Package: tidylaslog (via r-universe)

May 21, 2026

Type Package

Title Tidy Import, Indexing, and Export of LAS Well Log Data

Version 0.1.2

Description Provides tools for reading, parsing, indexing, and exporting LAS (Log ASCII Standard) well log files into tidy, analysis-ready tabular formats. The package separates LAS header information and log data into structured components, builds a searchable index across collections of LAS files, and enables reproducible subsetting of wells based on metadata or curve availability. Output tables can be written to CSV or Parquet formats to support large-scale statistical, machine learning, and earth science workflows. The tidy data structure follows Wickham (2014) <doi:10.18637/jss.v059.i10>. The LAS file structure follows the Canadian Well Logging Society LAS standard <https://www.cwls.org/wp-content/uploads/2017/02/Las2_Update_Jan2017.pdf>.

License MIT + file LICENSE

Encoding UTF-8

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports dplyr, tibble, tidyr

Suggests arrow, jsonlite, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL https://omodolor.github.io/tidylaslog/

BugReports https://github.com/Omodolor/tidylaslog/issues

Config/pak/sysreqs libicu-dev

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

Date/Publication 2026-02-17 08:58:19 UTC

RemoteUrl https://github.com/omodolor/tidylaslog

RemoteRef HEAD

RemoteSha d2df627bce4ceec2ce3bf6d06815987ba6c84dc6

Contents

`available_curves` *List available curve mnemonics in an index*

Description

List available curve mnemonics in an index

Usage

```
available_curves(index, county = NULL, top_n = NULL)
```

Arguments

`index`	Output of index_laslogs()
`county`	Optional county filter (character vector)
`top_n`	If not NULL, return only the top N most common curves

Value

Tibble with MNEM and n (count of wells containing the curve)

Examples

```
td <- tempdir()
f <- file.path(td, "a.las")

las_text <- c(
  " ~Version Information",
  " VERS. 2.0:",
  " WRAP. NO:",
  " ~Well Information",
  " STRT.M 1000:",
  " STOP.M 1001:",
  " STEP.M 1:",
  " NULL. -999.25:",
  " API . 1111111111:",
  " CNTY. TEST:",
  " ~Curve Information",
  " DEPT.M:",
  " GR.API:",
  " RHOB.G/C3:",
  " ~ASCII Log Data",
  " 1000 80 2.35",
  " 1001 82 2.36"
)

writeLines(las_text, f)
idx <- index_laslogs(td)
available_curves(idx, top_n = 5)
```

`batch_export_laslogs` *Index, filter, pull, and export LAS logs in one call*

Description

Index, filter, pull, and export LAS logs in one call

Usage

```
batch_export_laslogs(
  dir,
  out_dir,
  county = NULL,
  curves_any = NULL,
  curves_all = NULL,
  curves = NULL,
  output = c("wide", "long"),
  prefix = NULL,
  csv = TRUE,
  parquet = TRUE,
  write_index = TRUE,
```

```
  index_prefix = NULL
)
```

Arguments

dir	Folder containing .las files
out_dir	Output directory (absolute path, or relative to dir)
county	Optional county filter (character vector)
curves_any	Optional: keep wells with at least one of these curves
curves_all	Optional: keep wells with all of these curves
curves	Optional: curves to actually export (defaults to curves_all, else curves_any, else NULL=all)
output	"wide" or "long"
prefix	Optional file prefix. If NULL, an informative prefix is built.
csv	Write CSV?
parquet	Write Parquet?
write_index	If TRUE, also export wells_index/curves_index/files_index tables
index_prefix	Optional prefix for index files (defaults to prefix__index)

Value

Invisibly returns a list with index, apis, data, output paths, and manifest

Examples

```
td <- tempdir()
f <- file.path(td, "a.las")

las_text <- c(
  " ~Version Information",
  " VERS. 2.0:",
  " WRAP. NO:",
  " ~Well Information",
  " STRT.M 1000:",
  " STOP.M 1001:",
  " STEP.M 1:",
  " NULL. -999.25:",
  " API . 1111111111:",
  " CNTY. TEST:",
  " ~Curve Information",
  " DEPT.M:",
  " GR.API:",
  " ~ASCII Log Data",
  " 1000 80",
  " 1001 82"
)

writeLines(las_text, f)
```

```
res <- batch_export_laslogs(
  dir = td,
  out_dir = file.path(td, "exports"),
  county = "TEST",
  curves_any = "GR",
  output = "wide",
  csv = TRUE,
  parquet = FALSE,
  write_index = TRUE
)
names(res)
```

`index_laslogs` *Build a FAIR index for a folder of LAS files*

Description

Build a FAIR index for a folder of LAS files

Usage

```
index_laslogs(dir)
```

Arguments

`dir`	Folder containing .las files

Value

A list with wells_index, curves_index, files_index

Examples

```
td <- tempdir()
f1 <- file.path(td, "a.las")
f2 <- file.path(td, "b.las")

las_text <- c(
  " ~Version Information",
  " VERS. 2.0:",
  " WRAP. NO:",
  " ~Well Information",
  " STRT.M 1000:",
  " STOP.M 1001:",
  " STEP.M 1:",
  " NULL. -999.25:",
  " API . 1111111111:",
  " CNTY. TEST:",
  " ~Curve Information",
```

```
  " DEPT.M:",
  " GR.API:",
  " ~ASCII Log Data",
  " 1000 80",
  " 1001 82"
)

writeLines(las_text, f1)
writeLines(sub("1111111111", "2222222222", las_text), f2)

idx <- index_laslogs(td)
names(idx)
```

`pull_laslogs` *Pull log data for selected wells (optionally selected curves)*

Description

Pull log data for selected wells (optionally selected curves)

Usage

```
pull_laslogs(index, apis, curves = NULL, output = c("long", "wide"))
```

Arguments

`index`	Output of index_laslogs()
`apis`	Character vector of API values to load
`curves`	Optional curve mnemonics to keep (e.g., c("GR","RHOB","NPHI"))
`output`	"long" (tidy) or "wide" (ML-ready)

Value

A tibble combining all selected wells

Examples

```
td <- tempdir()
f <- file.path(td, "a.las")

las_text <- c(
  " ~Version Information",
  " VERS. 2.0:",
  " WRAP. NO:",
  " ~Well Information",
  " STRT.M 1000:",
  " STOP.M 1001:",
  " STEP.M 1:",
  " NULL. -999.25:",
```

```
  " API . 1111111111:",
  " CNTY. TEST:",
  " ~Curve Information",
  " DEPT.M:",
  " GR.API:",
  " ~ASCII Log Data",
  " 1000 80",
  " 1001 82"
)

writeLines(las_text, f)
idx <- index_laslogs(td)
dat <- pull_laslogs(idx, apis = "1111111111", curves = "GR", output = "long")
head(dat)
```

`read_laslog` *Read a LAS well log file (Log ASCII Standard) into a structured object*

Description

tidylaslog supports two equivalent representations of LAS log data:

Usage

```
read_laslog(file, output = c("long", "wide"))
```

Arguments

`file`	Path to a .las file
`output`	Output format:
	"wide" One row per depth per well, curves as columns (ML- and spreadsheet-ready).
	"long" One row per curve measurement with columns depth, mnemonic, and value (tidy format).

Details

- **Wide format**: one row per depth step per well, with each curve stored as a separate column.
- **Long format**: one row per measurement, with curve names stored in a `mnemonic` column and values in a `value` column.

Both formats contain the same information but are optimized for different workflows.

Value

An S3 object of class "laslog" with VERSION/WELL/CURVE/PARAMETER/OTHER/LOG

Examples

```
las_text <- c(
  " ~Version Information",
  " VERS. 2.0: CWLS LOG ASCII STANDARD",
  " WRAP. NO:",
  " ~Well Information",
  " STRT.M 1000: Start depth",
  " STOP.M 1002: Stop depth",
  " STEP.M 1: Step",
  " NULL. -999.25: Null value",
  " API . 1111111111: API number",
  " CNTY. TEST: County",
  " ~Curve Information",
  " DEPT.M: Depth",
  " GR.API: Gamma Ray",
  " ~ASCII Log Data",
  " 1000 80",
  " 1001 82",
  " 1002 79"
)
f <- tempfile(fileext = ".las")
writeLines(las_text, f)
x <- read_laslog(f, output = "long")
head(x$LOG)
```

---

read_laslog_header *Read LAS header only (no ~A data)*

Description

Read LAS header only (no ~A data)

Usage

```
read_laslog_header(file)
```

Arguments

`file`	Path to a .las file

Value

S3 object of class "laslog_header" with VERSION/WELL/CURVE/PARAMETER/OTHER plus provenance

Examples

```
las_text <- c(
  " ~Version Information",
  " VERS. 2.0: CWLS LOG ASCII STANDARD",
  " WRAP. NO:",
  " ~Well Information",
  " STRT.M 1000: Start depth",
  " STOP.M 1001: Stop depth",
  " STEP.M 1: Step",
  " NULL. -999.25: Null value",
  " API . 1111111111: API number",
  " CNTY. TEST: County",
  " ~Curve Information",
  " DEPT.M: Depth",
  " GR.API: Gamma Ray",
  " ~ASCII Log Data",
  " 1000 80",
  " 1001 82"
)
f <- tempfile(fileext = ".las")
writeLines(las_text, f)
h <- read_laslog_header(f)
names(h)
```

---

select_laslogs *Select wells from an index by metadata and curve availability*

Description

Select wells from an index by metadata and curve availability

Usage

```
select_laslogs(index, county = NULL, curves_any = NULL, curves_all = NULL)
```

Arguments

index	Output of index_laslogs()
county	Character vector of counties to keep (optional)
curves_any	Keep wells that have at least one of these curves (optional)
curves_all	Keep wells that have all of these curves (optional)

Value

Character vector of API values

Examples

```
td <- tempdir()
f <- file.path(td, "a.las")

las_text <- c(
  " ~Version Information",
  " VERS. 2.0:",
  " WRAP. NO:",
  " ~Well Information",
  " STRT.M 1000:",
  " STOP.M 1001:",
  " STEP.M 1:",
  " NULL. -999.25:",
  " API . 1111111111:",
  " CNTY. TEST:",
  " ~Curve Information",
  " DEPT.M:",
  " GR.API:",
  " ~ASCII Log Data",
  " 1000 80",
  " 1001 82"
)

writeLines(las_text, f)
idx <- index_laslogs(td)
apis <- select_laslogs(idx, county = "TEST", curves_any = "GR")
apis
```

---

tidylaslog *Universal entry point for reading, indexing, and exporting LAS well logs*

Description

`tidylaslog()` works with either a **single LAS file** or a **directory of LAS files**. It can return data directly to R or export analysis-ready tables to disk.

Usage

```
tidylaslog(
  x,
  county = NULL,
  curves_any = NULL,
  curves_all = NULL,
  curves = NULL,
  output = c("wide", "long"),
  out_dir = NULL,
  prefix = NULL,
  formats = c("csv", "parquet"),
```

```
  write_index = TRUE,
  write_meta = TRUE,
  meta_sections = c("VERSION", "WELL", "CURVE", "PARAMETER", "OTHER"),
  manifest = TRUE
)
```

Arguments

x	Path to a .las file OR a directory containing .las files.
county	Optional county filter (directory mode only).
curves_any	Keep wells that contain *at least one* of these curves (directory mode).
curves_all	Keep wells that contain *all* of these curves (directory mode).
curves	Curves to actually keep/export. Defaults to curves_all, then curves_any, otherwise all curves.
output	Output format: **"wide"** One row per depth per well, curves as columns (ML- and spreadsheet-ready). **"long"** One row per curve measurement with columns depth, mnemonic, and value (tidy format).
out_dir	If NULL, data are returned to R only. If provided, outputs are written to this directory. If relative (e.g. "exports"), it is created inside x when x is a directory.
prefix	Optional filename prefix for exported files.
formats	Output formats to write. One or both of "csv" and "parquet".
write_index	Write index tables (wells, curves, files) when exporting directories?
write_meta	Write metadata tables (WELL, CURVE, etc.) for single-file exports?
meta_sections	Which metadata sections to export ("VERSION", "WELL", "CURVE", "PARAMETER", "OTHER").
manifest	Write a JSON manifest describing the export?

Details

The function supports two equivalent representations of LAS log data:

- **Wide format**: one row per depth step per well, with each curve stored as a separate column.
- **Long format**: one row per measurement, with curve names stored in a mnemonic column and values in a value column.

Both formats contain the same information but are optimized for different workflows (machine learning vs tidy analysis).

Value

If out_dir is NULL:

Single file An S3 object of class "laslog" containing VERSION, WELL, CURVE, PARAMETER, OTHER, and LOG.

Directory A list with `index`, `apis`, and combined `data`.

If `out_dir` is provided:

Single file A list containing exported data paths, metadata paths, and an optional manifest.

Directory The full batch export result (see `batch_export_laslogs()`).

Examples

```
# ---- Single file mode (return to R) ----
las_text <- c(
  " ~Version Information",
  " VERS. 2.0:",
  " WRAP. NO:",
  " ~Well Information",
  " STRT.M 1000:",
  " STOP.M 1002:",
  " STEP.M 1:",
  " NULL. -999.25:",
  " API . 1111111111:",
  " CNTY. TEST:",
  " ~Curve Information",
  " DEPT.M:",
  " GR.API:",
  " ~ASCII Log Data",
  " 1000 80",
  " 1001 82",
  " 1002 79"
)
f <- tempfile(fileext = ".las")
writeLines(las_text, f)
obj <- tidylaslog(f, output = "long")
head(obj$LOG)

# ---- Directory mode (return to R) ----
td <- tempdir()
f1 <- file.path(td, "a.las")
f2 <- file.path(td, "b.las")
writeLines(las_text, f1)
writeLines(sub("1111111111", "2222222222", las_text), f2)
res <- tidylaslog(td, county = "TEST", curves_any = "GR", output = "wide")
names(res)

# ---- Export mode (CSV only, no arrow needed) ----
out_dir <- file.path(td, "exports_demo")
ex <- tidylaslog(td,
  county = "TEST",
  curves_any = "GR",
  output = "wide",
  out_dir = out_dir,
  formats = "csv",
  write_index = TRUE,
  manifest = FALSE
```

```
)
names(ex)
```

`write_laslogs` *Write LAS logs to CSV and/or Parquet*

Description

Write LAS logs to CSV and/or Parquet

Usage

```
write_laslogs(data, out_dir, prefix = "laslogs", csv = TRUE, parquet = TRUE)
```

Arguments

`data`	Tibble returned by pull_laslogs()
`out_dir`	Output directory
`prefix`	File prefix (no extension)
`csv`	Write CSV file?
`parquet`	Write Parquet file? (requires arrow)

Value

Invisibly returns output paths

Examples

```
out_dir <- tempdir()
df <- data.frame(api = "1111111111", depth = c(1000, 1001), GR = c(80, 82))
paths <- write_laslogs(df, out_dir = out_dir, prefix = "demo", csv = TRUE, parquet = FALSE)
paths
```

Index