot ser degut a diverses causes. Alguns exemples:

- Hi ha una relació més complexa que costa d'observar a ull, potser els increments i decrements de múltiples variables al mateix temps són els que estan afectant els nivells de felicitat.
- Les dades recollides no són prou fiables (massa pocs enquestats, selecció no aleatòria, etc.).
- Hi ha altres factors que no estem considerant que tenen una forta influència en aquests augments.

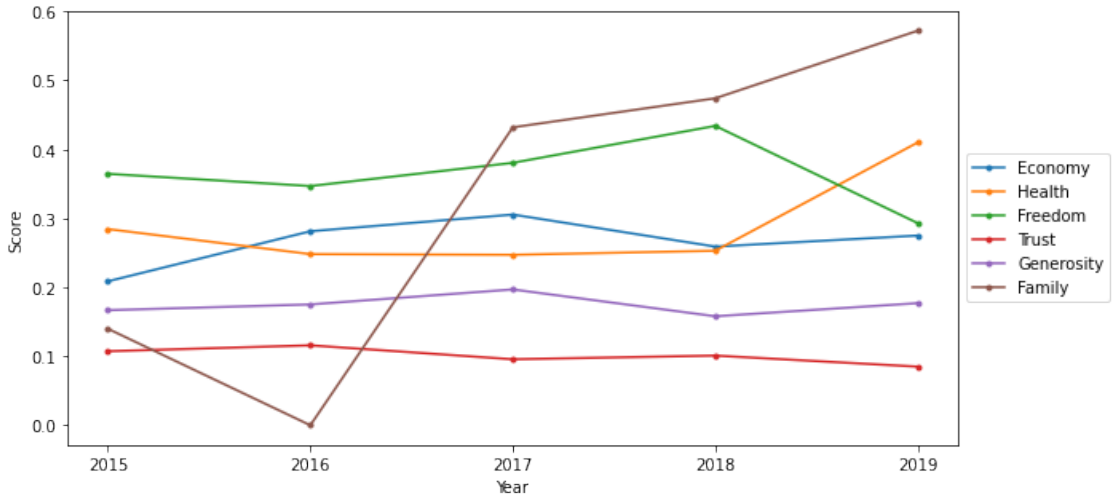
[104]:
```
togo_df = df[df['Country'] == 'Togo']
```

[164]:
```
togo_df.set_index('Year')['Happiness Score'].plot.bar()
plt.ylabel('Happiness Score')
plt.show()
```



[192]:
```
for col in ft_cols:
    plt.plot(togo_df[col].values, label=col, marker='.')

plt.xticks(range(len(benin_df)), benin_df['Year'])
plt.xlabel('Year')
plt.ylabel('Score')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()
```



Pel que fa als països on més han defallit els nivells de felicitat des de 2015 a

2019, trobem Venezuela a la cua amb una diferència negativa de -2, el doble que Lesotho i Zambia que la segueixen.

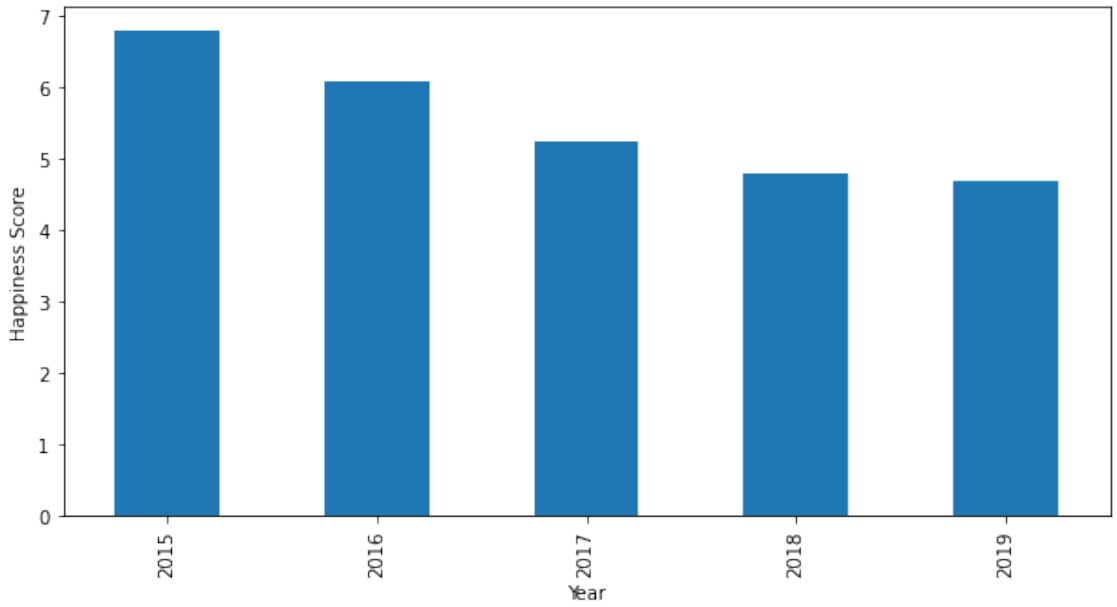
Pel que fa a Venezuela, veiem que ha anat decreixent la puntuació paulatinament, passant d'una bona puntuació de ~7 a una de ~5. Tal i com passava en els casos anteriors, aquests sis factors no semblen suficients per explicar aquesta gran devallada. El mateix ens trobaríem si observéssim els altres països amb males puntuacions.

[373]:
```
happiness_diff_2015_2019.sort_values(ascending=True).head()
```

[373]:
```
Country
Venezuela   -2.103
Lesotho     -1.096
Zambia      -1.022
Zimbabwe    -0.947
Haiti       -0.921
dtype: float64
```

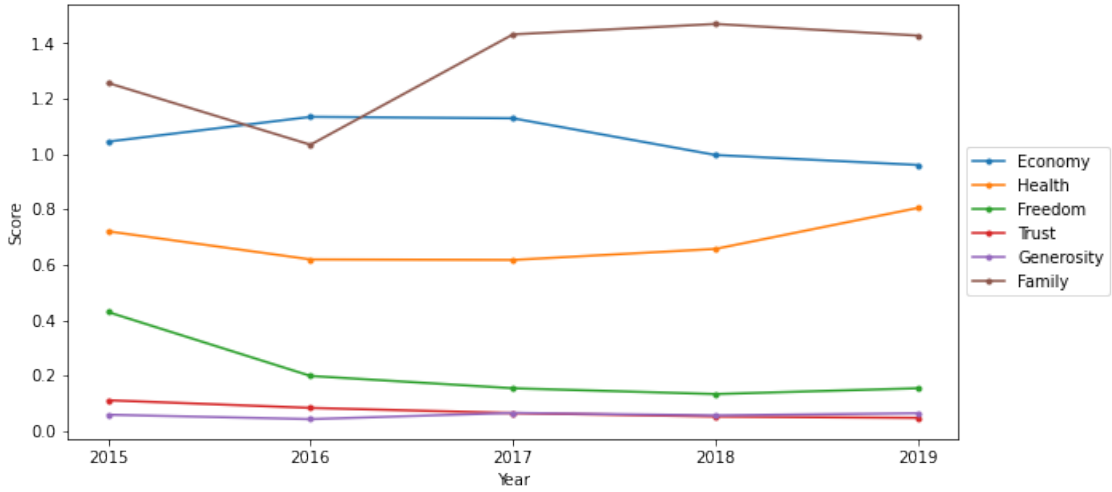
[165]:
```
vene_df = df[df['Country'] == 'Venezuela']
```

[166]:
```
vene_df.set_index('Year')['Happiness Score'].plot.bar()
plt.ylabel('Happiness Score')
plt.show()
```



[193]:
```
for col in ft_cols:
    plt.plot(vene_df[col].values, label=col, marker='.')
```

```
plt.xticks(range(len(benin_df)), benin_df['Year'])
plt.xlabel('Year')
plt.ylabel('Score')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()
```



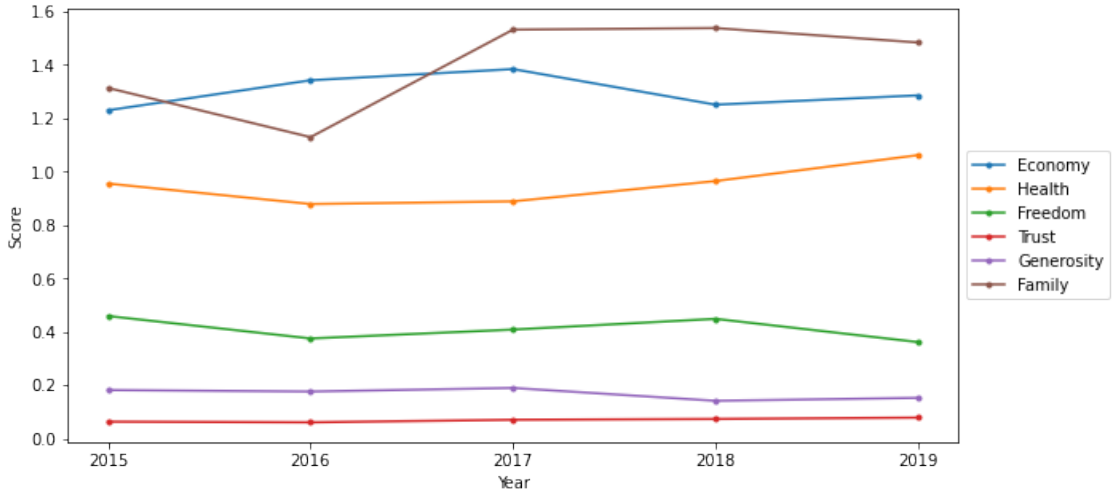
5.3 Espanya

Pel que fa als 6 atributs, sembla que Espanya s'ha mantingut en uns valors força similars al llarg dels anys, destacant un bon augment de la cohesió familiar i uns baixos nivells generals de confiança en el govern i de generositat.

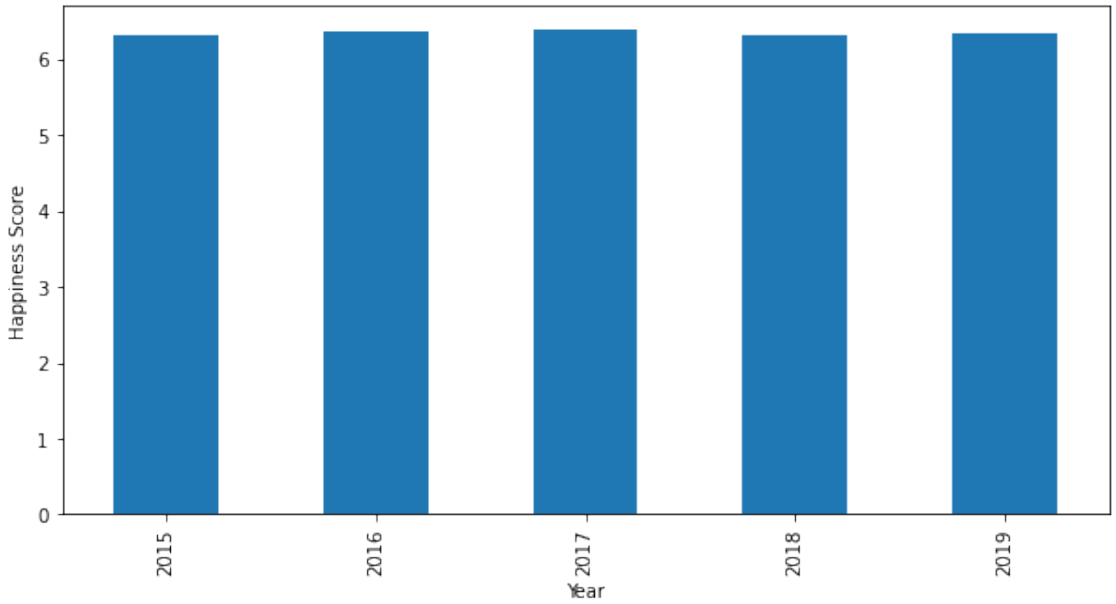
Pel que fa als nivells de felicitat, veiem que aquests també es mantenen en un rang força similar d'aproximadament 6,3 punts.

[195]:
```
for col in ft_cols:
    plt.plot(spain_df[col].values, label=col, marker='.')

plt.xticks(range(len(benin_df)), benin_df['Year'])
plt.xlabel('Year')
plt.ylabel('Score')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()
```



[168]:
```
spain_df.set_index('Year')['Happiness Score'].plot.bar()
plt.ylabel('Happiness Score')
plt.show()
```



Comparant Espanya amb la mitjana de la resta de països de la regió i a nivell mundial amb les dades de 2019, veiem que destaca especialment en salut, ja que està força per sobre de la mitjana mundial i lleugerament per sobre de la de la zona oest Europea. Una cosa similar passa amb ``Family'', encara que en menys mesura.

Pel que fa a la llibertat, confiança en el govern i generositat, veiem que encara ens queda molt per millorar, ja que està substancialment per sota de la mitjana de països de l'oest d'Europa i lleugerament per sota de la mitjana mundial.

En quan a l'economia, tot i trobar-se una mica per sota de la mitjana Europea, està molt per sobre de la mundial.

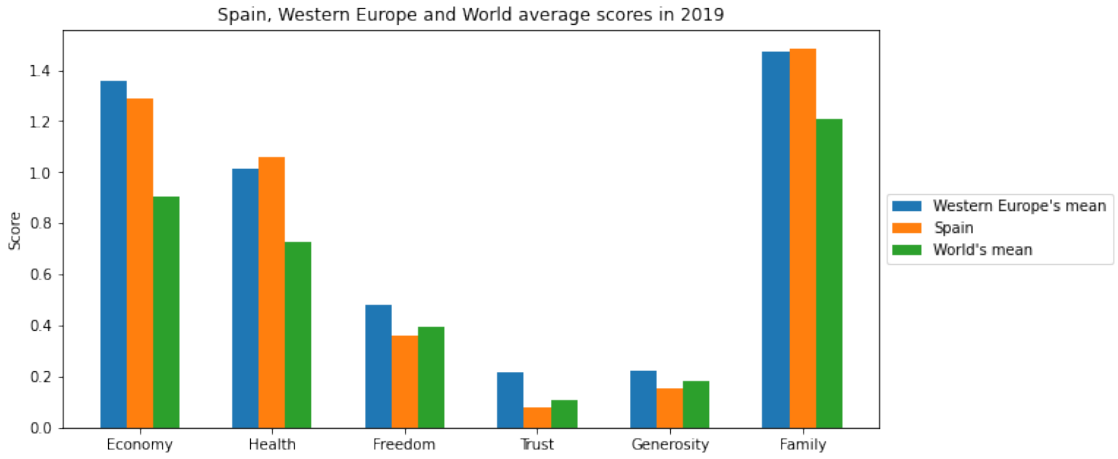
[143]:
```
# fts = ft_cols + ['Happiness Score']
fts = ft_cols

x = np.array(range(len(fts)))
y = western_europe_2019[fts].mean(axis=0)
z = spain_df[spain_df['Year'] == 2019][fts].values[0]
k = world_2019[fts].mean(axis=0)

ax = plt.subplot(111)
ax.bar(x-0.2, y, width=0.2, align='center', label='Western Europe\'s mean')
ax.bar(x, z, width=0.2, align='center', label='Spain')
ax.bar(x+0.2, k, width=0.2, align='center', label='World\'s mean')

ax.set_xticklabels([0]+fts)
ax.set_ylabel('Score')
ax.set_title('Spain, Western Europe and World average scores in 2019')

plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()
```



En quan a la puntuació de felicitat, Espanya també es troba entremig de la mitjana oest-europea i la mundial, estan però més a prop de la primera.

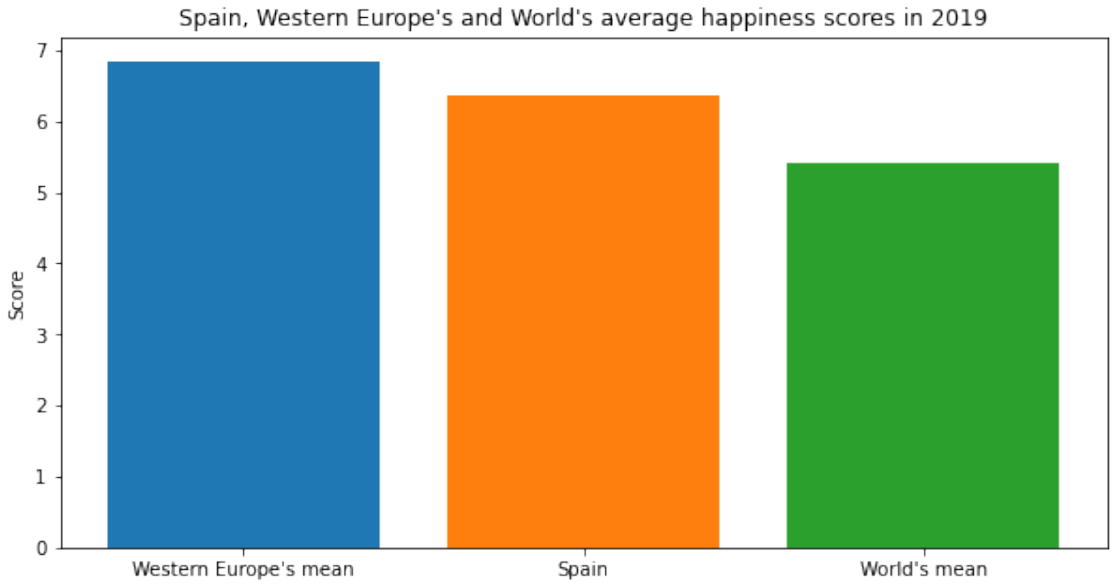
[161]:
```
fts = ['Happiness Score']
```

```
x = ['Western Europe\'s mean', 'Spain', 'World\'s mean']
y = western_europe_2019[fts].mean(axis=0)
z = spain_df[spain_df['Year'] == 2019][fts].values[0]
k = world_2019[fts].mean(axis=0)

plt.bar(x, [y[0], z[0], k[0]], color=[u'#1f77b4', u'#ff7f0e', u'#2ca02c'])

plt.ylabel('Score')
plt.title('Spain, Western Europe\'s and World\'s average happiness scores in␣
 ↪2019')

plt.show()
```



6 Resolució

Donant resposta a les preguntes plantejades als objectius:

Dels sis factors utilitzats (economia, família, esperança de vida, llibertat, confiança en les institucions públiques i generoistat), els que sembla especialment important per determinar la felicitat d'un país és l'economia, seguit per la salut i la llibertat. S'ha observat que en països més rics es valora més la llibertat que la salut, ja que probablement la bona assistència sanitària es dóna per feta.

Hem utilitzat un model Random Forest amb reducció de dimensionalitat amb PCA per tal de predir les puntuacions de felicitat de l'any 2019 utilitzant els sis factors entre els anys 2015 i 2018 com a entrenament. S'ha obtingut un error absolut mitjà de 0,37, consistint en una estimació molt bona considerant que els

valors de felicitat van de 3 a 8 punts, aproximadament.

Els països que més han augmentat la felicitat entre 2015 i 2019 són Benin i la Costa d'Ivory, amb augments de 1,5 i 1,3 respectivament. Els que han vist reduida la seva felicitat en major mesura són Venezuela i Lesotho, amb decreixements de -2,1 i 1,1 respectivament. No s'ha observat cap relació evident entre aquestes pujades i baixades i els sis indicadors utiltizats.

S'observen diferències evidents entre les distribucions de felicitat de les diferents regions del planeta, essent les que mostren felicitats més grans les corresponents a regions més desenvolupades: Europa-oest, Amèrica del nord i Austràlia i Nova Zelanda. Per altra banda, les regions de l'Àfrica subsahariana i Àsia del sud són les que tenen, generalment, felicitats més baixes.

Espanya mostra nivells relativament estables dels indicadors i la felicitat al llarg del temps. Destaca especialment en la salut i la cohesió familiar, ja que està per sobre de la mitjana europea i mundial. Per altra banda, la confiança en el govern, la generositat i la sensació de llibertat estan per sota de les altres dues mitjanes, mostrant que encara hi ha espai de millora.

7 Referències

[1] https://worldhappiness.report/

[2] www.gallup.com

[3] https://news.gallup.com/poll/122453/understanding-gallup-uses-cantril-scale.aspx

[4] https://www.forbes.com/sites/jamesellsmoor/2019/07/11/new-zealand-ditches-gdp-for-happines

[5] https://nypost.com/2017/03/22/that-world-happiness-survey-is-complete-crap/

[6] Marquez-Padilla, Fernanda, and Jorge Alvarez. ``Grading happiness: what grading systems tell us about cross-country wellbeing comparisons.'' Economics Bulletin 38.2 (2018): 1138-1155.

[7] https://www.kaggle.com/unsdsn/world-happiness?select=2018.csv

[8] https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.boxplot.html

[9] Razali, Nornadiah Mohd, and Yap Bee Wah. ``Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests.'' Journal of statistical modeling and analytics 2.1 (2011): 21-33.