Package: ustats (via r-universe)

October 17, 2026

Type Package

Title R Interface to Python Tools for Computing Higher-Order U-Statistics

Version 0.1.5

Description Provides an R interface to the Python package 'u-stats' <https://pypi.org/project/u-stats/> for efficient computation of higher-order U-statistics using Einstein summation notation, implementing the methods of Chen, Zhang, and Liu (2025) <doi:10.48550/arXiv.2508.12627>. The package automatically converts R objects to 'NumPy' or 'PyTorch' tensors via 'reticulate' and supports GPU acceleration when 'PyTorch' with 'CUDA' is available. Python dependencies are declared via 'reticulate' and can be installed automatically on first use. Designed for large-scale statistical estimation where numerical stability and performance are critical.

License MIT + file LICENSE

Encoding UTF-8

Imports reticulate (>= 1.41)

SystemRequirements Python (>= 3.11); Python packages: u-stats, numpy, torch (installed automatically on first use, or via setup_ustats())

Config/reticulate packages: - u-stats - numpy - torch

Suggests knitr, rmarkdown, rstudioapi, testthat (>= 3.0.0)

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/testthat/edition 3

URL https://cxy0714.github.io/U-Statistics-R/, https://github.com/cxy0714/U-Statistics-R, https://pypi.org/project/u-stats/

BugReports https://github.com/cxy0714/U-Statistics-R/issues

Language en-US

Config/pak/sysreqs libpng-dev python3

Repository https://cxy0714.r-universe.dev

Date/Publication 2026-06-19 08:48:20 UTC

RemoteUrl https://github.com/cxy0714/u-statistics-r

RemoteRef HEAD

RemoteSha cf40c189a684b710c40ed3c9bbfc90c58b205bd1

Contents

check_ustats_setup 2
setup_ustats 3
ustat 4

Index **6**

`check_ustats_setup` *Check ustats Python Environment Status*

Description

Reports whether Python and required modules for `ustat()` are available, including the detected PyTorch version and whether CUDA (GPU acceleration) can be used.

Usage

```
check_ustats_setup()
```

Details

Note that with **reticulate** (>= 1.41), calling this function may initialize Python and trigger the automatic, one-time provisioning of the declared Python dependencies if no Python environment is configured yet (this can involve a sizeable download the first time).

Value

Invisibly returns TRUE if environment is ready

Examples

```
## Not run:
check_ustats_setup()

## End(Not run)
```

setup_ustats *Set Up Python Environment for ustats*

Description

Installs and configures the Python environment required to run `ustat()`, including `u_stats`, `numpy`, and `torch`.

Usage

```
setup_ustats(
  method = c("auto", "virtualenv", "conda", "system"),
  envname = "r-ustats",
  gpu = FALSE,
  restart = FALSE,
  persist = FALSE
)
```

Arguments

`method`	Installation method for Python: • `"auto"` (default): use existing Python or install Miniconda • `"virtualenv"`: create a virtual environment • `"conda"`: create a conda environment • `"system"`: use system Python
`envname`	Name of the virtualenv/conda environment (default: `"r-ustats"`)
`gpu`	Logical; if `FALSE` (default), install the CPU-only build of PyTorch from the official PyTorch wheel index (`https://download.pytorch.org/whl/cpu`). The CPU build is much smaller (roughly 200 MB instead of more than 2 GB with bundled CUDA libraries on Linux) and is sufficient for machines without an NVIDIA GPU. Set `gpu = TRUE` to install the default PyPI build of PyTorch, which includes CUDA support on Linux; for GPU builds on Windows, or for a specific CUDA version, see https://pytorch.org/get-started/locally/.
`restart`	Logical; whether to restart the R session after setup
`persist`	Logical; if `TRUE`, print the `RETICULATE_PYTHON` configuration line that you can add to your `.Rprofile` yourself to make the environment persist across sessions. The function never writes to your files (default: FALSE)

Details

Most users do not need to call this function. With **reticulate** (>= 1.41), the Python dependencies declared by this package are provisioned automatically in a cached environment the first time Python is used (e.g. on the first call to `ustat()`). Call `setup_ustats()` only if you prefer a persistent, dedicated environment, or if you want to control how PyTorch is installed (see the `gpu` argument).

Note: PyTorch is strongly recommended. The NumPy backend is slower and may be numerically less stable for higher-order U-statistics.

Value

Invisibly returns `TRUE` if setup completed and the environment verifies, `FALSE` otherwise.

Examples

```
## Not run:
setup_ustats()                # CPU-only PyTorch (small, default)
setup_ustats(gpu = TRUE)      # default PyPI PyTorch (CUDA on Linux)
setup_ustats(method = "conda", envname = "ustats-env")

## End(Not run)
```

---

ustat *Compute a Higher-Order U-Statistic via Python*

Description

Computes a higher-order U-statistic from precomputed kernel tensors using the Python package `u_stats`. This function serves as an R interface and handles automatic data conversion via **reticulate**.

Usage

```
ustat(
  tensors,
  expression,
  backend = c("torch", "numpy"),
  average = TRUE,
  dtype = NULL
)
```

Arguments

`tensors`	A list of numeric vectors, matrices, or arrays representing kernel evaluations. All tensors must have compatible dimensions.
`expression`	Either a character string in Einstein notation or a list of numeric vectors of length 1 or 2 describing index structure.
`backend`	Character string specifying the computation backend: `"torch"` (default) or `"numpy"`.
`average`	Logical; if `TRUE` (default), return the averaged U-statistic. Otherwise returns the raw sum.
`dtype`	Optional character string specifying numeric precision for tensors converted from R. Must be one of `"float32"` or `"float64"`. If `NULL` (default), precision is chosen automatically: • `float32` when using the Torch backend with CUDA available • `float64` otherwise

Details

The U-statistic structure can be specified using either:

- An Einstein summation string (e.g. `"ab,bc->"`), or
- A nested list of index vectors (e.g. `list(c(1,2), c(2,3))`)

This function requires a working Python environment with the `u_stats` package installed. With **reticulate** (>= 1.41) the required Python packages are provisioned automatically the first time Python is used, so no manual setup is needed in most cases. To create a persistent environment instead (or to choose between the CPU-only and CUDA builds of PyTorch), use `setup_ustats()`; use `check_ustats_setup()` to verify the configuration.

R numeric objects are converted to NumPy arrays using the selected precision. If Python tensors (e.g., Torch tensors) are supplied directly, they are passed through unchanged.

Value

A numeric scalar containing the computed U-statistic.

Examples

```
## Not run:
setup_ustats()

v1 <- runif(100)
H1 <- matrix(runif(100), 10, 10)
H2 <- matrix(runif(100), 10, 10)

ustat(list(H1, H2), "ab,bc->")
ustat(list(H1, H2), "ab,bc->", dtype = "float32")
ustat(list(H1, H2), "ab,bc->", dtype = NULL)  # auto precision

## End(Not run)
```

Index

check_ustats_setup, 2

setup_ustats, 3

ustat, 4