

Subdaily Weather Scenario Generator

Documentation for GenerateSubDailyWeatherScenario.r

Overview

The **Subdaily Weather Scenario Generator** is an R script that transforms subdaily climate time series (e.g., hourly or sub-hourly data) by applying daily-level climate change scenarios. It takes existing subdaily measurements and proportionally scales them according to transformation factors derived from a daily weather scenario file.

This tool is essential for downscaling daily climate change projections to subdaily temporal resolutions while preserving the original subdaily temporal patterns (e.g., hourly precipitation distribution, diurnal temperature cycles).

Purpose

Climate change scenario files are often generated at daily timesteps through methods such as:

- Delta shift transformations
- Precipitation stretching
- Drought simulations

However, many hydrological, ecological, and environmental models require subdaily (hourly or sub-hourly) input data. This script bridges that gap by:

1. **Matching** subdaily records to daily scenario shifts by date
2. **Scaling** subdaily precipitation proportionally to daily transformations
3. **Adjusting** subdaily temperature by daily temperature offsets
4. **Preserving** the original subdaily temporal patterns and variability

Methodology

Date Matching

The date component (YYYY-MM-DD) is extracted from the subdaily DateTime field and matched to the Date field in the daily weather scenario file. Each subdaily timestep within a day receives the same transformation factors as that day.

Precipitation Transformation

Precipitation is only transformed when the subdaily OriginalPrecipitation > 0.

DeltaShift Precipitation:



$$\text{subdaily_DeltaShiftPrecipitation} = \text{subdaily_OriginalPrecipitation} \times (\text{daily_DeltaShiftPrecipitation} / \text{daily_OriginalPrecipitation})$$

Scenario Precipitation:



$$\text{subdaily_ScenarioPrecipitation} = \text{subdaily_OriginalPrecipitation} \times (\text{daily_ScenarioPrecipitation} / \text{daily_OriginalPrecipitation})$$

This approach ensures:

- The subdaily precipitation pattern (timing and distribution) is preserved
- The daily total after transformation matches the daily scenario value
- Zero precipitation values remain zero

Temperature Transformation

Temperature is adjusted for **all** subdaily records.

DeltaShift Temperature:



$$\text{subdaily_DeltaShiftTemperature} = \text{subdaily_OriginalTemperature} + (\text{daily_DeltaShiftTemperature} - \text{daily_OriginalTemperature})$$

This approach:

- Shifts the entire subdaily temperature profile by the same offset
- Preserves the diurnal temperature cycle shape
- Maintains subdaily temperature variability

Input Files

1. Subdaily Time Series

Default filename: short_subdaily.csv

Required columns (case-insensitive):

- DateTime: Timestamp in format YYYY-MM-DD HH:MM:SS or YYYY-MM-DD HH:MM
- Precipitation: Subdaily precipitation values (mm or other units)
- Air_temperature: Subdaily temperature values (°C)

Example:



DateTime,Precipitation,Air_temperature

2020-01-01 00:00:00,0.5,5.2

2020-01-01 01:00:00,0.8,5.0

2020-01-01 02:00:00,0.3,4.8

2. Daily Weather Scenario File

Default filename: dailyWeatherScenario.csv

Required columns (case-insensitive):

- Date: Date in format YYYY-MM-DD
- OriginalPrecipitation: Daily total precipitation from original data
- DeltaShiftPrecipitation: Daily precipitation after delta shifts
- OriginalTemperature: Daily mean temperature from original data
- DeltaShiftTemperature: Daily mean temperature after delta shifts
- ScenarioPrecipitation: Daily precipitation after full scenario transformation

Example:



Date,OriginalPrecipitation,DeltaShiftPrecipitation,OriginalTemperature,DeltaShiftTemperature,ScenarioPrecipitation

2020-01-01,12.5,13.75,10.2,11.2,14.8

2020-01-02,8.3,9.13,9.8,10.8,9.5

3. Daily Scenario Metadata (Optional)

Default filename: dailyWeatherScenario.json

Contains metadata about the daily scenario (scenario name, parameters, monthly offsets, etc.). This information is incorporated into the subdaily metadata output.

Output Files

1. Subdaily Scenario CSV

Default filename: subDailyWeatherScenario.csv

Columns:

- DateTime: Original subdaily timestamp
- OriginalPrecipitation: Original subdaily precipitation
- DeltaShiftPrecipitation: Delta-shifted subdaily precipitation
- OriginalTemperature: Original subdaily temperature
- DeltaShiftTemperature: Delta-shifted subdaily temperature
- ScenarioPrecipitation: Full scenario subdaily precipitation

2. Subdaily Scenario Metadata JSON

Default filename: subDailyWeatherScenario.json

Contains:

- **All metadata from the daily scenario JSON** (scenario name, parameters, monthly offsets, etc.)
- **Subdaily generation information:**
 - Generation timestamp
 - Input file names
 - Output file names
- **Subdaily summary statistics:**
 - Total records
 - DateTime range
 - Precipitation totals (original, delta-shifted, scenario)
 - Temperature means (original, delta-shifted)
 - Records with precipitation

Usage

Installation

Ensure R is installed along with the required packages. The script will automatically install missing packages.

Required R packages:

- lubridate - for date/time handling
- jsonlite - for JSON file operations

Basic Usage



r

```
# Source the script  
source("GenerateSubDailyWeatherScenario.r")
```

```
# Run with default parameters  
result <- generate_subdaily_weather_scenario()
```

This will look for:

- short_subdaily.csv (subdaily input)
- dailyWeatherScenario.csv (daily shifts)
- dailyWeatherScenario.json (daily metadata)

And create:

- subDailyWeatherScenario.csv (subdaily output)
- subDailyWeatherScenario.json (subdaily metadata)

Custom File Paths



r

```
result <- generate_subdaily_weather_scenario(  
  subdaily_file = "my_hourly_data.csv",  
  daily_shifts_file = "my_daily_scenario.csv",  
  output_csv = "my_subdaily_scenario.csv",  
  output_json = "my_subdaily_metadata.json",  
  daily_json_file = "my_daily_metadata.json"  
)
```

Function Parameters

Parameter	Default	Description
subdaily_file	"short_subdaily.csv"	Path to input subdaily CSV file
daily_shifts_file	"dailyWeatherScenario.csv"	Path to daily shifts CSV file
output_csv	"subDailyWeatherScenario.csv"	Path to output subdaily CSV file
output_json	"subDailyWeatherScenario.json"	Path to output JSON metadata file
daily_json_file	"dailyWeatherScenario.json"	Path to daily JSON metadata file

Return Value

The function returns a data frame (invisibly) containing the subdaily scenario data. This can be captured for further analysis:



r

```
subdaily_scenario <- generate_subdaily_weather_scenario()  
  
# Access the data  
head(subdaily_scenario)  
summary(subdaily_scenario)
```

Technical Details

Case-Insensitive Column Matching

All column name matching is performed case-insensitively. The script will correctly identify columns named:

- DateTime, datetime, DATETIME
- Precipitation, precipitation, PRECIPITATION
- Air_temperature, air_temperature, Temperature, etc.

DateTime Parsing

The script attempts to parse DateTime in two formats:

- 1. YYYY-MM-DD HH:MM:SS
- 2. YYYY-MM-DD HH:MM

If your data uses a different format, you may need to pre-process it.

Missing Data Handling

- If a subdaily date cannot be matched to the daily shifts file, a warning is issued
- Records with missing values in critical fields are handled gracefully
- The script reports the number of unmatched records

Mass Balance Preservation

When subdaily values are aggregated back to daily totals, they will match the daily scenario totals (within floating-point precision). This ensures physical consistency.

Zero Precipitation

Subdaily timesteps with zero precipitation remain zero in all output scenarios. This preserves the wet/dry pattern of the subdaily data.

Workflow Example

Step 1: Generate Daily Scenario

First, generate a daily weather scenario using one of the daily scenario scripts:



r

```
source("SimulateDailyDrought.r")
daily_result <- drought_simulation_with_shifts(
  input_file = "short.csv",
  output_file = "dailyWeatherScenario.csv"
)
```

This creates:

- dailyWeatherScenario.csv
- dailyWeatherScenario.json

Step 2: Apply to Subdaily Data



r

```
source("GenerateSubDailyWeatherScenario.r")
subdaily_result <- generate_subdaily_weather_scenario(
  subdaily_file = "short_subdaily.csv",
  daily_shifts_file = "dailyWeatherScenario.csv"
)
```

This creates:

- subDailyWeatherScenario.csv
- subDailyWeatherScenario.json

Step 3: Verify Results



r

```
# Check summary statistics
summary(subdaily_result)

# Verify daily totals match
library(dplyr)
subdaily_daily_totals <- subdaily_result %>%
  mutate(Date = as.Date(DateTime)) %>%
  group_by(Date) %>%
  summarize(
    subdaily_total_precip = sum(ScenarioPrecipitation),
    subdaily_mean_temp = mean(DeltaShiftTemperature)
  )

# Compare with daily scenario file
daily_scenario <- read.csv("dailyWeatherScenario.csv")
```

Error Messages and Troubleshooting

"Subdaily file not found"

Solution: Ensure the subdaily CSV file exists in the working directory or provide the full path.

"Could not find 'DateTime' column"

Solution: Verify that your subdaily file contains a column with DateTime, datetime, or similar name (case-insensitive).

"Could not parse DateTime column"

Solution: Ensure your DateTime column uses format YYYY-MM-DD HH:MM:SS or YYYY-MM-DD HH:MM.

"X subdaily records could not be matched to daily shifts"

Solution: This warning indicates that some subdaily dates are not present in the daily shifts file. Ensure the daily shifts file covers the entire subdaily date range.

"Daily shifts file must contain: Date, OriginalPrecipitation..."

Solution: Verify that your daily shifts file has all required columns with correct names (case-insensitive).

Summary Statistics Output

The script prints comprehensive statistics:



=== SUMMARY STATISTICS ===

Total subdaily records: 8760

Date range: 2020-01-01 00:00:00 to 2020-12-31 23:00:00

Precipitation:

Original total: 1245.67

DeltaShift total: 1370.24

Scenario total: 1456.89

Records with precipitation: 3421

Temperature:

Original mean: 10.45 °C

DeltaShift mean: 11.45 °C

Mean change: 1.00 °C

Best Practices

1. **Always verify date coverage:** Ensure your daily scenario file covers all dates in the subdaily data
2. **Check units:** Verify that precipitation and temperature units are consistent between files
3. **Validate outputs:** Aggregate subdaily back to daily and compare with the daily scenario file
4. **Preserve metadata:** The JSON files provide complete traceability - keep them with your data
5. **Document your workflow:** Note which daily scenario method was used (drought, stretch, etc.)

Integration with Other Scripts

This subdaily generator is designed to work with the output of daily scenario scripts:

- **SimulateDailyDrought.r** - Drought simulation with delta shifts
- **StretchDailyPrecipitation.r** - Precipitation stretching with delta shifts

All of these scripts produce output files compatible with the subdaily generator.

Limitations

1. **Temporal resolution:** The script does not create new subdaily timesteps; it only transforms existing ones
 2. **Spatial resolution:** No spatial interpolation is performed; each location must have its own subdaily file
 3. **Date format:** DateTime must be in a parseable format (YYYY-MM-DD HH:MM:SS or similar)
 4. **Temporal patterns:** The relative subdaily pattern (e.g., hourly distribution) is preserved; the script does not modify the timing of events
-

Contact and Support

For questions, issues, or contributions related to this script, please refer to the project repository documentation.

Version History

- **v1.0** - Initial release
 - Basic subdaily transformation
 - Precipitation scaling and temperature adjustment
 - JSON metadata generation
 - Case-insensitive column matching
-

License

This script is part of the weather scenario generation toolkit and follows the same license as the parent project.

Last updated: 2025