stretchDailyPrecipitation.r Documentation

Overview

`stretchDailyPrecipitation.r` is an R script that transforms daily precipitation and temperature time series data through a two-stage process combining climate change projections with extreme event intensification. The script emphasizes extreme precipitation events while maintaining mass balance (total precipitation sum remains constant).

What the Code Does

Two-Stage Transformation Process

Stage 1: Delta Shifts (Climate Change Projections)

- Applies monthly temperature offsets to all days
- Applies monthly precipitation percentage changes only to days with precipitation > 0
- Represents baseline climate change projections

Stage 2: Precipitation Stretching (Extreme Event Intensification)

- Calculates cumulative distribution (z-values) for precipitation data
- Applies sigmoid-based stretch function to precipitation above a specified threshold percentile
- Uses four optimized parameters (a, b, c, d) to maintain mass balance
- Intensifies extreme precipitation events while preserving total precipitation sum

Mass Balance Preservation

The script ensures that:

```
sum(stretched_precipitation) = sum(shifted_precipitation)
```

This is achieved through iterative optimization using the Nelder-Mead algorithm, which adjusts four parameters until the convergence tolerance is met.

How to Use

Function Signature

```r
```

```
stretch_precipitation_with_offsets(
  input_file,
  offset_file = "MonthlyDeltaShifts.csv",
  output_file = "dailyWeatherScenario.csv",
  threshold,
  stretch_factor,
  scenario_name = "Default Scenario",
  date_format = "%Y-%m-%d",
  tolerance = 0.01,
  max_iter = 1000
)
```

Parameters

Parameter	Type	Default	Description
`input_file`	character	*required*	Path to input CSV file with date, precipitation, and temperature columns
`offset_file`	character	`"MonthlyDeltaShifts.csv"`	Path to CSV file with monthly offsets (Month, PPctChange, Toffset)
`output_file`	character	`"dailyWeatherScenario.csv"`	Path for output CSV file
`threshold`	numeric	*required*	Threshold percentile (0-100) above which to stretch precipitation
`stretch_factor`	numeric	*required*	Maximum stretch percentage (e.g., 50 for 50% increase at extreme)
`scenario_name`	character	`"Default Scenario"`	Name of the weather scenario for metadata
`date_format`	character	`"%Y-%m-%d"`	Date parsing format string
`tolerance`	numeric	`0.01`	Convergence tolerance for mass balance (0.01 = 1%)
`max_iter`	numeric	`1000`	Maximum iterations for optimization algorithm

Input File Requirements

Main Input File (e.g., `short.csv`):

- **date**: Date column (case-insensitive; accepts variants like "datetime", "time")
- **precipitation**: Precipitation values (handles typos like "precipitaition", "precip")
- **air_temperature**: Temperature values in °C (handles variants like "temp", "temperature", "air_temp")

Offset File (e.g., `MonthlyDeltaShifts.csv`):

- **Month**: Integer 1-12 (January through December)
- **PPctChange**: Precipitation percent change for each month (can be positive or negative)

- **Toffset**: Temperature offset in °C for each month
- Must contain exactly 12 rows (one for each month)

Basic Usage Example

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

# Run with standard parameters
result <- stretch_precipitation_with_offsets(
  input_file = "short.csv",
  threshold = 95,       # Stretch precipitation above 95th percentile
  stretch_factor = 50   # Maximum 50% increase at extreme values
)
```

Advanced Usage Example

```r
# Custom scenario with different parameters
result <- stretch_precipitation_with_offsets(
  input_file = "my_climate_data.csv",
  offset_file = "RCP85_2050_deltas.csv",
  output_file = "extreme_scenario_2050.csv",
  threshold = 90,          # Stretch above 90th percentile
  stretch_factor = 75,     # Maximum 75% increase
  scenario_name = "RCP 8.5 - 2050 High Extremes",
  date_format = "%Y/%m/%d", # Custom date format
  tolerance = 0.005,       # Tighter convergence (0.5%)
  max_iter = 2000          # More iterations if needed
)
```

Output Files

1. Main Output CSV File

Contains the following columns:

Column	Description
`Date`	Date of observation
`OriginalPrecipitation`	Original precipitation values from input
`DeltaShiftPrecipitation`	Precipitation after applying monthly delta shifts (Stage 1)
`StretchedPrecipitation`	Final stretched precipitation (Stage 2)
`StretchFactor`	Actual stretch factor applied to each day (0 to stretch_factor%)
`ZValue`	Cumulative distribution percentile (0-100) for each precipitation value
`OriginalTemperature`	Original temperature values from input
`DeltaShiftTemperature`	Temperature after applying monthly offsets

2. Metadata JSON File

A JSON file (`output_file_metadata.json`) containing:

- Input/output file paths
- Monthly offsets applied
- User-specified parameters (threshold, stretch_factor, etc.)
- Optimized parameters (a, b, c, d)
- Precipitation transformation summary
  - Original total
  - Shifted total
  - Stretched total
- Temperature transformation summary
  - Original mean
  - Shifted mean
- Convergence statistics
- Date range processed

Algorithm Details

Step-by-Step Process

1. **Read and validate input data**
   - Load CSV files with flexible column name matching
   - Parse dates using multiple format attempts
   - Validate required columns exist
2. **Apply monthly delta shifts (Stage 1)**

- For each month (1-12):
    - Add temperature offset to all days
    - Multiply precipitation by (100 + PPctChange)/100 for days with precip > 0
- Calculate and display shift summary statistics

3. **Calculate cumulative distribution**
    - Compute z-values (0-100 percentiles) for all precipitation values
    - Used to identify which values exceed the threshold
4. **Optimize stretch parameters (Stage 2)**
    - Initialize parameters: a = 1.0, b = 1.0, c = 1.0, d = 1.0
    - Use Nelder-Mead optimization to find parameters that maintain mass balance
    - Objective: minimize |sum(stretched) - sum(shifted)| / sum(shifted)
5. **Apply sigmoid stretch function**
    - For precipitation above threshold: apply smooth stretch based on z-value
    - Stretch increases gradually from threshold to maximum stretch_factor
    - Parameters (a, b, c, d) control the shape of the sigmoid curve
6. **Validate convergence**
    - Check that |sum(stretched) - sum(shifted)| / sum(shifted) < tolerance
    - Warn if convergence not achieved
7. **Write output files**
    - Save transformed data to CSV
    - Save metadata and parameters to JSON
    - Display comprehensive summary statistics

Convergence and Mass Balance

The optimization ensures that the total precipitation sum is preserved within the specified tolerance:

```r
convergence_error = |sum(stretched) - sum(shifted)| / sum(shifted)
```

If `convergence_error > tolerance`, a warning is issued but the results are still returned.

Required R Packages

The script automatically checks for and installs required packages:

- **lubridate**: For flexible date parsing and manipulation
- **jsonlite**: For writing metadata JSON files

Base R functions are used for:

- CSV reading/writing (`read.csv`, `write.csv`)
- Optimization (`optim` with Nelder-Mead method)
- Statistical calculations

Console Output

The script provides detailed progress information:

1. File reading confirmation
2. Date parsing format used
3. Monthly offset application (for each month 1-12)
4. Precipitation shift summary
5. Temperature shift summary
6. Optimization progress
7. Optimized parameter values (a, b, c, d)
8. Convergence results
9. Final transformation summary
10. Output file locations

Error Handling

The script validates inputs and provides helpful error messages:

- **Missing required columns**: Identifies which columns are missing
- **Invalid threshold**: Must be between 0 and 100
- **Invalid stretch_factor**: Must be >= 0
- **Date parsing failures**: Tries multiple formats before failing
- **Missing offset file**: Checks file exists before reading
- **Incomplete offset data**: Ensures all 12 months are present
- **Convergence issues**: Warns if tolerance not met

Tips for Best Results

Choosing Threshold

- **90-95th percentile**: Typical for emphasizing extreme events
- **Higher threshold (95-99)**: Only affects the most extreme events
- **Lower threshold (80-90)**: Affects more precipitation days

Choosing Stretch Factor

- **25-50%**: Moderate intensification of extremes
- **50-100%**: Strong intensification
- **>100%**: Very aggressive stretching (use with caution)

Convergence Issues

If convergence tolerance is not met:

- Increase `max_iter` (try 2000-5000)
- Relax `tolerance` (try 0.02 or 0.05)
- Check that input data is reasonable (no extreme outliers)

Performance

- Typical runtime: 10-60 seconds for 10,000-30,000 daily records
- Most time spent in optimization loop
- Larger datasets or tighter tolerances require more time

Example Workflow

```
r
```

```r
# 1. Source the script
source("stretchDailyPrecipitation.r")

# 2. Prepare input files
#    - short.csv (daily climate data)
#    - MonthlyDeltaShifts.csv (monthly climate change deltas)

# 3. Run transformation for moderate extreme event scenario
moderate_result <- stretch_precipitation_with_offsets(
  input_file = "short.csv",
  threshold = 95,
  stretch_factor = 50,
  scenario_name = "Moderate Extremes - 2050"
)

# 4. Run transformation for severe extreme event scenario
severe_result <- stretch_precipitation_with_offsets(
  input_file = "short.csv",
  output_file = "severe_extremes_2050.csv",
  threshold = 90,
  stretch_factor = 100,
  scenario_name = "Severe Extremes - 2050"
)

# 5. Compare results
summary(moderate_result$StretchedPrecipitation)
summary(severe_result$StretchedPrecipitation)
```

Comparison to Other Scripts

This script differs from `drought_sim_delta.r`:

Feature	stretchDailyPrecipitation.r	drought_sim_delta.r
Stage 2 Focus	Extreme event intensification	Seasonal drought simulation
Method	Sigmoid stretch above threshold	Seasonal redistribution (spring/summer → fall/winter)
Use Case	Modeling intensified precipitation extremes	Modeling drought impacts
Mass Balance	Preserves total precipitation	Preserves total precipitation
Parameters	threshold, stretch_factor	drought_factor

Both scripts apply the same Stage 1 delta shifts, but differ in how they transform precipitation in Stage 2.

License