

HANNAH, CHARLOTTE, MALON

IS THE WEATHER CORRELATED WITH THE WORLD HAPPINESS RANKING?

Focus on process instead of analysis

DATA SOURCES

- World Happiness Report —> [CSV download](#)
- meteostat.net —> [API for weather data](#)
- ncdc.noaa.gov/cdo-web/ —> [CSV download for weather data](#)

WORKFLOW

- Download World Happiness Report.csv

3 main business decisions for data cleaning:

- Focus on the countries that are present in all 5 years (2015-2019)
- Limit data to the ranking and score
- Keep only the countries which have a corresponding capital available

World Happiness Report

Data Explorer

78.96 KB

- 2015.csv
- 2016.csv
- 2017.csv
- 2018.csv
- 2019.csv

< 2015.csv (16.17 KB)

Download

Grid

Fullscreen

Detail

Compact

Column

10 of 12 columns

About this file

Happiness rank and scores by country, 2015

<u>A</u> Country	<u>A</u> Region	# Happiness Rank	# Happiness Score	# Standard Error
Name of the country.	Region the country belongs to.	Rank of the country based on the Happiness Score.	A metric measured in 2015 by asking the sampled people the question: "How would you rate your happiness on a	The standard error of happiness score.

Final countries

In [88]: capsdf

Out[88]:

	Country_final	Capital
0	Mauritius	Port Louis
1	Austria	Vienna
2	Iceland	Reykjavik
3	South Korea	Seoul
4	Bosnia and Herzegovina	Sarajevo
...
136	New Zealand	Wellington
137	Yemen	Sana'a
138	Hungary	Budapest
139	Kyrgyzstan	Bishkek
140	Canada	Ottawa

141 rows × 2 columns

Combining all years

Score, rank, countries

[109]:

	ID	pwc	score_15	rank_15	rank_16	score_16	score_17	rank_17	score_18	rank_18	score_19	rank_19	Capi	Capitals
0	1	Mauritius	5.828	57.0	57.0	5.835	5.825	57.0	5.875	57.0	5.888	57.0	Port Louis	Port Louis
1	2	Austria	7.284	10.0	10.0	7.291	7.284	10.0	7.272	10.0	7.246	10.0	Vienna	Vienna
2	3	Iceland	7.522	4.0	4.0	7.498	7.494	4.0	7.495	4.0	7.494	4.0	Reykjavik	Reykjavik
3	4	South Korea	5.855	54.0	54.0	5.919	5.850	54.0	5.915	54.0	5.895	54.0	Seoul	Seoul
4	5	Bosnia and Herzegovina	5.268	78.0	78.0	5.389	5.279	78.0	5.398	78.0	5.386	78.0	Sarajevo	Sarajevo
...
136	137	New Zealand	7.364	8.0	8.0	7.334	7.314	8.0	7.324	8.0	7.307	8.0	Wellington	Wellington
137	138	Yemen	3.655	151.0	151.0	3.607	3.471	151.0	3.408	151.0	3.380	151.0	Sana'a	Sana'a
138	139	Hungary	5.759	62.0	NaN	NaN	5.758	62.0	5.752	62.0	5.758	62.0	Budapest	Budapest
139	140	Kyrgyzstan	5.124	86.0	86.0	5.177	5.230	86.0	5.246	86.0	5.261	86.0	Bishkek	Bishkek
140	141	Canada	7.286	9.0	9.0	7.313	7.284	9.0	7.314	9.0	7.278	9.0	Ottawa	Ottawa

141 rows × 14 columns

WORKFLOW

- Retrieve weather data through the API

Necessary steps to get the right data

- Request API key
- Final country list of World Happiness Report
- Corresponding capitals
- Collect longitude + latitude of specific weather station (for capitals)
- Call 'Finding Stations' part of API for lat + lon information
- Call 'Daily Data' part of API for actual weather data

Errors and struggles

MATCH COUNTRIES WITH CAPITALS

Source: <https://github.com/porimol/countryinfo>

```
country = CountryInfo( 'Denmark' )  
print(country.capital())
```

Copenhagen

```
capitals = {}  
  
for country in country_list:  
    try:  
        capital_output = CountryInfo(country)  
        capitals.update({country : (capital_output.capital())})  
    except Exception:  
        capitals.update({country : ("N.A.")})
```


FIRST API RESULT (manual input)

```
{'meta': {'exec_time': 0.038, 'generated': '2020-08-27 17:04:38'},  
  'data': [{'id': '02974',  
             'name': {'en': 'Helsinki-Vantaa'},  
             'country': 'FI',  
             'region': None,  
             'national': None,  
             'wmo': '02974',  
             'icao': 'EFHK',  
             'iata': 'HEL',  
             'latitude': 60.3167,  
             'longitude': 24.9667,  
             'elevation': 51,  
             'timezone': 'Europe/Helsinki',  
             'active': True},
```

‘FINDING STATIONS’ SAVING LAT + LON INFO

```
capital_list
lat = {}
lon = {}
for capital in capital_list:
    try:
        headers = {"x-api-key": getpass.getpass()}
        url = f"https://api.meteostat.net/v2/stations/search?query={'capital'}"
        r = requests.get(url, headers=headers)
        results = r.json()
        print(capital)
        #url = f"https://api.meteostat.net/v2/stations/search?query={capital}"
        if results["data"][0]["active"] == True:
            lat.update({capital : (results["data"][0]["latitude"])})
            lon.update({capital : (results["data"][0]["longitude"])})
            print(lat)
            print(lon)
        #
        else:
            #
            print(capital)
            #
            print("it's not working hunny")
    except:
        print("wasn't here")
        continue
```

‘FINDING STATIONS’ EXPECTED OUTPUT

```
{'helsinki': 41.2833}
{'helsinki': 2.0667}
{'helsinki': 41.2833, 'barcelona': 60.3167}
{'helsinki': 2.0667, 'barcelona': 24.9667}
{'helsinki': 41.2833, 'barcelona': 60.3167, 'amsterdam': 52.3702}
{'helsinki': 2.0667, 'barcelona': 24.9667, 'amsterdam': 52.3702}
```

‘DAILY DATA’ EXPECTED OUTPUT

```
{'meta': {'source': 'National Oceanic and Atmospheric Administration, Deutscher Wetterdienst',  
  'exec_time': 0.127,  
  'generated': '2020-08-28 10:53:26'},  
'data': [{'date': '2018-08-27',  
  'tavg': 25.1,  
  'tmin': 22.5,  
  'tmax': 29,  
  'prcp': 0,  
  'snow': None,  
  'wdir': 240,  
  'wspd': 12,  
  'wpgt': None,  
  'pres': 1016,  
  'tsun': None}]}
```

WORKFLOW

- Weather info from ncdc.noaa.gov/ .CSV

Business decisions for data cleaning:

- Focus on top 3 ranked countries per year (2015-2019)
- Keep only the weather information that made sense
temperatures (min, max, avg), rain,

Cleaning

```
In [24]: df15.shape
```

```
Out[24]: (17, 11)
```

```
In [25]: df15.isna().sum()
```

```
Out[25]: Station_ID      0  
         Name           0  
         Year           0  
         Snow          10  
         #<0C           6  
         #>21.1C        6  
         #>32.2C        6  
         Precipitation   1  
         Temp.Average    6  
         Temp.Max        6  
         Temp.Min        4  
         dtype: int64
```


Dataframe

In [34]: df19

Out[34]:

	Station_ID	Name	Year	#<0C	#>21.1C	Precipitation	Temp.Average	Temp.Max	Temp.Min
0	NOE00110860	STADLANDET, NO	2019-09	0.0	0.0	9.13	51.7	57.8	45.6
1	NOE00110860	STADLANDET, NO	2019-10	0.0	0.0	9.53	44.8	49.7	39.8
2	NOE00110860	STADLANDET, NO	2019-11	1.0	0.0	3.43	35.3	39.0	31.6
3	NOE00110860	STADLANDET, NO	2019-12	0.0	0.0	14.22	39.4	43.7	35.1
4	NOE00109993	KVIKNE I OSTERDAL, NO	2019-09	NaN	NaN	2.43	NaN	NaN	NaN
...
2004	FIE00144057	VARKAUS KOSULANNIEMI, FI	2019-12	7.0	0.0	2.72	30.6	33.7	27.5
2005	FIE00146117	KUUSAMO TEERIRANTA, FI	2019-09	NaN	NaN	1.63	NaN	NaN	NaN
2006	FIE00146117	KUUSAMO TEERIRANTA, FI	2019-10	NaN	NaN	3.87	NaN	NaN	NaN
2007	FIE00146117	KUUSAMO TEERIRANTA, FI	2019-11	NaN	NaN	3.61	NaN	NaN	NaN
2008	FIE00146117	KUUSAMO TEERIRANTA, FI	2019-12	NaN	NaN	2.58	NaN	NaN	NaN

2009 rows x 9 columns

OPEN TO DOS

- Merge downloaded weather information with the World Happiness ranking
- Analyse trends over time
- Conclusion