

Package: tooth (via r-universe)

July 23, 2026

Title Dental Public Health Indices and Odontogram Visualizations

Version 0.5.0

Description Computes dental caries indices (DMFT, DMFS, dmft, dmfs) from surface-level clinical examination data and produces odontogram heatmap visualizations of per-tooth-surface outcomes. Supports primary and permanent dentition with configurable teeth per quadrant (5 to 8), separate root and coronal caries tallying, long and wide input formats, stratified output, and FDI/Universal/quadrant tooth numbering conversion.

License MIT + file LICENSE

URL <https://github.com/ddmsel/tooth>

BugReports <https://github.com/ddmsel/tooth/issues>

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports dplyr (>= 1.0.0), ggplot2 (>= 3.4.0), patchwork, purrr, rlang, tibble

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author David Selvaraj [aut, cre, cph]
(<https://orcid.org/0000-0003-4055-9493>), Suchitra Nelson
[aut]

Maintainer David Selvaraj <dms256@case.edu>

Repository <https://ddmsel.r-universe.dev>

Date/Publication 2026-07-13 21:15:39 UTC
RemoteUrl <https://github.com/ddmsel/tooth>
RemoteRef HEAD
RemoteSha cfcbdbb8a9e925005dba5bad3821eadeff1dfeda

Contents

build_odontogram	2
calc_dmfs	5
calc_dmft	6
draw_tooth	8
pivot_to_long	9
sim_exam	10
summarise_dmft	11
tooth_config	12
tooth_convert	12
Index	14

build_odontogram	<i>Build a surface-level odontogram heatmap</i>
------------------	---

Description

Draws a full-arch odontogram with colour-coded tooth surfaces. Supports primary and permanent dentition (5–8 teeth per quadrant), selective surfaces, stratification, summary statistics with significance testing, and configurable tooth numbering.

Usage

```
build_odontogram(  
  data,  
  value_col = "prop",  
  dentition = "permanent",  
  teeth_per_quadrant = NULL,  
  title = "Surface Odontogram",  
  subtitle = NULL,  
  color_low = "#FFFFFF",  
  color_high = "#C62828",  
  na_color = "grey90",  
  min_val = NULL,  
  max_val = NULL,  
  legend_title = "Proportion",  
  show_roots = TRUE,  
  surfaces = c("buc", "lin", "mes", "dis", "occ", "rootb", "rootl", "rootm", "rootd"),  
  show_labels = TRUE,
```

```

    label_size = 1.8,
    tooth_label_size = 3,
    numbering = c("quadrant", "fdi", "universal"),
    strata = NULL,
    strata_labels = NULL,
    stats = NULL,
    stats_test = NULL,
    stats_var = NULL,
    stats_raw = NULL,
    footnote = NULL,
    combine = TRUE,
    ncol = 1
  )

```

Arguments

<code>data</code>	Data frame with at least <code>tooth_num</code> (e.g. "ur1"), <code>tooth_surface</code> (e.g. "buc", "occ", "rootb"), and a numeric value column. If stratifying, include the column named in <code>strata</code> .
<code>value_col</code>	Name of the numeric column to map to fill colour.
<code>dentition</code>	"permanent" or "primary".
<code>teeth_per_quadrant</code>	Integer 5–8 (default derived from dentition).
<code>title, subtitle</code>	Plot title/subtitle.
<code>color_low, color_high</code>	Gradient endpoints (default "#FFFFFF" to "#C62828").
<code>na_color</code>	Fill for missing surfaces (default "grey90").
<code>min_val, max_val</code>	Lower and upper limits for the colour scale. Both auto-detected when NULL. Set manually to fix the range across multiple plots.
<code>legend_title</code>	Legend title.
<code>show_roots</code>	Logical; draw root caries bars.
<code>surfaces</code>	Character vector of surfaces to draw.
<code>show_labels</code>	Logical; show surface abbreviation labels (B, L, M, D, O, RB, RL, RM, RD) inside each surface polygon.
<code>label_size</code>	Numeric; size of surface labels.
<code>tooth_label_size</code>	Numeric; size of tooth number labels (default 3).
<code>numbering</code>	Tooth numbering system for display: "quadrant" (default), "fdi", or "universal".
<code>strata</code>	Optional column name for stratification.
<code>strata_labels</code>	Optional named character vector to relabel strata in panel titles.
<code>stats</code>	Optional data frame of per-group summary statistics to display below each panel. Must contain a column matching <code>strata</code> and numeric columns to display (e.g. <code>n</code> , <code>mean_DMFT</code> , <code>mean_DT</code>). See Details.

stats_test	Method for comparing stats across strata: "t" for t-test (default), "wilcox" for Wilcoxon rank-sum, or NULL to skip. P-values are shown as significance stars.
stats_var	Column name in stats_raw to test for differences (e.g. "DMFT"). Required when stats_test is not NULL.
stats_raw	Optional data frame of individual-level data used for significance testing. Must contain a column matching strata and a numeric column matching stats_var. Required when stats_test is set.
footnote	Optional character string displayed below the plot as a caption. Use for abbreviation explanations and p-value definitions.
combine	Logical; if strata is set, combine panels into one plot (default TRUE) or return a named list.
ncol	Number of columns when combining stratified panels.

Details

Summary statistics: Pass a data frame to stats with one row per stratum. Example:

```
stats_df <- data.frame(
  treatment = c("SDF", "ART"),
  n = c(120, 115),
  mean_DT = c(2.3, 2.8),
  mean_DMFT = c(5.1, 5.6)
)
build_odontogram(d, strata = "treatment", stats = stats_df)
```

Significance stars: When stats_test and stats_var are set and raw data is provided via stats_raw, p-values are computed and appended: *** p < 0.001, ** p < 0.01, * p < 0.05, ns otherwise.

Value

A ggplot object (or list of ggplots if combine = FALSE).

Examples

```
library(tibble)
d <- expand_grid(
  tooth_num = paste0(rep(c("ur", "ul", "lr", "ll"), each=7), 1:7),
  tooth_surface = c("buc", "lin", "mes", "dis", "occ"),
  stringsAsFactors = FALSE
)
d$prop <- runif(nrow(d))
build_odontogram(d, teeth_per_quadrant = 7)
```

calc_dmfs

*Calculate DMFS / dmfs index from surface-level data***Description**

Computes Decayed, Missing, and Filled Surfaces per person. Can separate root from coronal surfaces. Accepts long or wide format.

Usage

```
calc_dmfs(
  data,
  format = c("long", "wide"),
  id = "record_id",
  group = NULL,
  strata = NULL,
  lesion_col = "lesion_code",
  activity_col = "act",
  filling_col = "filling_code",
  tooth_code_col = "code",
  root_lesion_col = NULL,
  root_surfaces = c("rootb", "rootl", "rootm", "rootd"),
  decayed_codes = c(3, 4, 5, 6),
  activity_codes = c(2),
  filled_codes = c(1, 2, 4, 5, 6, 7, 8),
  missing_codes = c(1),
  root_decayed_codes = NULL,
  consider_activity = TRUE,
  consider_activity_coronal = NULL,
  consider_activity_root = NULL
)
```

Arguments

data	Data frame. In long format : one row per tooth-surface with columns for id, tooth_num, tooth_surface, and clinical codes. In wide format : one row per person-tooth with surface codes in separate columns.
format	"long" (default) or "wide". See Details.
id	Person-identifier column (default "record_id").
group	Optional grouping column(s) for repeated measures, e.g. "redcap_event_name". Character vector.
strata	Optional stratification column(s), e.g. "treatment". The output will include these columns for downstream group summaries.
lesion_col, activity_col, filling_col, tooth_code_col	Column names for ICDAS codes (long format).

root_lesion_col	Optional column for root surface lesion codes. When provided, root caries (RDT) is calculated separately from coronal (DT). Set to NULL to ignore root caries (default).
root_surfaces	Character vector of surface names considered root surfaces. Default c("rootb", "rootl", "rootm", "rootr").
decayed_codes	Numeric vector of lesion codes considered decayed (default c(3, 4, 5, 6) for ICDAS).
activity_codes	Numeric vector of activity codes indicating active caries (default c(2)).
filled_codes	Numeric vector of filling codes indicating a restoration is present (default c(1, 2, 4, 5, 6, 7, 8)).
missing_codes	Numeric vector of tooth-level codes indicating the tooth is missing (default c(1)).
root_decayed_codes	Numeric vector for root caries codes (default same as decayed_codes).
consider_activity	Logical; overall default for whether a lesion must also carry an active-caries code (activity_codes) to be counted as decayed. TRUE (default) requires activity; FALSE counts any lesion in decayed_codes regardless of activity. Standard epidemiological D(3)MFT counts all cavitated lesions irrespective of activity, so set FALSE to match that convention. When FALSE, the activity column is not referenced and need not be present.
consider_activity_coronal, consider_activity_root	Optional logical overrides for the coronal and root components. NULL (default) inherits consider_activity. Use these to treat crown and root caries differently (e.g. require activity coronally but not for roots, where activity is often not assessed): consider_activity_root = FALSE.

Value

A tibble with one row per person (and group/strata) containing: DS, FS, MS, DFS, DMFS, plus optional RDS (root decayed surfaces) when root_lesion_col is set, and binary indicators.

calc_dmft

Calculate DMFT / dmft index from surface- or tooth-level data

Description

Computes Decayed, Missing, and Filled Teeth counts per person. Handles both primary and permanent dentition. Can separate root caries from coronal caries. Accepts long or wide format data.

Usage

```
calc_dmft(
  data,
  format = c("long", "wide"),
  id = "record_id",
```

```

group = NULL,
strata = NULL,
lesion_col = "lesion_code",
activity_col = "act",
filling_col = "filling_code",
tooth_code_col = "code",
root_lesion_col = NULL,
root_surfaces = c("rootb", "rootl", "rootm", "rootd"),
decayed_codes = c(3, 4, 5, 6),
activity_codes = c(2),
filled_codes = c(1, 2, 4, 5, 6, 7, 8),
present_codes = c(2, 3, 4, 5, 6, 7, 8),
missing_codes = c(1),
root_decayed_codes = NULL,
consider_activity = TRUE,
consider_activity_coronal = NULL,
consider_activity_root = NULL
)

```

Arguments

data	Data frame. In long format : one row per tooth-surface with columns for id, tooth_num, tooth_surface, and clinical codes. In wide format : one row per person-tooth with surface codes in separate columns.
format	"long" (default) or "wide". See Details.
id	Person-identifier column (default "record_id").
group	Optional grouping column(s) for repeated measures, e.g. "redcap_event_name". Character vector.
strata	Optional stratification column(s), e.g. "treatment". The output will include these columns for downstream group summaries.
lesion_col, activity_col, filling_col, tooth_code_col	Column names for ICDAS codes (long format).
root_lesion_col	Optional column for root surface lesion codes. When provided, root caries (RDT) is calculated separately from coronal (DT). Set to NULL to ignore root caries (default).
root_surfaces	Character vector of surface names considered root surfaces. Default c("rootb", "rootl", "rootm", "rootd").
decayed_codes, activity_codes, filled_codes, present_codes, missing_codes	Numeric vectors defining clinical code categories. Defaults match ICDAS. See the <i>Diagnostic threshold</i> section for decayed_codes.
root_decayed_codes	Numeric vector for root caries codes (default same as decayed_codes).
consider_activity	Logical; overall default for whether a lesion must also carry an active-carries code (activity_codes) to be counted as decayed. TRUE (default) requires activity; FALSE counts any lesion in decayed_codes regardless of activity. Standard

epidemiological D(3)MFT counts all cavitated lesions irrespective of activity, so set FALSE to match that convention. When FALSE, the activity column is not referenced and need not be present.

`consider_activity_coronal`, `consider_activity_root`

Optional logical overrides for the coronal and root components. NULL (default) inherits `consider_activity`. Use these to treat crown and root caries differently (e.g. require activity coronally but not for roots, where activity is often not assessed): `consider_activity_root = FALSE`.

Value

A tibble with one row per person (and group/strata) containing:

DT Decayed teeth (coronal)

FT Filled teeth (not decayed)

MT Missing teeth

DFT Decayed or filled teeth

DMFT Decayed + missing + filled teeth

num_teeth Count of teeth present

RDT Root-decayed teeth (only when `root_lesion_col` is set)

DT_yn, **DFT_yn** Binary indicators

Diagnostic threshold

The default `decayed_codes = c(3, 4, 5, 6)` implements the **D3 threshold**: only ICDAS codes 3–6 (localised enamel breakdown through extensive cavitation) are scored as decayed, while ICDAS 1–2 (non-cavitated initial enamel lesions) are treated as sound. This D(3)MFT convention is the one most commonly reported in caries epidemiology. To use the more sensitive D1 threshold that also counts early enamel lesions, pass `decayed_codes = c(1, 2, 3, 4, 5, 6)`.

Wide format

When `format = "wide"`, each row is one person (and optional group). You must supply columns named like `{tooth}_{surface}_{code_type}` or use the `wide_cols` parameter. Alternatively, pass a `pivot_spec` — but the easiest approach is to reshape to long with `pivot_to_long()` first.

`draw_tooth`

Draw a single tooth as polygon geometry

Description

Produces crown surface wedges (B/L/M/D/O) and up to four root surface bars (RB/RL/RM/RD), plus labels, outlines, and diagonals — all as data frames ready for `ggplot2`. Any surface can be excluded via the `surfaces` argument.

Usage

```
draw_tooth(
  tooth_id,
  quadrant,
  tooth_num,
  is_upper,
  surface_values,
  value_col = "prop",
  x_offset = 0,
  y_offset = 0,
  tooth_size = 1,
  show_roots = TRUE,
  surfaces = c("buc", "lin", "mes", "dis", "occ", "rootb", "rootl", "rootm", "rootd"),
  display_label = NULL
)
```

Arguments

tooth_id	Character label, e.g. "ur1".
quadrant	One of "ur", "ul", "lr", "ll".
tooth_num	Numeric position within the quadrant.
is_upper	Logical; TRUE for upper arch.
surface_values	Data frame with tooth_surface and a value column. Recognised surfaces: buc, lin, mes, dis, occ, rootb, rootl, rootm, rootd.
value_col	Name of the value column in surface_values.
x_offset, y_offset	Numeric offsets for positioning.
tooth_size	Numeric side length of the crown square (default 1).
show_roots	Logical; draw root surface bars (default TRUE).
surfaces	Character vector of surfaces to include. Omit any you don't want drawn.
display_label	Optional custom label for the tooth number (e.g. FDI notation "11" instead of "ur1"). When NULL, uses tooth_id.

Value

Named list of tibbles: crown, roots, labels, num_label, diags, outline.

pivot_to_long	<i>Pivot wide tooth data to long format</i>
---------------	---

Description

Helper to reshape wide-format data (one row per person-tooth, surfaces as separate columns) into the long format expected by `calc_dmft()`.

Usage

```
pivot_to_long(  
  data,  
  id = "record_id",  
  group = NULL,  
  strata = NULL,  
  lesion_suffix = "_les",  
  filling_suffix = "_fil",  
  activity_suffix = "_act",  
  code_suffix = "_code"  
)
```

Arguments

- data Wide data frame.
- id Person ID column name.
- group Optional grouping columns.
- strata Optional stratification columns.
- lesion_suffix, filling_suffix, activity_suffix
 Suffixes that identify the surface-level columns (default "_les", "_fil", "_act").
- code_suffix Suffix for tooth-level code columns (default "_code").

Details

Expects columns following the naming convention: {tooth}_{surface}_les, {tooth}_{surface}_fil, {tooth}_{surface}_act, and {tooth}_code for tooth-level status.

Value

A long-format tibble.

sim_exam	<i>Simulated dental examination data</i>
----------	--

Description

A simulated surface-level dental examination dataset for 20 patients, each with 7 teeth per quadrant (28 teeth) and 5 coronal surfaces plus 2 root surfaces per tooth. Designed for demonstrating and testing `calc_dmft()`, `calc_dmfs()`, and `build_odontogram()`.

Usage

```
sim_exam
```

Format

A data frame with 3,920 rows and 7 columns:

record_id Patient identifier (character, "P01" to "P20").

tooth_num Tooth identifier in quadrant notation (e.g. "ur1", "ll7").

tooth_surface Surface name: buc, lin, mes, dis, occ, rootb, or rootl.

lesion_code ICDAS lesion code (integer 0-6). 0 = sound, 3-6 = caries.

filling_code Filling/restoration code (integer). 0 = none, 1-8 = restored.

act Lesion activity code (integer). 1 = inactive, 2 = active.

code Tooth-level status code (integer). 1 = missing, 2-8 = present.

Examples

```
data(sim_exam)
head(sim_exam)
calc_dmft(sim_exam)
```

summarise_dmft

Summarise DMFT results by group

Description

Takes the output of `calc_dmft()` and produces a summary table with n, mean number of teeth, mean DT, mean DMFT, and optionally other columns — ready to pass directly to `build_odontogram()` via `stats` and `stats_raw`.

Usage

```
summarise_dmft(dmft_data, group_col, digits = 1)
```

Arguments

<code>dmft_data</code>	Data frame returned by <code>calc_dmft()</code> , optionally merged with grouping variables (e.g. treatment arm, visit frequency).
<code>group_col</code>	Column name to summarise by (e.g. "treatment").
<code>digits</code>	Number of decimal places for means (default 1).

Value

A list with two elements:

stats Summary data frame with one row per group, containing n, mean_teeth, mean_DT, mean_DMFT. Column names match what `build_odontogram()` expects for `stats`.

raw The input data frame, for passing to `stats_raw` in `build_odontogram()`.

Examples

```
data(sim_exam)
dmft <- calc_dmft(sim_exam)
dmft$treatment <- ifelse(
  as.numeric(gsub("P", "", dmft$record_id)) <= 10, "SDF", "ART"
)
result <- summarise_dmft(dmft, "treatment")
result$stats
# Pass directly to build_odontogram:
# build_odontogram(d, strata = "treatment",
#   stats = result$stats, stats_raw = result$raw,
#   stats_test = "wilcox", stats_var = "DMFT")
```

tooth_config	<i>Generate tooth configuration for an arch</i>
--------------	---

Description

Creates a tibble of tooth IDs for primary or permanent dentition with a configurable number of teeth per quadrant (5-8).

Usage

```
tooth_config(dentition = c("permanent", "primary"), teeth_per_quadrant = NULL)
```

Arguments

dentition Character: "permanent" (default) or "primary".
 teeth_per_quadrant Integer 5-8. Defaults to 8 for permanent, 5 for primary.

Value

A tibble with columns: quadrant, num, tooth_id, is_upper.

tooth_convert	<i>Convert between tooth numbering systems</i>
---------------	--

Description

Converts between FDI (ISO 3950), Universal (ADA), and the internal quadrant format used by this package ("ur1", "115", etc.).

Usage

```
tooth_convert(x, from = "fdi", to = "quadrant", dentition = "permanent")
```

Arguments

x	Character or numeric vector of tooth numbers.
from	Numbering system of the input: "fdi", "universal", or "quadrant".
to	Target numbering system: "fdi", "universal", or "quadrant".
dentition	"permanent" or "primary". Required when converting from/to "universal" with primary teeth.

Details

The "universal" system follows the standard ADA conventions: permanent teeth are numbered **1–32** (1 = upper-right third molar, 32 = lower-right third molar), while primary teeth use the **letters A–T** (A = upper-right second primary molar, T = lower-right second primary molar). FDI two-digit codes are 11–48 for permanent and 51–85 for primary dentition.

Value

Character vector in the target system.

Examples

```
tooth_convert("11", from = "fdi", to = "quadrant")
tooth_convert("ur1", from = "quadrant", to = "fdi")
tooth_convert(c("1", "16", "17", "32"), from = "universal", to = "fdi")
# Primary dentition uses ADA letters A-T for the universal system:
tooth_convert("ur1", from = "quadrant", to = "universal",
              dentition = "primary") # "E"
tooth_convert("K", from = "universal", to = "fdi", dentition = "primary")
```

Index

* datasets

sim_exam, [10](#)

build_odontogram, [2](#)

build_odontogram(), [10](#), [11](#)

calc_dmfs, [5](#)

calc_dmfs(), [10](#)

calc_dmft, [6](#)

calc_dmft(), [9–11](#)

draw_tooth, [8](#)

pivot_to_long, [9](#)

pivot_to_long(), [8](#)

sim_exam, [10](#)

summarise_dmft, [11](#)

tooth_config, [12](#)

tooth_convert, [12](#)