1. Import Libraries

In [5]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

2. Load Datasets

In [6]:
```
# Importing data frames
df_climate = pd.read_csv("data/climate_change_indicators.csv")
df_ugly = pd.read_csv("data/data_is_ugly.csv")
df_weather = pd.read_csv("data/GlobalWeatherRepository.csv")
df_sleep = pd.read_csv("data/Sleep_health_and_lifestyle_dataset.csv")
df_imdb = pd.read_csv("data/IMDb_updated.csv")
```

3. Check Size and Structure

In [7]:
```
# Preview dataset Climate
print("Dataset Size: {} rows, {} columns".format(df_climate.shape[0], df_climate.shape[1]
print(df_climate.head())
# Pass
```

```
Dataset Size: 225 rows, 72 columns
   ObjectId                      Country ISO2 ISO3  \
0         1  Afghanistan, Islamic Rep. of   AF  AFG
1         2                      Albania   AL  ALB
2         3                      Algeria   DZ  DZA
3         4               American Samoa   AS  ASM
4         5      Andorra, Principality of   AD  AND

                                           Indicator           Unit  \
0  Temperature change with respect to a baseline ...  Degree Celsius
1  Temperature change with respect to a baseline ...  Degree Celsius
2  Temperature change with respect to a baseline ...  Degree Celsius
3  Temperature change with respect to a baseline ...  Degree Celsius
4  Temperature change with respect to a baseline ...  Degree Celsius

                                              Source CTS_Code  \
0  Food and Agriculture Organization of the Unite...     ECCS
1  Food and Agriculture Organization of the Unite...     ECCS
2  Food and Agriculture Organization of the Unite...     ECCS
3  Food and Agriculture Organization of the Unite...     ECCS
4  Food and Agriculture Organization of the Unite...     ECCS

                    CTS_Name  \
0  Surface Temperature Change
1  Surface Temperature Change
2  Surface Temperature Change
3  Surface Temperature Change
4  Surface Temperature Change

                               CTS_Full_Descriptor  ...  F2013  F2014  \
0  Environment, Climate Change, Climate Indicator...  ...  1.281  0.456
1  Environment, Climate Change, Climate Indicator...  ...  1.333  1.198
2  Environment, Climate Change, Climate Indicator...  ...  1.192  1.690
3  Environment, Climate Change, Climate Indicator...  ...  1.257  1.170
4  Environment, Climate Change, Climate Indicator...  ...  0.831  1.946

   F2015  F2016  F2017  F2018  F2019  F2020  F2021  F2022
0  1.093  1.555  1.540  1.544  0.910  0.498  1.327  2.012
1  1.569  1.464  1.121  2.028  1.675  1.498  1.536  1.518
2  1.121  1.757  1.512  1.210  1.115  1.926  2.330  1.688
3  1.009  1.539  1.435  1.189  1.539  1.430  1.268  1.256
4  1.690  1.990  1.925  1.919  1.964  2.562  1.533  3.243

[5 rows x 72 columns]
```

In [8]:

```python
# Preview dataset Ugly
print("Dataset Size: {} rows, {} columns".format(df_ugly.shape[0], df_ugly.shape[1]))
print(df_ugly.head())
# Not Pass, many links that are not working anymore
```

```
Dataset Size: 8596 rows, 10 columns
  post_id           created_at  \
0  xcmklj  2022-09-12 20:03:15
1  xckzxj  2022-09-12 18:54:50
2  xcd52w  2022-09-12 13:40:16
3  xcbjt6  2022-09-12 12:30:07
4  xcbjhv  2022-09-12 12:29:45

                                               title link_flair_text  score  \
0  Infographic that came with my North Carolina v...             NaN      1
1  You cannot tell me those are different shades ...             NaN      1
2                                            Neymar     Clusterfuck      1
3           Youngest WORLD Number 1 In the OPEN ERA             NaN      1
4                                why the tails omg             NaN      1

   num_comments   posted_by                              image_url  \
0             0  TheTraceur  https://i.redd.it/q5aqzkcqhhn91.jpg
1             0       Fnorv  https://i.redd.it/1r4jft2i2fn91.jpg
2             0  RedFlare07  https://i.redd.it/r6a3i6jllfn91.jpg
3             0  PhillyManc  https://i.redd.it/cq2akup8kbn91.png
4             0  biglezmate  https://i.redd.it/zv0nz7jmpcn91.png

                                           full_link   nsfw
0  https://www.reddit.com/r/dataisugly/comments/x...  False
1  https://www.reddit.com/r/dataisugly/comments/x...  False
2  https://www.reddit.com/r/dataisugly/comments/x...  False
3  https://www.reddit.com/r/dataisugly/comments/x...  False
4  https://www.reddit.com/r/dataisugly/comments/x...  False
```

In [9]:
```
# Preview dataset Sleep
print("Dataset Size: {} rows, {} columns".format(df_sleep.shape[0], df_sleep.shape[1]))
print(df_sleep.head())
# Not Pass, too small for this objective
```

```
Dataset Size: 374 rows, 13 columns
   Person ID Gender  Age            Occupation  Sleep Duration  \
0          1   Male   27     Software Engineer             6.1
1          2   Male   28                Doctor             6.2
2          3   Male   28                Doctor             6.2
3          4   Male   28  Sales Representative             5.9
4          5   Male   28  Sales Representative             5.9

   Quality of Sleep  Physical Activity Level  Stress Level BMI Category  \
0                 6                       42             6   Overweight
1                 6                       60             8       Normal
2                 6                       60             8       Normal
3                 4                       30             8        Obese
4                 4                       30             8        Obese

  Blood Pressure  Heart Rate  Daily Steps Sleep Disorder
0         126/83          77         4200            NaN
1         125/80          75        10000            NaN
2         125/80          75        10000            NaN
3         140/90          85         3000    Sleep Apnea
4         140/90          85         3000    Sleep Apnea
```

In [94]:
```
# Preview dataset Weather
print("Dataset Size: {} rows, {} columns".format(df_weather.shape[0], df_weather.shape[1]
print(df_weather.head())
# Pass
```

```
Dataset Size: 40346 rows, 41 columns
       country     location_name  latitude  longitude        timezone  \
0  Afghanistan             Kabul     34.52      69.18      Asia/Kabul
1      Albania            Tirana     41.33      19.82   Europe/Tirane
2      Algeria           Algiers     36.76       3.05  Africa/Algiers
3      Andorra  Andorra La Vella     42.50       1.52  Europe/Andorra
4       Angola            Luanda     -8.84      13.23   Africa/Luanda

   last_updated_epoch      last_updated  temperature_celsius  \
0          1693301400  2023-08-29 14:00                 28.8
1          1693301400  2023-08-29 11:30                 27.0
2          1693301400  2023-08-29 10:30                 28.0
3          1693301400  2023-08-29 11:30                 10.2
4          1693301400  2023-08-29 10:30                 25.0

   temperature_fahrenheit condition_text  ...  air_quality_PM2.5  \
0                    83.8          Sunny  ...                7.9
1                    80.6  Partly cloudy  ...               28.2
2                    82.4  Partly cloudy  ...                6.4
3                    50.4          Sunny  ...                0.5
4                    77.0  Partly cloudy  ...              139.6

   air_quality_PM10  air_quality_us-epa-index air_quality_gb-defra-index  \
0              11.1                         1                          1
1              29.6                         2                          3
2               7.9                         1                          1
3               0.8                         1                          1
4             203.3                         4                         10

    sunrise    sunset  moonrise   moonset      moon_phase  moon_illumination
0  05:24 AM  06:24 PM  05:39 PM  02:48 AM  Waxing Gibbous                 93
1  06:04 AM  07:19 PM  06:50 PM  03:25 AM  Waxing Gibbous                 93
2  06:16 AM  07:21 PM  06:46 PM  03:50 AM  Waxing Gibbous                 93
3  07:16 AM  08:34 PM  08:08 PM  04:38 AM  Waxing Gibbous                 93
4  06:11 AM  06:06 PM  04:43 PM  04:41 AM  Waxing Gibbous                 93

[5 rows x 41 columns]
```

In [10]:

```
# Preview dataset IMDB movies
print("Dataset Size: {} rows, {} columns".format(df_imdb.shape[0], df_imdb.shape[1]))
print(df_imdb.head())
# Pass
```

```
Dataset Size: 10000 rows, 6 columns
   Unnamed: 0                         title  \
0           0                      Ad Astra
1           1                     Bloodshot
2           2              Bad Boys for Life
3           3                       Ant-Man
4           4  Percy Jackson: Sea of Monsters

                                            overview original_language  \
0  The near future, a time when both hope and har...                en
1  After he and his wife are murdered, marine Ray...                en
2  Marcus and Mike are forced to confront new thr...                en
3  Armed with the astonishing ability to shrink i...                en
4  In their quest to confront the ultimate evil, ...                en

   vote_count  vote_average
0        2853           5.9
1        1349           7.2
2        2530           7.1
3       13611           7.1
4        3542           5.9
```

4. Data Types, Formats and Missing Values

In [11]:
```
# Data types climate
print(df_climate.dtypes)
print(df_climate.isnull().sum())
```

```
ObjectId       int64
Country       object
ISO2          object
ISO3          object
Indicator     object
              ...
F2018        float64
F2019        float64
F2020        float64
F2021        float64
F2022        float64
Length: 72, dtype: object
ObjectId      0
Country       0
ISO2          2
ISO3          0
Indicator     0
             ..
F2018        12
F2019        12
F2020        13
F2021        12
F2022        12
Length: 72, dtype: int64
```

In [103…
```
# Data types Weather
print(df_weather.dtypes)
print(df_weather.isnull().sum())
```

```
country                         object
location_name                   object
latitude                       float64
longitude                      float64
timezone                        object
last_updated_epoch               int64
last_updated                    object
temperature_celsius            float64
temperature_fahrenheit         float64
condition_text                  object
wind_mph                       float64
wind_kph                       float64
wind_degree                      int64
wind_direction                  object
pressure_mb                    float64
pressure_in                    float64
precip_mm                      float64
precip_in                      float64
humidity                         int64
cloud                            int64
feels_like_celsius             float64
feels_like_fahrenheit          float64
visibility_km                  float64
visibility_miles               float64
uv_index                       float64
gust_mph                       float64
gust_kph                       float64
air_quality_Carbon_Monoxide    float64
air_quality_Ozone              float64
air_quality_Nitrogen_dioxide   float64
air_quality_Sulphur_dioxide    float64
air_quality_PM2.5              float64
air_quality_PM10               float64
air_quality_us-epa-index         int64
air_quality_gb-defra-index       int64
sunrise                         object
sunset                          object
moonrise                        object
moonset                         object
moon_phase                      object
moon_illumination                int64
dtype: object
country                        0
location_name                  0
latitude                       0
longitude                      0
timezone                       0
last_updated_epoch             0
last_updated                   0
temperature_celsius            0
temperature_fahrenheit         0
condition_text                 0
wind_mph                       0
wind_kph                       0
wind_degree                    0
wind_direction                 0
pressure_mb                    0
pressure_in                    0
precip_mm                      0
precip_in                      0
humidity                       0
cloud                          0
feels_like_celsius             0
feels_like_fahrenheit          0
visibility_km                  0
visibility_miles               0
uv_index                       0
```

```
gust_mph                        0
gust_kph                        0
air_quality_Carbon_Monoxide     0
air_quality_Ozone               0
air_quality_Nitrogen_dioxide    0
air_quality_Sulphur_dioxide     0
air_quality_PM2.5               0
air_quality_PM10                0
air_quality_us-epa-index        0
air_quality_gb-defra-index      0
sunrise                         0
sunset                          0
moonrise                        0
moonset                         0
moon_phase                      0
moon_illumination               0
dtype: int64
```

In [12]:
```
# Data types IMDB
print(df_imdb.dtypes)
print(df_imdb.isnull().sum())
```

```
Unnamed: 0             int64
title                 object
overview              object
original_language     object
vote_count             int64
vote_average         float64
dtype: object
Unnamed: 0            0
title                 0
overview             30
original_language     0
vote_count            0
vote_average          0
dtype: int64
```

5. Statistical Summary

In [108…
```
# Descriptive Statistics for Climate data
print(df_climate.describe())
# Pass
```

```
         ObjectId       F1961       F1962       F1963       F1964       F1965  \
count  225.000000  188.000000  189.000000  188.000000  188.000000  188.000000
mean   113.000000    0.163053   -0.013476   -0.006043   -0.070059   -0.247027
std     65.096083    0.405080    0.341812    0.387348    0.309305    0.270734
min      1.000000   -0.694000   -0.908000   -1.270000   -0.877000   -1.064000
25%     57.000000   -0.097000   -0.164000   -0.205500   -0.236500   -0.392500
50%    113.000000    0.064500   -0.056000   -0.003000   -0.056000   -0.230500
75%    169.000000    0.318500    0.114000    0.230500    0.132500   -0.091500
max    225.000000    1.892000    0.998000    1.202000    1.097000    0.857000

            F1966       F1967       F1968       F1969  ...       F2013  \
count  192.000000  191.000000  191.000000  190.000000  ...  216.000000
mean     0.105505   -0.110832   -0.199110    0.157942  ...    0.931199
std      0.378423    0.339484    0.270131    0.308540  ...    0.321595
min     -1.801000   -1.048000   -1.634000   -0.900000  ...    0.118000
25%     -0.035750   -0.259500   -0.340000   -0.009000  ...    0.743500
50%      0.098000   -0.146000   -0.187000    0.204000  ...    0.897000
75%      0.277000    0.015000   -0.067000    0.349000  ...    1.187500
max      1.151000    1.134000    0.476000    0.939000  ...    1.643000

            F2014       F2015       F2016       F2017       F2018       F2019  \
count  216.000000  216.000000  213.000000  214.000000  213.000000  213.000000
mean     1.114815    1.269773    1.439521    1.280785    1.302113    1.443061
std      0.564903    0.462162    0.401091    0.393999    0.596786    0.467510
min     -0.092000   -0.430000    0.250000    0.017000    0.238000    0.050000
25%      0.744000    1.017750    1.147000    1.027500    0.865000    1.169000
50%      0.986500    1.215000    1.446000    1.282000    1.125000    1.412000
75%      1.335500    1.520500    1.714000    1.535000    1.834000    1.698000
max      2.704000    2.613000    2.459000    2.493000    2.772000    2.689000

            F2020       F2021       F2022
count  212.000000  213.000000  213.000000
mean     1.552038    1.343531    1.382113
std      0.621930    0.484692    0.669279
min      0.229000   -0.425000   -1.305000
25%      1.161750    1.019000    0.878000
50%      1.477000    1.327000    1.315000
75%      1.826250    1.629000    1.918000
max      3.691000    2.676000    3.243000

[8 rows x 63 columns]
```

In [13]:

```
# Descriptive Statistics for Weather data
print(df_weather.describe())
# Pass
```

```
           latitude     longitude  last_updated_epoch  temperature_celsius  \
count  40346.000000  40346.000000        4.034600e+04         40346.000000
mean      19.298562     21.760141        1.702473e+09            19.179078
std       24.521626     65.682857        5.363903e+06            10.718576
min      -41.300000   -175.200000        1.693301e+09           -41.900000
25%        3.750000     -6.840000        1.697662e+09            12.000000
50%       17.250000     23.240000        1.702668e+09            22.000000
75%       41.320000     49.880000        1.707156e+09            27.000000
max       64.100000    179.220000        1.711556e+09            45.400000

       temperature_fahrenheit      wind_mph      wind_kph   wind_degree  \
count            40346.000000  40346.000000  40346.000000  40346.000000
mean                66.522233      7.471611     12.027016    162.518242
std                 19.293569      5.172169      8.325958    106.324676
min                -43.400000      2.200000      3.600000      1.000000
25%                 53.600000      3.800000      6.100000     70.000000
50%                 71.600000      5.800000      9.400000    150.000000
75%                 80.600000     10.500000     16.900000    250.000000
max                113.700000     91.900000    148.000000    360.000000

        pressure_mb   pressure_in  ...      gust_kph  \
count  40346.000000  40346.000000  ...  40346.000000
mean    1013.890943     29.939522  ...     20.027718
std        7.437666      0.219491  ...     11.906650
min      958.000000     28.290000  ...      0.000000
25%     1010.000000     29.830000  ...     11.100000
50%     1013.000000     29.910000  ...     18.000000
75%     1018.000000     30.060000  ...     26.300000
max     1074.000000     31.710000  ...    155.200000

       air_quality_Carbon_Monoxide  air_quality_Ozone  \
count                 40346.000000       40346.000000
mean                    577.529794          43.759971
std                    1382.216080          32.658034
min                      96.800000           0.000000
25%                     237.000000          18.400000
50%                     290.400000          40.800000
75%                     447.300000          63.700000
max                   41870.102000         555.000000

       air_quality_Nitrogen_dioxide  air_quality_Sulphur_dioxide  \
count                  40346.000000                 40346.000000
mean                      14.511496                     8.242131
std                       27.044095                    22.470273
min                        0.000000                     0.000000
25%                        1.000000                     0.500000
50%                        4.500000                     1.800000
75%                       15.600000                     6.400000
max                      575.800000                   557.000000

       air_quality_PM2.5  air_quality_PM10  air_quality_us-epa-index  \
count       40346.000000       40346.00000              40346.000000
mean           25.276535          45.65170                  1.613791
std            63.684429         106.59568                  1.067879
min             0.500000           0.50000                  1.000000
25%             2.500000           4.70000                  1.000000
50%             7.500000          12.90000                  1.000000
75%            23.000000          39.50000                  2.000000
max          1558.800000        3566.40000                  6.000000

       air_quality_gb-defra-index  moon_illumination
count                40346.000000       40346.000000
mean                     2.456427          52.067268
std                      2.682182          35.314977
min                      1.000000           0.000000
25%                      1.000000          17.000000
50%                      1.000000          53.000000
```

```
75%                     2.000000          88.000000
max                    10.000000         100.000000

[8 rows x 30 columns]
```

In [112…

```python
# Descriptive Statistics for IMDB data
print(df_imdb.describe())
# Not Pass
```

```
        Unnamed: 0    vote_count  vote_average
count  10000.00000  10000.000000  10000.000000
mean    4999.50000   1020.825100      6.306230
std     2886.89568   1992.305005      1.354259
min        0.00000      0.000000      0.000000
25%     2499.75000    143.000000      5.800000
50%     4999.50000    332.000000      6.500000
75%     7499.25000    926.250000      7.100000
max     9999.00000  25148.000000     10.000000
```

In [114…

```python
# Descriptive Statistics for categorical IMDB data
for column in df_imdb.select_dtypes(include=['object']).columns:
    print(df_imdb[column].value_counts())
    print("\n")
# Not Pass
```

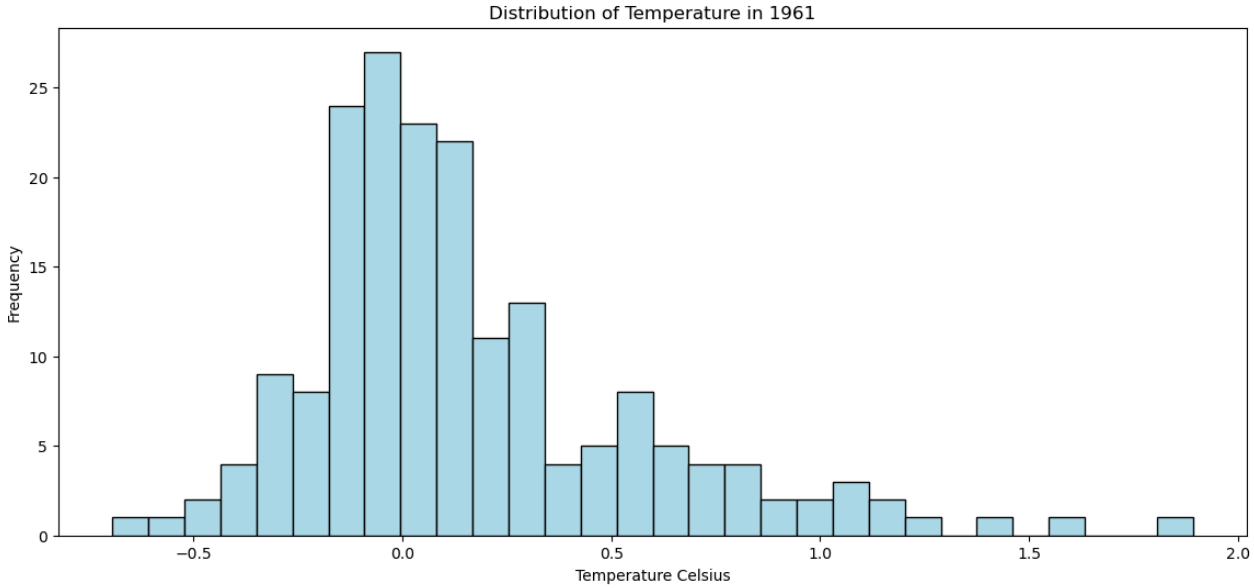
```
title
Dracula                                4
Beauty and the Beast                   4
Godzilla                               3
Fallen                                 3
Les Misérables                         3
                                      ..
The Imaginarium of Doctor Parnassus    1
Along Came Polly                       1
Grand Illusion                         1
Teenage Mutant Ninja Turtles III       1
Woochi: The Demon Slayer               1
Name: count, Length: 9689, dtype: int64


overview
Plot unknown.
5
The plot is unknown at this time.
2
Wilbur the pig is scared of the end of the season, because he knows that come that time,
he will end up on the dinner table. He hatches a plan with Charlotte, a spider that lives
in his pen, to ensure that this will never happen.
2
Bernie works at a Las Vegas casino, where he uses his innate ability to bring about misfo
rtune in those around him to jinx gamblers into losing. His imposing boss, Shelly Kaplow,
is happy with the arrangement. But Bernie finds unexpected happiness when he begins datin
g attractive waitress Natalie Belisario.                                        1
A fat Lawyer finds himself growing "Thinner" when an old gypsy man places a hex on him. N
ow the lawyer must call upon his friends in organized crime to help him persuade the gyps
y to lift the curse. Time is running out for the desperate lawyer as he draws closer to h
is own death, and grows ever thinner.                                           1

..
Six short stories that explore the extremities of human behavior involving people in dist
ress.
1
King Randolph sends for his cousin Duchess Rowena to help turn his daughters, Princess Ge
nevieve and her 11 sisters, into better ladies. But the Duchess takes away all the sister
s fun, including the sisters favorite pastime: dancing.Thinking all hope is lost they fin
d a secret passageway to a magical land were they can dance the night away.    1
An orphaned boy raised by underground creatures called Boxtrolls comes up from the sewers
and out of his box to save his family and the town from the evil exterminator, Archibald
Snatcher.
1
At an exclusive country club, an ambitious young caddy, Danny Noonan, eagerly pursues a c
addy scholarship in hopes of attending college and, in turn, avoiding a job at the lumber
yard. In order to succeed, he must first win the favour of the elitist Judge Smails, and
then the caddy golf tournament which Smails sponsors.                           1
Spanning four centuries in Korea, this epic action-adventure concerns a powerful pipe and
a trio of wizards who will do anything to protect it.
1
Name: count, Length: 9964, dtype: int64


original_language
en    8326
fr     385
ja     269
it     152
es     145
de      96
ko      92
cn      72
zh      71
hi      70
ru      69
```

```
da      32
sv      30
pt      22
id      21
ta      18
nl      13
no      13
tr      12
pl      10
tl       9
th       9
ar       6
sr       6
te       5
fa       5
cs       4
ml       4
he       4
fi       3
hu       3
af       2
el       2
ro       2
ab       1
lv       1
mr       1
ka       1
ms       1
vi       1
et       1
eu       1
sh       1
bs       1
nb       1
pa       1
la       1
is       1
xx       1
mk       1
sq       1
uk       1
Name: count, dtype: int64
```

6. Distribution Analysis

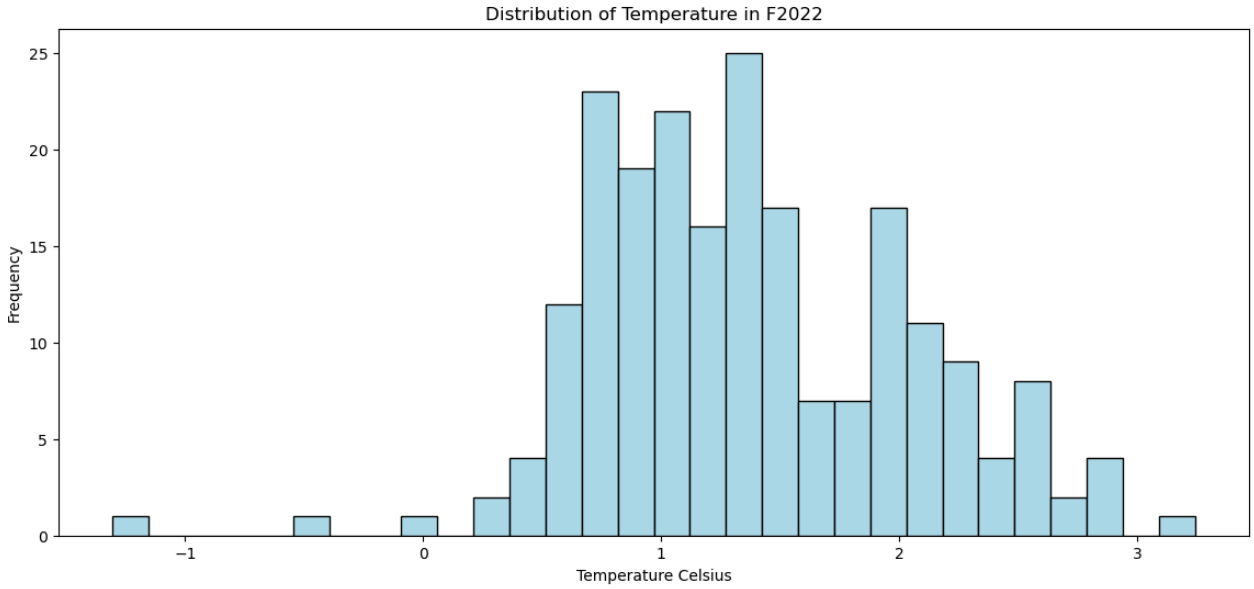
In [14]:

```python
#df_climate
# Distribution of temperature in the first notation
plt.figure(figsize=(14, 6))
plt.hist(df_climate['F1961'], bins=30, color='lightblue', edgecolor='black')
plt.title('Distribution of Temperature in 1961')
plt.xlabel('Temperature Celsius')
plt.ylabel('Frequency')
plt.show()
```



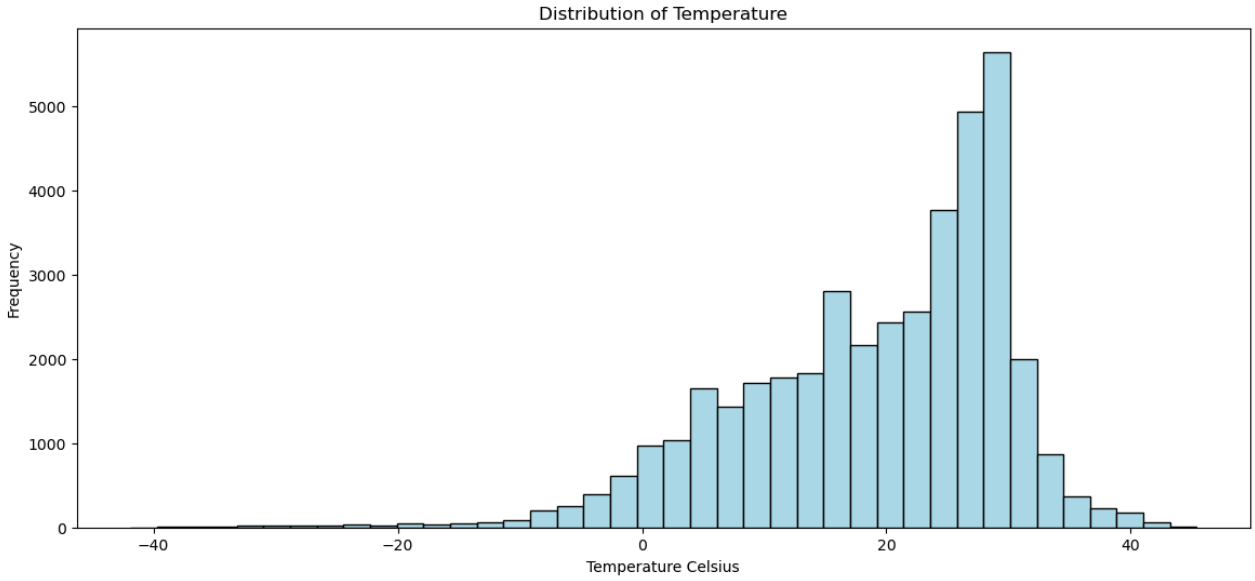
In [162…

```
#df_climate
# Distribution of temperature in the last notation
plt.figure(figsize=(14, 6))
plt.hist(df_climate['F2022'], bins=30, color='lightblue', edgecolor='black')
plt.title('Distribution of Temperature in F2022')
plt.xlabel('Temperature Celsius')
plt.ylabel('Frequency')
plt.show()
```

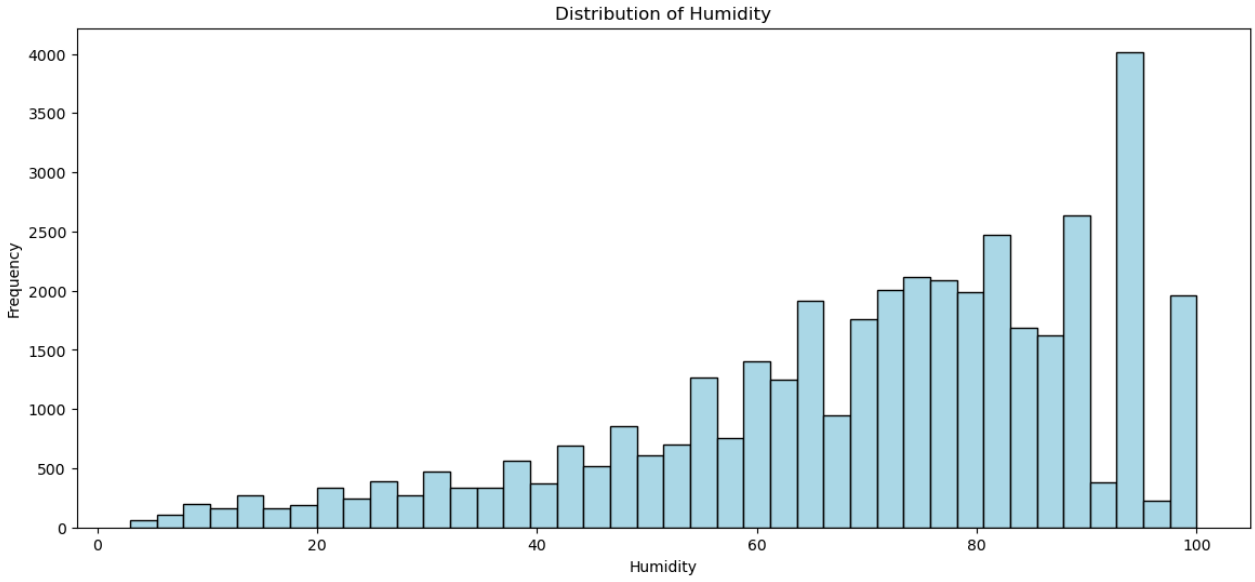


In [15]:

```
#df_weather
# Distribution of temperature
plt.figure(figsize=(14, 6))
plt.hist(df_weather['temperature_celsius'], bins=40, color='lightblue', edgecolor='black'
plt.title('Distribution of Temperature')
plt.xlabel('Temperature Celsius')
plt.ylabel('Frequency')
plt.show()
```



In [16]:
```
#df_weather
# Distribution of Humidity
plt.figure(figsize=(14, 6))
plt.hist(df_weather['humidity'], bins=40, color='lightblue', edgecolor='black')
plt.title('Distribution of Humidity')
plt.xlabel('Humidity')
plt.ylabel('Frequency')
plt.show()
```



In [ ]: