In [1]:
```
import polars as pl
```

In [3]:
```
import requests
import pandas as pd

# Define the parameters
params = {
    "latitude": 43.7001,  # Toronto
    "longitude": -79.4163,
    "start_date": "2000-02-02",
    "end_date": "2025-05-03",
    "hourly": ["temperature_2m", "apparent_temperature", "precipitation",
               "wind_speed_10m", "snow_depth", "sunshine_duration",
               "direct_radiation", "wind_gusts_10m"],
    "timezone": "America/New_York"
}

# API endpoint
url = "https://archive-api.open-meteo.com/v1/archive"

# Request data
response = requests.get(url, params=params)

# Convert to JSON
data = response.json()

# Convert hourly data into a DataFrame
df = pd.DataFrame(data["hourly"])

# Save to CSV
df.to_csv("toronto_weather.csv", index=False)

print("✅ Dataset saved as toronto_weather.csv")
```

```
✅ Dataset saved as toronto_weather.csv
```

In [5]:
```
df=pl.read_csv("toronto_weather.csv")
print(df.shape)
print(df.columns)
```

```
(221352, 9)
['time', 'temperature_2m', 'apparent_temperature', 'precipitation', 'wind_speed_1
0m', 'snow_depth', 'sunshine_duration', 'direct_radiation', 'wind_gusts_10m']
```

In [6]:
```
print(df.head(5))
```

```
shape: (5, 9)
┌──────────────┬──────────────┬──────────────┬──────────────┬───┬──────────────┬──────────────┬──────────────┐
│ time         ┆ temperature_ ┆ apparent_tem ┆ precipitatio ┆ … ┆ snow_depth   ┆ sunshine_du ┆ direct_radi  ┆ wind_gusts_  │
│ ---          ┆ 2m           ┆ perature     ┆ n            ┆   ┆ ---          ┆ ration       ┆ ation        ┆ 10m          │
│ str          ┆ ---          ┆ ---          ┆ ---          ┆   ┆ f64          ┆ ---          ┆ ---          ┆ ---          │
│              ┆ f64          ┆ f64          ┆ f64          ┆   ┆              ┆ f64          ┆ f64          ┆ f64          │
╞══════════════╪══════════════╪══════════════╪══════════════╪═══╪══════════════╪══════════════╪══════════════╪══════════════╡
│ 2000-02-02T0 ┆ -8.6         ┆ -15.0        ┆ 0.0          ┆ … ┆ 0.06         ┆ 0.0          ┆ 0.0          ┆ 40.0         │
│ 0:00         ┆              ┆              ┆              ┆   ┆              ┆              ┆              ┆              │
│ 2000-02-02T0 ┆ -9.1         ┆ -15.6        ┆ 0.0          ┆ … ┆ 0.06         ┆ 0.0          ┆ 0.0          ┆ 39.2         │
│ 1:00         ┆              ┆              ┆              ┆   ┆              ┆              ┆              ┆              │
│ 2000-02-02T0 ┆ -9.8         ┆ -16.4        ┆ 0.0          ┆ … ┆ 0.06         ┆ 0.0          ┆ 0.0          ┆ 39.6         │
│ 2:00         ┆              ┆              ┆              ┆   ┆              ┆              ┆              ┆              │
│ 2000-02-02T0 ┆ -10.5        ┆ -17.2        ┆ 0.0          ┆ … ┆ 0.06         ┆ 0.0          ┆ 0.0          ┆ 40.3         │
│ 3:00         ┆              ┆              ┆              ┆   ┆              ┆              ┆              ┆              │
│ 2000-02-02T0 ┆ -11.2        ┆ -17.9        ┆ 0.0          ┆ … ┆ 0.06         ┆ 0.0          ┆ 0.0          ┆ 39.6         │
│ 4:00         ┆              ┆              ┆              ┆   ┆              ┆              ┆              ┆              │
└──────────────┴──────────────┴──────────────┴──────────────┴───┴──────────────┴──────────────┴──────────────┴──────────────┘
```

BASIC EDA TECHNIQUES WITH POLARS

In [9]:
```
#Handling missing data
print("Null values per column")
print(df.null_count())
```

```
Null values per column
shape: (1, 9)
┌──────┬──────────────┬──────────────┬──────────────┬───┬──────────────┬──────────────┬──────────────┐
│ time ┆ temperature_ ┆ apparent_temp ┆ precipitation ┆ … ┆ snow_depth   ┆ sunshine_dura ┆ direct_radiat ┆ wind_gusts_1 │
│ ---  ┆ 2m           ┆ erature       ┆ ---          ┆   ┆ ---          ┆ tion         ┆ ion          ┆ 0m           │
│ u32  ┆ ---          ┆ ---           ┆ u32          ┆   ┆ u32          ┆ ---          ┆ ---          ┆ ---          │
│      ┆ u32          ┆ u32           ┆              ┆   ┆              ┆ u32          ┆ u32          ┆ u32          │
╞══════╪══════════════╪═══════════════╪══════════════╪═══╪══════════════╪══════════════╪══════════════╪══════════════╡
│ 0    ┆ 0            ┆ 0             ┆ 0            ┆ … ┆ 0            ┆ 0            ┆ 0            ┆ 0            │
└──────┴──────────────┴───────────────┴──────────────┴───┴──────────────┴──────────────┴──────────────┴──────────────┘
```

In [15]:
```
df_clean=df.drop_nulls()
```

In [18]:
```
q1 = df.select(pl.col("wind_gusts_10m").quantile(0.25)).item()
q3 = df.select(pl.col("wind_gusts_10m").quantile(0.75)).item()
iqr = q3 - q1

lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr
```

In [20]:
```
#Filter the outliers
df_iqr_filtered=df.filter((pl.col("wind_gusts_10m")>=lower_bound) & (pl.col("win

print("Before cleaning:")
print(df.select("wind_gusts_10m").describe())
```

```
Before cleaning:
shape: (9, 2)
```

statistic	wind_gusts_10m
str	f64
count	221352.0
null_count	0.0
mean	27.587008
std	12.689799
min	1.1
25%	18.0
50%	25.9
75%	35.3
max	119.9

In [21]:
```
print("\nAfter Cleaning:")
print(df_iqr_filtered.select("wind_gusts_10m").describe())
```

```
After Cleaning:
shape: (9, 2)
```

statistic	wind_gusts_10m
str	f64
count	218320.0
null_count	0.0
mean	27.022483
std	11.809316
min	1.1
25%	18.0
50%	25.6
75%	34.9
max	61.2

In [33]:
```
#Filtering rows by condition
cold_df=df.filter(pl.col("temperature_2m")<-15)
print(f"Number of freezing hours:{cold_df.height}")
print(cold_df.select(["temperature_2m", "apparent_temperature", "wind_gusts_10m"
```

```
Number of freezing hours:2361
shape: (10, 3)
```

temperature_2m	apparent_temperature	wind_gusts_10m
f64	f64	f64
-17.0	-22.1	14.0
-15.2	-20.3	15.8
-15.8	-21.0	15.5
-16.4	-21.7	15.1
-17.0	-22.3	15.8
-17.6	-22.6	15.5
-17.4	-22.1	14.8
-15.4	-20.9	22.3
-15.8	-21.3	21.2
-15.1	-20.2	19.4

In [32]:
```
#Selecting specific columns
subset_df=df.select([
    "temperature_2m",
    "precipitation",
    "wind_gusts_10m"])
print(subset_df.head(5))
```

```
shape: (5, 3)
```

temperature_2m	precipitation	wind_gusts_10m
f64	f64	f64
-8.6	0.0	40.0
-9.1	0.0	39.2
-9.8	0.0	39.6
-10.5	0.0	40.3
-11.2	0.0	39.6

In [3]:
```
#Chained Transformations

import polars as pl
df=pl.read_csv("toronto_weather.csv")

result = (
    df
    .with_columns((pl.col("temperature_2m") - pl.col("apparent_temperature")).al
    .filter(pl.col("feels_like_diff") > 0)
    .sort("feels_like_diff", descending=True)
    .select(["temperature_2m", "apparent_temperature", "feels_like_diff"])
)
print(result)
```

shape: (179_481, 3)

temperature_2m	apparent_temperature	feels_like_diff
---	---	---
f64	f64	f64
-13.6	-24.6	11.0
-13.7	-24.6	10.9
-14.0	-24.7	10.7
0.1	-10.4	10.5
-12.9	-23.4	10.5
…	…	…
21.7	21.6	0.1
19.2	19.1	0.1
19.2	19.1	0.1
17.7	17.6	0.1
19.2	19.1	0.1

In [4]:
```
print(df.schema)
```

```
Schema({'time': String, 'temperature_2m': Float64, 'apparent_temperature': Float6
4, 'precipitation': Float64, 'wind_speed_10m': Float64, 'snow_depth': Float64, 's
unshine_duration': Float64, 'direct_radiation': Float64, 'wind_gusts_10m': Float6
4})
```

In [8]:
```
daily_summary = (
    df.select([
        pl.col("temperature_2m").mean().alias("avg_temp"),
        pl.col("precipitation").sum().alias("total_precip"),
        pl.col("sunshine_duration").sum().alias("total_sunshine")
    ])
)

print(daily_summary)
```

shape: (1, 3)

avg_temp	total_precip	total_sunshine
---	---	---
f64	f64	f64
8.514634	20922.5	2.7245e8

In [9]:
```
#Rolling Window(Moving Averages)

df_rolling_avg=df.with_columns(
                    pl.col("temperature_2m")
                    .rolling_mean(window_size=3)
                    .alias("rolling_avg_temp")
)
print(df_rolling_avg.head(5))
```

```
shape: (5, 10)
┌──────────────┬──────────────┬──────────────┬──────────────┬───┬──────────────┬──────────┬──────────┐
│ time         ┆ temperature_ ┆ apparent_tem ┆ precipitati ┆ … ┆ sunshine_du ┆ direct_radi ┆ wind_gusts_ ┆ rolling_avg │
│ ---          ┆ 2m           ┆ perature     ┆ on          ┆   ┆ ration      ┆ ation       ┆ 10m         ┆ _temp       │
│ str          ┆ ---          ┆ ---          ┆ ---         ┆   ┆ ---         ┆ ---         ┆ ---         ┆ ---         │
│              ┆ f64          ┆ f64          ┆ f64         ┆   ┆ f64         ┆ f64         ┆ f64         ┆ f64         │
╞══════════════╪══════════════╪══════════════╪═════════════╪═══╪══════════════╪══════════╪══════════╡
│ 2000-02-02T0 ┆ -8.6         ┆ -15.0        ┆ 0.0         ┆ … ┆ 0.0         ┆ 0.0         ┆ 40.0        ┆ null        │
│ 0:00         ┆              ┆              ┆             ┆   ┆             ┆             ┆             ┆             │
│ 2000-02-02T0 ┆ -9.1         ┆ -15.6        ┆ 0.0         ┆ … ┆ 0.0         ┆ 0.0         ┆ 39.2        ┆ null        │
│ 1:00         ┆              ┆              ┆             ┆   ┆             ┆             ┆             ┆             │
│ 2000-02-02T0 ┆ -9.8         ┆ -16.4        ┆ 0.0         ┆ … ┆ 0.0         ┆ 0.0         ┆ 39.6        ┆ -9.166667   │
│ 2:00         ┆              ┆              ┆             ┆   ┆             ┆             ┆             ┆             │
│ 2000-02-02T0 ┆ -10.5        ┆ -17.2        ┆ 0.0         ┆ … ┆ 0.0         ┆ 0.0         ┆ 40.3        ┆ -9.8        │
│ 3:00         ┆              ┆              ┆             ┆   ┆             ┆             ┆             ┆             │
│ 2000-02-02T0 ┆ -11.2        ┆ -17.9        ┆ 0.0         ┆ … ┆ 0.0         ┆ 0.0         ┆ 39.6        ┆ -10.5       │
│ 4:00         ┆              ┆              ┆             ┆   ┆             ┆             ┆             ┆             │
└──────────────┴──────────────┴──────────────┴─────────────┴───┴──────────────┴──────────┴──────────┘
```

In [11]:
```
df_percent_change=df.with_columns((pl.col("wind_gusts_10m")-pl.col("wind_gusts_1
print(df_percent_change.head(5))
```

```
shape: (5, 10)
┌──────────────┬──────────────┬──────────────┬──────────────┬───┬──────────────┬──────────┬──────────┐
│ time         ┆ temperature_ ┆ apparent_tem ┆ precipitati ┆ … ┆ sunshine_du ┆ direct_radi ┆ wind_gusts_ │
│ ---          ┆ 2m           ┆ perature     ┆ on          ┆   ┆ ration      ┆ ation       ┆ 10m         ┆ wind_gusts_ │
│ str          ┆ ---          ┆ ---          ┆ ---         ┆   ┆ ---         ┆ ---         ┆ ---         ┆ pct_change  │
│              ┆ f64          ┆ f64          ┆ f64         ┆   ┆ f64         ┆ f64         ┆ f64         ┆ f64         │
╞══════════════╪══════════════╪══════════════╪═════════════╪═══╪══════════════╪══════════╪══════════╡
│ 2000-02-02T0 ┆ -8.6         ┆ -15.0        ┆ 0.0         ┆ … ┆ 0.0          ┆ 0.0      ┆ 40.0        ┆ null        │
│ 0:00         ┆              ┆              ┆             ┆   ┆              ┆          ┆             ┆             │
│ 2000-02-02T0 ┆ -9.1         ┆ -15.6        ┆ 0.0         ┆ … ┆ 0.0          ┆ 0.0      ┆ 39.2        ┆ -1.8        │
│ 1:00         ┆              ┆              ┆             ┆   ┆              ┆          ┆             ┆             │
│ 2000-02-02T0 ┆ -9.8         ┆ -16.4        ┆ 0.0         ┆ … ┆ 0.0          ┆ 0.0      ┆ 39.6        ┆ -0.6        │
│ 2:00         ┆              ┆              ┆             ┆   ┆              ┆          ┆             ┆             │
│ 2000-02-02T0 ┆ -10.5        ┆ -17.2        ┆ 0.0         ┆ … ┆ 0.0          ┆ 0.0      ┆ 40.3        ┆ -0.3        │
│ 3:00         ┆              ┆              ┆             ┆   ┆              ┆          ┆             ┆             │
│ 2000-02-02T0 ┆ -11.2        ┆ -17.9        ┆ 0.0         ┆ … ┆ 0.0          ┆ 0.0      ┆ 39.6        ┆ -1.7        │
│ 4:00         ┆              ┆              ┆             ┆   ┆              ┆          ┆             ┆             │
└──────────────┴──────────────┴──────────────┴─────────────┴───┴──────────────┴──────────┴──────────┘
```

In [13]:
```python
#Ranking
top_radiation = (
    df.sort("direct_radiation", descending=True)
      .with_columns(
          pl.col("direct_radiation").rank(method="dense", descending=True).alias
      )
      .select(["time", "direct_radiation", "radiation_rank"])
      .head(10)   # top 10 records
)

print(top_radiation)
```

shape: (10, 3)

time	direct_radiation	radiation_rank
str	f64	u32
2006-06-10T14:00	909.0	1
2006-06-10T13:00	895.0	2
2015-06-07T14:00	895.0	2
2014-06-19T14:00	890.0	3
2009-05-17T14:00	889.0	4
2015-06-06T14:00	887.0	5
2009-06-05T14:00	884.0	6
2004-05-30T14:00	882.0	7
2013-05-25T14:00	882.0	7
2003-05-21T14:00	881.0	8

In [19]:
```
#Conditional Column Creation(Categorization)

df_categorization = df.with_columns([
    pl.when(pl.col("precipitation") > 0.1)
    .then(pl.lit("Rainy"))
    .when(pl.col("snow_depth") > 0.05)
    .then(pl.lit("Snowy"))
    .when(pl.col("temperature_2m") > 30)
    .then(pl.lit("Hot"))
    .otherwise(pl.lit("Clear"))
    .alias("weather_type")
])
print(df_categorization.head(5))
```

```
shape: (5, 10)
┌──────────────┬──────────────┬──────────────┬──────────────┬───┬──────────────┬──────────────┬──────────────┐
│ time         ┆ temperature_ ┆ apparent_tem ┆ precipitati ┆ … ┆ sunshine_du ┆ direct_radi ┆ wind_gusts_ ┆ weather_typ │
│ ---          ┆ 2m           ┆ perature     ┆ on          ┆   ┆ ration      ┆ ation       ┆ 10m         ┆ e           │
│ str          ┆ ---          ┆ ---          ┆ ---         ┆   ┆ ---         ┆ ---         ┆ ---         ┆ ---         │
│              ┆ f64          ┆ f64          ┆ f64         ┆   ┆ f64         ┆ f64         ┆ f64         ┆ str         │
╞══════════════╪══════════════╪══════════════╪═════════════╪═══╪══════════════╪══════════════╪══════════════╪══════════════╡
│ 2000-02-02T0 ┆ -8.6         ┆ -15.0        ┆ 0.0         ┆ … ┆ 0.0         ┆ 0.0         ┆ 40.0        ┆ Snowy       │
│ 0:00         ┆              ┆              ┆             ┆   ┆             ┆             ┆             ┆             │
│ 2000-02-02T0 ┆ -9.1         ┆ -15.6        ┆ 0.0         ┆ … ┆ 0.0         ┆ 0.0         ┆ 39.2        ┆ Snowy       │
│ 1:00         ┆              ┆              ┆             ┆   ┆             ┆             ┆             ┆             │
│ 2000-02-02T0 ┆ -9.8         ┆ -16.4        ┆ 0.0         ┆ … ┆ 0.0         ┆ 0.0         ┆ 39.6        ┆ Snowy       │
│ 2:00         ┆              ┆              ┆             ┆   ┆             ┆             ┆             ┆             │
│ 2000-02-02T0 ┆ -10.5        ┆ -17.2        ┆ 0.0         ┆ … ┆ 0.0         ┆ 0.0         ┆ 40.3        ┆ Snowy       │
│ 3:00         ┆              ┆              ┆             ┆   ┆             ┆             ┆             ┆             │
│ 2000-02-02T0 ┆ -11.2        ┆ -17.9        ┆ 0.0         ┆ … ┆ 0.0         ┆ 0.0         ┆ 39.6        ┆ Snowy       │
│ 4:00         ┆              ┆              ┆             ┆   ┆             ┆             ┆             ┆             │
└──────────────┴──────────────┴──────────────┴──────────────┴───┴──────────────┴──────────────┴──────────────┴──────────────┘
```

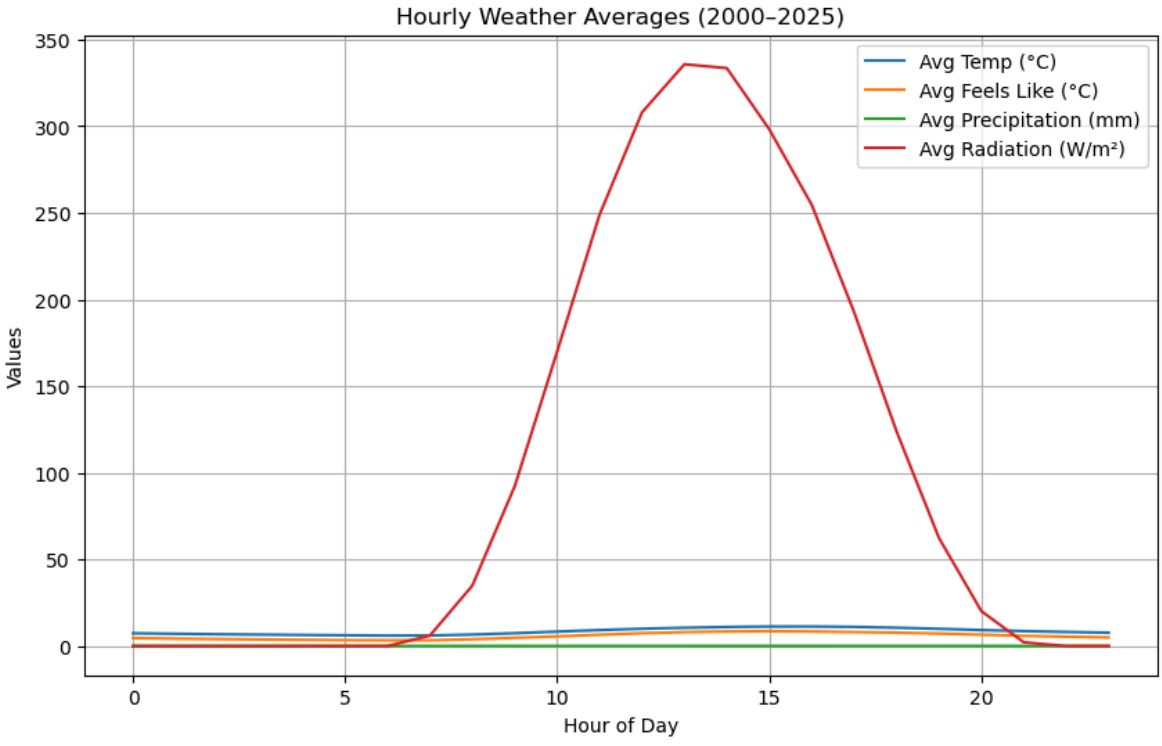
In [4]:

```python
import polars as pl
df=pl.read_csv("toronto_weather.csv")

# Extract hour from time
df_with_hour = df.with_columns(
    pl.col("time").str.strptime(pl.Datetime, "%Y-%m-%dT%H:%M").dt.hour().alias("
)

# Pivot-style aggregation: average of each hour across all days
hourly_avg = (
    df_with_hour
    .group_by("hour")
    .agg([
        pl.col("temperature_2m").mean().alias("avg_temp"),
        pl.col("apparent_temperature").mean().alias("avg_feels_like"),
        pl.col("precipitation").mean().alias("avg_precip"),
        pl.col("direct_radiation").mean().alias("avg_radiation")
    ])
    .sort("hour")
)

print(hourly_avg)

import matplotlib.pyplot as plt
```

```
# (using the hourly_avg DataFrame we built earlier)

plt.figure(figsize=(10, 6))

plt.plot(hourly_avg["hour"], hourly_avg["avg_temp"], label="Avg Temp (°C)")
plt.plot(hourly_avg["hour"], hourly_avg["avg_feels_like"], label="Avg Feels Like
plt.plot(hourly_avg["hour"], hourly_avg["avg_precip"], label="Avg Precipitation
plt.plot(hourly_avg["hour"], hourly_avg["avg_radiation"], label="Avg Radiation (

plt.title("Hourly Weather Averages (2000-2025)")
plt.xlabel("Hour of Day")
plt.ylabel("Values")
plt.legend()
plt.grid(True)
plt.show()
```

shape: (24, 5)

hour	avg_temp	avg_feels_like	avg_precip	avg_radiation
i8	f64	f64	f64	f64
0	7.409335	4.622693	0.090621	0.0
1	7.110983	4.314507	0.087759	0.0
2	6.855741	4.053497	0.088485	0.0
3	6.642361	3.831378	0.087184	0.0
4	6.431042	3.619983	0.08893	0.0
…	…	…	…	…
19	9.968654	7.212577	0.090881	62.446384
20	9.271788	6.581958	0.089515	20.165564
21	8.647392	5.910051	0.086707	2.209693
22	8.160794	5.404001	0.087976	0.0
23	7.757747	4.98764	0.088713	0.0



In [22]:
```
#Lazy Evaluation
lazy_df = (
```

```
    pl.scan_csv("toronto_weather.csv")  # File not loaded yet
    .filter(pl.col("temperature_2m") > 25)
    .select(["time","temperature_2m"])
)
```

In [23]:
```
result = lazy_df.collect()  #  Runs everything, returns a real DataFrame
print(result)
```

```
shape: (7_878, 2)
┌──────────────────┬────────────────┐
│ time             ┆ temperature_2m │
│ ---              ┆ ---            │
│ str              ┆ f64            │
╞══════════════════╪════════════════╡
│ 2000-05-06T15:00 ┆ 25.1           │
│ 2000-06-10T12:00 ┆ 25.3           │
│ 2000-06-10T13:00 ┆ 26.1           │
│ 2000-06-10T14:00 ┆ 26.7           │
│ 2000-06-10T15:00 ┆ 27.4           │
│ …                ┆ …              │
│ 2024-08-31T16:00 ┆ 26.5           │
│ 2024-08-31T17:00 ┆ 25.5           │
│ 2024-09-01T13:00 ┆ 25.1           │
│ 2024-09-01T14:00 ┆ 25.5           │
│ 2024-09-01T15:00 ┆ 25.3           │
└──────────────────┴────────────────┘
```

In [ ]: