# Package: tdigest (via r-universe)

June 4, 2026

**Type** Package

**Title** Wicked Fast, Accurate Quantiles Using t-Digests

**Version** 0.4.3

**Date** 2026-05-05

**Description** The t-Digest construction algorithm, by Dunning, (2019) <doi:10.48550/arXiv.1902.04023>, uses a variant of 1-dimensional k-means clustering to produce a very compact data structure that allows accurate estimation of quantiles. This t-Digest data structure can be used to estimate quantiles, compute other rank statistics or even to estimate related measures like trimmed means. The advantage of the t-Digest over previous digests for this purpose is that the t-Digest handles data with full floating point resolution. The accuracy of quantile estimates produced by t-Digests can be orders of magnitude more accurate than those produced by previous digest algorithms. Methods are provided to create and update t-Digests and retrieve quantiles from the accumulated distributions.

**URL** https://git.sr.ht/~hrbrmstr/tdigest

**BugReports** https://todo.sr.ht/~hrbrmstr/tdigest

**Copyright** file inst/COPYRIGHTS

**Encoding** UTF-8

**License** MIT + file LICENSE

**Suggests** testthat, covr, spelling

**Depends** R (>= 3.5.0)

**Imports** magrittr, stats

**RoxygenNote** 7.3.1

**Language** en-US

**NeedsCompilation** yes

**Author** Bob Rudis [aut, cre] (ORCID: <https://orcid.org/0000-0001-5670-2640>), Ted Dunning [aut]

(t-Digest algorithm; <https://github.com/tdunning/t-digest/>),
Andrew Werner [aut] (Original C+ code;
<https://github.com/ajwerner/tdigest>)

**Maintainer** Bob Rudis <bob@rud.is>

**Repository** https://cran.r-universe.dev

**Date/Publication** 2026-05-05 16:59:33 UTC

**RemoteUrl** https://github.com/cran/tdigest

**RemoteRef** HEAD

**RemoteSha** 921bf4e4b218946a2e465e8f807b94427da8ffe5

# Contents



---

`as.list.tdigest` *Serialize a tdigest object to an R list or unserialize a serialized tdigest list back into a tdigest object*

---

## Description

These functions make it possible to create & populate a tdigest, serialize it out, read it in at a later time and continue populating it enabling compact distribution accumulation & storage for large, "continuous" datasets.

## Usage

```
## S3 method for class 'tdigest'
as.list(x, ...)

as_tdigest(x)
```

## Arguments

| | |
|---|---|
| `x` | a tdigest object or a tdigest_list object |
| `...` | unused |

## Examples

```
set.seed(1492)
x <- sample(0:100, 1000000, replace = TRUE)
td <- tdigest(x, 1000)
as_tdigest(as.list(td))
```

---

`td_add` *Add a value to the t-Digest with the specified count*

---

## Description

Add a value to the t-Digest with the specified count

## Usage

```
td_add(td, val, count)
```

## Arguments

| | |
|---|---|
| `td` | t-Digest object |
| `val` | value |
| `count` | count |

## Value

the original, updated `tdigest` object

## Examples

```
td <- td_create(10)
td_add(td, 0, 1)
```

---

`td_create` *Allocate a new histogram*

---

## Description

Allocate a new histogram

## Usage

```
td_create(compression = 100)

is_tdigest(td)
```

### Arguments

| | |
|---|---|
| compression | the input compression value; should be >= 1.0; this will control how aggressively the t-Digest compresses data together. The original t-Digest paper suggests using a value of 100 for a good balance between precision and efficiency. It will land at very small (think like 1e-6 percentile points) errors at extreme points in the distribution, and compression ratios of around 500 for large data sets (~1 million datapoints). Defaults to 100. |
| td | t-digest object |

### Value

a tdigest object

### References

Computing Extremely Accurate Quantiles Using t-Digests

### Examples

```
td <- td_create(10)
```

---

## td_merge *Merge one t-Digest into another*

---

### Description

Merge one t-Digest into another

### Usage

```
td_merge(from, into)
```

### Arguments

| | |
|---|---|
| from, into | t-Digests |

### Value

into

a tdigest object

## td_quantile_of *Return the quantile of the value*

### Description

Return the quantile of the value

### Usage

```
td_quantile_of(td, val)
```

### Arguments

| | |
|---|---|
| td | t-Digest object |
| val | value |

### Value

the computed quantile (double)

## td_total_count *Total items contained in the t-Digest*

### Description

Total items contained in the t-Digest

### Usage

```
td_total_count(td)

## S3 method for class 'tdigest'
length(x)
```

### Arguments

| | |
|---|---|
| td | t-Digest object |
| x | a tdigest object |

### Value

double containing the size of the t-Digest

## Examples

```
td <- td_create(10)
td_add(td, 0, 1)
td_total_count(td)
length(td)
```

---

`td_value_at` *Return the value at the specified quantile*

---

## Description

Return the value at the specified quantile

## Usage

```
td_value_at(td, q)

## S3 method for class 'tdigest'
x[i, ...]
```

## Arguments

| | |
|---|---|
| `td` | t-Digest object |
| `q` | quantile (range 0:1) |
| `x` | a tdigest object |
| `i` | quantile (range 0:1) |
| `...` | unused |

## Value

the computed quantile (`double`)

## Examples

```
td <- td_create(10)

td_add(td, 0, 1) %>%
  td_add(10, 1)

td_value_at(td, 0.1)
td_value_at(td, 0.5)
td[0.1]
td[0.5]
```

`tquantile` *Calculate sample quantiles from a t-Digest*

## Description

Calculate sample quantiles from a t-Digest

## Usage

```
tquantile(td, probs)

## S3 method for class 'tdigest'
quantile(x, probs = seq(0, 1, 0.25), ...)
```

## Arguments

| | |
|---|---|
| `td` | t-Digest object |
| `probs` | numeric vector of probabilities with values in range 0:1 |
| `x` | numeric vector whose sample quantiles are wanted |
| `...` | unused |

## Value

a `numeric` vector containing the requested quantile values

## References

Computing Extremely Accurate Quantiles Using t-Digests

## Examples

```
set.seed(1492)
x <- sample(0:100, 1000000, replace = TRUE)
td <- tdigest(x, 1000)
tquantile(td, c(0, .01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99, 1))
quantile(td)
```

# Index