

# Package ‘tidydp’

May 8, 2026

**Type** Package

**Title** Tidy Differential Privacy

**Version** 0.1.0

**Description** A tidy-style interface for applying differential privacy to data frames. Provides pipe-friendly functions to add calibrated noise, compute private statistics, and track privacy budgets using the epsilon-delta differential privacy framework. Implements the Laplace mechanism (Dwork et al. 2006 <[doi:10.1007/11681878\\_14](https://doi.org/10.1007/11681878_14)>) and the Gaussian mechanism for achieving differential privacy as described in Dwork and Roth (2014) <[doi:10.1561/0400000042](https://doi.org/10.1561/0400000042)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Imports** magrittr, stats

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

**VignetteBuilder** knitr

**URL** <https://github.com/ttarler/tidydp>

**BugReports** <https://github.com/ttarler/tidydp/issues>

**NeedsCompilation** no

**Author** Thomas Tarler [aut, cre]

**Maintainer** Thomas Tarler <[ttarler@gmail.com](mailto:ttarler@gmail.com)>

**Repository** CRAN

**Date/Publication** 2025-11-27 19:00:02 UTC

## Contents

check_privacy_budget . . . . .	2
dp_add_noise . . . . .	2
dp_count . . . . .	3
dp_mean . . . . .	4
dp_sum . . . . .	5
new_privacy_budget . . . . .	6
print.privacy_budget . . . . .	6

**Index****8**


---

check\_privacy\_budget    *Check Privacy Budget*

---

**Description**

Checks if a proposed operation would exceed the privacy budget

**Usage**

```
check_privacy_budget(budget, epsilon_required, delta_required = 0)
```

**Arguments**

budget                    A privacy budget object  
 epsilon\_required            Epsilon required for the operation  
 delta\_required    Delta required for the operation (default: 0)

**Value**

Logical indicating if budget is sufficient

**Examples**

```
budget <- new_privacy_budget(epsilon_total = 1.0)
check_privacy_budget(budget, epsilon_required = 0.5)
```

---

dp\_add\_noise                    *Add Differentially Private Noise to Data Frame Columns*

---

**Description**

Adds calibrated Laplace or Gaussian noise to specified numeric columns in a data frame to achieve differential privacy. This is the primary function for column-level privacy.

**Usage**

```
dp_add_noise(
  data,
  columns,
  epsilon,
  delta = NULL,
  lower = NULL,
  upper = NULL,
  mechanism = NULL,
  .budget = NULL
)
```

**Arguments**

data	A data frame
columns	Character vector of column names to add noise to
epsilon	Privacy parameter (smaller = more privacy, more noise)
delta	Privacy parameter for Gaussian mechanism (default: NULL, uses Laplace)
lower	Named numeric vector of lower bounds for each column
upper	Named numeric vector of upper bounds for each column
mechanism	Either "laplace" or "gaussian" (auto-selected based on delta if NULL)
.budget	Optional privacy budget object to track expenditure

**Value**

Data frame with noise added to specified columns

**Examples**

```
data <- data.frame(age = c(25, 30, 35, 40), income = c(50000, 60000, 70000, 80000))
private_data <- data %>%
  dp_add_noise(
    columns = c("age", "income"),
    epsilon = 0.1,
    lower = c(age = 0, income = 0),
    upper = c(age = 100, income = 200000)
  )
```

---

dp\_count

*Differentially Private Count*


---

**Description**

Computes a differentially private count of rows, optionally grouped by specified columns.

**Usage**

```
dp_count(data, epsilon, delta = NULL, group_by = NULL, .budget = NULL)
```

**Arguments**

data	A data frame
epsilon	Privacy parameter
delta	Privacy parameter (default: NULL, uses Laplace mechanism)
group_by	Character vector of column names to group by (optional)
.budget	Optional privacy budget object to track expenditure

**Value**

Data frame with (possibly grouped) counts

**Examples**

```
data <- data.frame(city = c("NYC", "LA", "NYC", "LA", "NYC"),
                  age = c(25, 30, 35, 40, 45))

# Overall count
dp_count(data, epsilon = 0.1)

# Grouped count
data %>% dp_count(epsilon = 0.1, group_by = "city")
```

---

dp_mean	<i>Differentially Private Mean</i>
---------	------------------------------------

---

**Description**

Computes a differentially private mean of a numeric column.

**Usage**

```
dp_mean(
  data,
  column,
  epsilon,
  delta = NULL,
  lower = NULL,
  upper = NULL,
  group_by = NULL,
  .budget = NULL
)
```

**Arguments**

data	A data frame
column	Column name to compute mean of
epsilon	Privacy parameter
delta	Privacy parameter (default: NULL, uses Laplace mechanism)
lower	Lower bound of the data range
upper	Upper bound of the data range
group_by	Character vector of column names to group by (optional)
.budget	Optional privacy budget object to track expenditure

**Value**

Data frame with (possibly grouped) private means

**Examples**

```
data <- data.frame(city = c("NYC", "LA", "NYC", "LA"),
                  income = c(50000, 60000, 70000, 80000))
data %>% dp_mean("income", epsilon = 0.1, lower = 0, upper = 200000, group_by = "city")
```

---

dp_sum	<i>Differentially Private Sum</i>
--------	-----------------------------------

---

**Description**

Computes a differentially private sum of a numeric column.

**Usage**

```
dp_sum(
  data,
  column,
  epsilon,
  delta = NULL,
  lower = NULL,
  upper = NULL,
  group_by = NULL,
  .budget = NULL
)
```

**Arguments**

data	A data frame
column	Column name to compute sum of
epsilon	Privacy parameter
delta	Privacy parameter (default: NULL, uses Laplace mechanism)
lower	Lower bound of the data range
upper	Upper bound of the data range
group_by	Character vector of column names to group by (optional)
.budget	Optional privacy budget object to track expenditure

**Value**

Data frame with (possibly grouped) private sums

**Examples**

```
data <- data.frame(city = c("NYC", "LA", "NYC", "LA"),
                  sales = c(100, 200, 150, 250))
data %>% dp_sum("sales", epsilon = 0.1, lower = 0, upper = 1000, group_by = "city")
```

---

```
new_privacy_budget    Create a New Privacy Budget
```

---

**Description**

Initializes a privacy budget tracker for managing epsilon and delta across multiple differentially private operations. The budget uses composition theorems to track cumulative privacy loss.

**Usage**

```
new_privacy_budget(epsilon_total, delta_total = 1e-05, composition = "basic")
```

**Arguments**

```
epsilon_total    Total epsilon budget available
delta_total      Total delta budget available (default: 1e-5)
composition      Method for budget composition: "basic" or "advanced" (default: "basic")
```

**Value**

A privacy budget object (list with class "privacy\_budget")

**Examples**

```
budget <- new_privacy_budget(epsilon_total = 1.0, delta_total = 1e-5)
```

---

```
print.privacy_budget    Print Privacy Budget
```

---

**Description**

Print Privacy Budget

**Usage**

```
## S3 method for class 'privacy_budget'
print(x, ...)
```

**Arguments**

- x                    A privacy budget object
- ...                 Additional arguments (unused)

**Value**

Returns the privacy budget object invisibly. Called primarily for the side effect of printing budget information to the console, including total epsilon and delta budgets, amounts spent, remaining budget, composition method, and number of operations executed.

# Index

`check_privacy_budget`, [2](#)

`dp_add_noise`, [2](#)

`dp_count`, [3](#)

`dp_mean`, [4](#)

`dp_sum`, [5](#)

`new_privacy_budget`, [6](#)

`print.privacy_budget`, [6](#)