

Package: carwatch (via r-universe)

September 21, 2026

Title Processing of 'CARWatch' Sampling Logs and Saliva Data

Version 1.0.1

Description Import and reconstruct saliva-sampling studies recorded by the 'CARWatch' application. Registration metadata and raw barcode events are converted into auditable study days and scheduled sample positions using a two-pass issue-review workflow. Functions assess sampling-time compliance, merge laboratory saliva measurements, calculate response features, and create quality-control visualizations. The application is described by Richer et al. (2023)
<[doi:10.1016/j.psyneuen.2023.106073](https://doi.org/10.1016/j.psyneuen.2023.106073)>.

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

Depends R (>= 4.3)

Imports boot, clock (>= 0.7.0), digest (>= 0.6.0), dplyr (>= 1.1.0), fs (>= 1.6.0), ggplot2 (>= 3.5.0), grid, jsonlite (>= 1.8.0), readr (>= 2.1.0), rlang (>= 1.1.0), tibble (>= 3.2.0), vctrs (>= 0.6.0), withr (>= 3.0.0)

Suggests covr, DT, knitr, pkgdown, rmarkdown, roxygen2, shiny, shinytest2, testthat (>= 3.2.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL <https://carwatch-tools.github.io/carwatch-r/>,
<https://github.com/carwatch-tools/carwatch-r>

BugReports <https://github.com/carwatch-tools/carwatch-r/issues>

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake make libuv1-dev libx11-dev

Repository <https://carwatch-tools.r-universe.dev>

Date/Publication 2026-09-21 07:12:45 UTC

RemoteUrl <https://github.com/carwatch-tools/carwatch-r>

RemoteRef HEAD

RemoteSha 426b510a26b8378be44d811f06ee2e8c72d6d54f

Contents

as_sample_events	3
as_study_days	3
auc	4
compute_features	4
compute_features_from_carwatch	5
conversion_report_editor	6
convert_raw_logs	7
drop_non_compliant_samples	8
extract_registration_schedule	9
find_non_compliant_samples	10
find_sampling_anomalies	10
generate_synthetic_study_data	11
initial_value	12
interactive_sampling_timeline	13
max_increase	13
max_value	14
mean_se	14
merge_saliva	15
new_carwatch_results	16
new_sampling_compliance_checker	17
plot_compliance_overview	17
plot_saliva_curve	18
plot_sampling_timeline	19
plot_timing_deviation	20
read_conversion_report	21
read_manual_diary	21
read_raw_logs	22
read_raw_logs_from_participant_dirs	22
read_saliva	23
read_study_manager_export	24
read_study_results	24
saliva_feature_wide_to_long	25
sample_times_datetime_to_minute	25
slope	26
standard_features	26
summarize_compliance	27
summarize_protocol	28
summarize_source_audit	28
write_conversion_report	29
write_study_results	29

<code>as_sample_events</code>	3
-------------------------------	---

Index	30
--------------	-----------

<code>as_sample_events</code>	<i>Extract the explicit sample-level table</i>
-------------------------------	--

Description

Extract the explicit sample-level table

Usage

```
as_sample_events(data)
```

Arguments

`data` Complete canonical Study Results.

Value

A tibble with one row per participant, day, and scheduled sample.

<code>as_study_days</code>	<i>Extract the explicit day-level table</i>
----------------------------	---

Description

Extract the explicit day-level table

Usage

```
as_study_days(data)
```

Arguments

`data` Complete canonical Study Results.

Value

A tibble with one row per participant and canonical day.

auc	<i>Compute area under the curve with respect to ground and increase</i>
-----	---

Description

Compute area under the curve with respect to ground and increase

Usage

```
auc(
  data,
  saliva_type = "cortisol",
  remove_s0 = FALSE,
  compute_auc_post = FALSE,
  sample_times = NULL
)
```

Arguments

data	Saliva data or canonical results.
saliva_type	Measurement column name(s).
remove_s0	Whether to remove baseline sample S0.
compute_auc_post	Whether to add post-baseline AUCi.
sample_times	Sampling-time vector or column name.

Value

Group-level AUC values.

compute_features	<i>Compute the standard CARWatch saliva response feature set</i>
------------------	--

Description

Compute the standard CARWatch saliva response feature set

Usage

```
compute_features(
  data,
  saliva_type = "cortisol",
  group_levels = NULL,
  sample_level = NULL,
  sample_times = NULL,
  slope_pairs = NULL,
  remove_s0 = FALSE
)
```

Arguments

data	Saliva data or canonical results.
saliva_type	Measurement column name(s).
group_levels	Grouping column names.
sample_level	Sample-position column name.
sample_times	Sampling-time vector or column name.
slope_pairs	Sample pairs for slopes.
remove_s0	Whether to remove baseline sample S0.

Value

Group-level response features.

Examples

```
curve <- tibble::tibble(
  participant = "P01", sample = paste0("S", 1:4),
  time_min = c(0, 30, 45, 60), cortisol = c(5, 9, 8, 7)
)
compute_features(
  curve,
  group_levels = "participant",
  sample_level = "sample",
  sample_times = "time_min"
)
```

```
compute_features_from_carwatch
```

Compute CARWatch saliva features using sample position and actual time

Description

Compute CARWatch saliva features using sample position and actual time

Usage

```
compute_features_from_carwatch(  
  data,  
  saliva_type = "cortisol",  
  slope_pairs = NULL,  
  group_levels = NULL  
)
```

Arguments

data	Canonical results or sample events.
saliva_type	Measurement column name.
slope_pairs	Sample pairs for slopes.
group_levels	Additional R column names that identify separate curves.

Value

Per participant-day response features.

Examples

```
fixture <- system.file("extdata", "parity", "v1.0.0", package = "carwatch")  
results <- read_study_results(file.path(fixture, "results.csv"))  
saliva <- read_saliva(file.path(fixture, "saliva.csv"))  
merged <- merge_saliva(results, saliva)  
compute_features_from_carwatch(merged)
```

conversion_report_editor
<i>Edit conversion decisions interactively</i>

Description

The table is shown on the left and issue-specific decision controls on the right. The report remains immutable until the selected row passes the same validation used by `convert_raw_logs()`. When `raw_logs` is supplied, `Refresh remaining issues` applies the cumulative decisions to the original events with `errors = "warn"` and replaces the visible table with unresolved issues from the new conversion. A diary-backed decision that cannot be applied is reset to `Leave unresolved` while successfully applied decisions are hidden. Resolved upstream decisions remain in the history returned by `Done`. Mouse and arrow-key row selection both update the decision controls.

Usage

```
conversion_report_editor(  
  report,  
  launch = interactive(),  
  host = "127.0.0.1",  
  port = NULL,  
  raw_logs = NULL,  
  protocol_manifest = NULL,  
  sampling_schedule = NULL,  
  manual_diary = NULL,  
  check_compliance = TRUE,  
  compliance_checker = new_sampling_compliance_checker()  
)
```

Arguments

report	A conversion report list or its editable issue table.
launch	Run the Shiny gadget immediately. Set FALSE to return a shiny.appobj for tests, embedding, or deployment.
host	Host passed to <code>shiny::runApp()</code> .
port	Optional port passed to <code>shiny::runApp()</code> .
raw_logs	Optional immutable raw events. Required to enable refreshing the conversion after decisions change.
protocol_manifest, sampling_schedule, manual_diary, check_compliance, compliance_checker	Conversion settings forwarded unchanged by Refresh remaining issues.

Value

A validated issue tibble, NULL after cancellation, or a Shiny app object when launch = FALSE.

convert_raw_logs	<i>Convert immutable CARWatch raw logs to canonical Study Results</i>
------------------	---

Description

The first pass only reports issues. Decisions passed through issue_decisions are the only allowed route for corrections; raw event rows are never changed.

Usage

```
convert_raw_logs(  
  raw_logs,  
  protocol_manifest = NULL,  
  errors = c("raise", "warn", "error"),
```

```

    create_report = FALSE,
    issue_decisions = NULL,
    sampling_schedule = NULL,
    manual_diary = NULL,
    check_compliance = TRUE,
    compliance_checker = new_sampling_compliance_checker()
  )

```

Arguments

<code>raw_logs</code>	Immutable events returned by <code>read_raw_logs()</code> .
<code>protocol_manifest</code>	Optional ordered registration configuration.
<code>errors</code>	Unresolved-issue handling: "raise", "warn", or legacy "error".
<code>create_report</code>	Return a list containing results and the issue report.
<code>issue_decisions</code>	A prior editable issue report with decisions.
<code>sampling_schedule</code>	Optional fallback schedule for <code>use_default</code> sample decisions. Supply a position-indexed vector, a named sample list, or a named day list containing either form.
<code>manual_diary</code>	Optional normalized manual diary.
<code>check_compliance</code>	Whether to calculate timing compliance.
<code>compliance_checker</code>	Timing tolerance configuration.

Value

Canonical results, or a results/report list.

Examples

```

fixture <- system.file("extdata", "parity", "v1.0.0", package = "carwatch")
raw_logs <- read_raw_logs(file.path(fixture, "raw", "VP01"))
converted <- convert_raw_logs(raw_logs, errors = "warn", create_report = TRUE)
converted$report$issues
as_sample_events(converted$results)

```

`drop_non_compliant_samples`

Remove non-compliant observations

Description

Remove non-compliant observations

Usage

```
drop_non_compliant_samples(  
  data,  
  drop_entire_day = TRUE,  
  drop_unassessed = FALSE  
)
```

Arguments

data	Canonical results or sample events.
drop_entire_day	Whether one failed sample removes its day.
drop_unassessed	Whether missing assessments are removed.

Value

Filtered data preserving canonical results when supplied.

extract_registration_schedule

Extract the registration-aware protocol schedule

Description

Extract the registration-aware protocol schedule

Usage

```
extract_registration_schedule(  
  raw_logs,  
  protocol_manifest = NULL,  
  errors = c("raise", "warn", "error")  
)
```

Arguments

raw_logs	Immutable events returned by read_raw_logs() .
protocol_manifest	Optional ordered registration configuration.
errors	Unresolved-issue handling: "raise", "warn", or legacy "error".

Value

A tibble with registration-aware sample positions.

find_non_compliant_samples
<i>Identify non-compliant samples</i>

Description

Identify non-compliant samples

Usage

find_non_compliant_samples(data)

Arguments

data Canonical results or sample events.

Value

Rows with failed compliance.

find_sampling_anomalies
<i>Identify tube and day mismatches</i>

Description

Identify tube and day mismatches

Usage

find_sampling_anomalies(data)

Arguments

data Canonical results or sample events.

Value

Rows with sampling anomalies.

generate_synthetic_study_data

Generate deterministic local CARWatch example data

Description

Generated anomalies are represented by ordinary raw-log omissions and a matching manual diary/decision report. The source events themselves are never patched. Study Manager snake-case, camelCase, and QR aliases are accepted, including multiple registration blocks and opaque saliva IDs.

Usage

```
generate_synthetic_study_data(
    output_dir,
    study_config = NULL,
    n_participants = NULL,
    non_compliant_sample_ratio = 0.1,
    missing_awakening_time_ratio = 0.01,
    missing_sampling_time_ratio = 0.02,
    random_state = 42L,
    create_cortisol_data = FALSE,
    overwrite = FALSE,
    validate = TRUE
)
```

Arguments

output_dir	Target directory.
study_config	Study Manager-style list, decoded CARWATCH; QR payload, or NULL for the four-day CAR default.
n_participants	Number of generated participants. When NULL, use the configured value or the Study Manager default of 80.
non_compliant_sample_ratio	Proportion of expected samples with timing deliberately outside the default compliance tolerance. Relative and fixed clock-time samples use their respective tolerance-aware deviation ranges.
missing_awakening_time_ratio	Proportion of participant-days with a missing awakening event and first sample.
missing_sampling_time_ratio	Total proportion of expected scans omitted.
random_state	Integer seed, or NULL for unseeded generation. Separate deterministic streams are used for timings, anomaly selection, and cortisol.
create_cortisol_data	Whether to create position-indexed cortisol.csv.

- overwrite Whether an existing target directory may be replaced.
- validate Run advisory and submitted-decision conversion after writing.

Value

The normalized output directory path.

Examples

```
output <- tempfile("carwatch-study-")
generate_synthetic_study_data(
  output,
  study_config = list(study_days = 1, saliva_distances = c(0, 30)),
  n_participants = 1,
  non_compliant_sample_ratio = 0,
  missing_awakening_time_ratio = 0,
  missing_sampling_time_ratio = 0,
  validate = FALSE
)
unlink(output, recursive = TRUE)
```

<code>initial_value</code>	<i>Return the initial value in each saliva curve</i>
----------------------------	--

Description

Return the initial value in each saliva curve

Usage

```
initial_value(data, saliva_type = "cortisol", remove_s0 = FALSE)
```

Arguments

- data Saliva data or canonical results.
- saliva_type Measurement column name(s).
- remove_s0 Whether to remove baseline sample S0.

Value

Group-level initial values.

interactive_sampling_timeline

Open an interactive participant-day sampling timeline

Description

This R-native replacement for the Python notebook widget uses Shiny. The app owns participant/day selectors and redraws `plot_sampling_timeline()`.

Usage

```
interactive_sampling_timeline(
  data,
  show_expected = TRUE,
  launch = interactive(),
  host = "127.0.0.1",
  port = NULL
)
```

Arguments

<code>data</code>	Complete canonical results or a sample-event tibble.
<code>show_expected</code>	Whether registered target times are drawn.
<code>launch</code>	Run the app immediately. Set FALSE to return a <code>shiny.appobj</code> for testing, embedding, or deployment.
<code>host</code>	Host passed to <code>shiny::runApp()</code> .
<code>port</code>	Optional port passed to <code>shiny::runApp()</code> .

Value

A Shiny app object when `launch = FALSE`; otherwise the result of `shiny::runApp()`.

max_increase

Compute maximum increase from the initial sample

Description

Compute maximum increase from the initial sample

Usage

```
max_increase(
  data,
  saliva_type = "cortisol",
  remove_s0 = FALSE,
  percent = FALSE
)
```

Arguments

- data Saliva data or canonical results.
- saliva_type Measurement column name(s).
- remove_s0 Whether to remove baseline sample S0.
- percent Whether to return the percentage increase.

Value

Group-level maximum increase.

max_value	<i>Compute the maximum measured value in each saliva curve</i>
-----------	--

Description

Compute the maximum measured value in each saliva curve

Usage

```
max_value(data, saliva_type = "cortisol", remove_s0 = FALSE)
```

Arguments

- data Saliva data or canonical results.
- saliva_type Measurement column name(s).
- remove_s0 Whether to remove baseline sample S0.

Value

Group-level maximum values.

mean_se	<i>Compute mean and standard error by sample</i>
---------	--

Description

Compute mean and standard error by sample

Usage

```
mean_se(data, saliva_type = "cortisol", group_cols = NULL, remove_s0 = FALSE)
```

Arguments

data	Saliva data or canonical results.
saliva_type	Measurement column name(s).
group_cols	Grouping columns.
remove_s0	Whether to remove baseline sample S0.

Value

Mean and standard error by sample.

merge_saliva	<i>Merge laboratory saliva values into canonical Study Results</i>
--------------	--

Description

Raw tube identifiers remain opaque. With match_on = "sample" values are matched to the recorded tube (falling back to the scheduled tube); with match_on = "position" they are matched to registration-aware positions.

Usage

```
merge_saliva(
  study_results,
  saliva,
  correct_swaps = TRUE,
  match_on = c("sample", "position", "physical_id"),
  missing_carwatch_data = c("ignore", "raise"),
  metadata_cols = NULL
)
```

Arguments

study_results	Complete canonical results.
saliva	Long laboratory tibble.
correct_swaps	Match physical tubes to their recorded position.
match_on	"sample"/"physical_id" or "position".
missing_carwatch_data	Whether unmatched laboratory rows are ignored or rejected.
metadata_cols	Optional laboratory columns to retain as metadata rather than numeric measurements. When omitted, non-numeric non-key columns are treated as metadata. This is the R equivalent of additional pandas index levels. Constant participant-day values are stored at the canonical day level; values varying within any day stay sample-level everywhere.

Value

Complete canonical results with laboratory and merge-provenance fields.

Examples

```
fixture <- system.file("extdata", "parity", "v1.0.0", package = "carwatch")
results <- read_study_results(file.path(fixture, "results.csv"))
saliva <- read_saliva(file.path(fixture, "saliva.csv"))
merged <- merge_saliva(results, saliva)
as_sample_events(merged)[c("participant", "sample", "cortisol")]
```

new_carwatch_results	<i>Construct canonical CARWatch results</i>
----------------------	---

Description

A `carwatch_results` object is an R-native representation of the Python package's participant-indexed, three-level wide table. Internally it is a tibble with a participant column and safe column names. The `column_spec` attribute maps every value column reversibly to day, sample, and variable; opaque sample IDs are never encoded into the internal names.

Usage

```
new_carwatch_results(data, column_spec, display_only = FALSE)
```

Arguments

<code>data</code>	Tibble containing participant and one column per result value.
<code>column_spec</code>	Tibble with name, day, sample, and variable.
<code>display_only</code>	Whether required provenance has intentionally been removed.

Value

A `carwatch_results` object: a tibble with one row per participant, a `column_spec` attribute mapping value columns to day, sample, and variable, and a `carwatch_display_only` attribute.

`new_sampling_compliance_checker`*Configure CARWatch sampling-compliance tolerances*

Description

Configure CARWatch sampling-compliance tolerances

Usage

```
new_sampling_compliance_checker(  
  awakening_delay_tolerance_min = 5,  
  sampling_delay_tolerance_min = 5,  
  absolute_time_tolerance_min = 15,  
  check_absolute_times = TRUE  
)
```

Arguments

`awakening_delay_tolerance_min`
Tolerance for the first relative sample.

`sampling_delay_tolerance_min`
Tolerance for later relative samples.

`absolute_time_tolerance_min`
Tolerance for fixed-time samples.

`check_absolute_times`
Whether fixed-time samples are evaluated.

Value

A `carwatch_compliance_checker` list containing the configured tolerance values and the `check_absolute_times` flag.

`plot_compliance_overview`*Plot cohort sampling compliance*

Description

Plot cohort sampling compliance

Usage

```
plot_compliance_overview(
  data,
  by = "sample_position",
  view = c("proportion", "heatmap")
)
```

Arguments

data	Canonical results or sample events.
by	Summary grouping variable.
view	Plot form.

Value

A ggplot object.

Examples

```
fixture <- system.file("extdata", "parity", "v1.0.0", package = "carwatch")
results <- read_study_results(file.path(fixture, "results.csv"))
plot_compliance_overview(results)
```

plot_saliva_curve	<i>Plot CARWatch-aligned saliva response curves</i>
-------------------	---

Description

Actual time_min values define the horizontal axis. Faint lines show participant-day curves. Thick lines connect group means calculated separately at each registered sample_position; shaded ribbons show BCa bootstrap confidence intervals when at least two measurements contribute.

Usage

```
plot_saliva_curve(
  data,
  value = "cortisol",
  participant = NULL,
  day = NULL,
  group_by = NULL,
  ci = 95,
  n_boot = 1000,
  seed = 0,
  show_individual = TRUE
)
```

Arguments

data	Canonical results or sample events.
value	Non-empty name of the saliva measurement column.
participant	Optional exact participant filter.
day	Optional exact canonical-day filter.
group_by	Optional character vector defining separate aggregate curves.
ci	Bootstrap confidence level between 0 and 100, or NULL for no band.
n_boot	Positive number of bootstrap resamples.
seed	Integer random seed, or NULL to use the current R RNG state.
show_individual	Whether individual participant-day curves are drawn.

Value

A ggplot object.

Examples

```
fixture <- system.file("extdata", "parity", "v1.0.0", package = "carwatch")
results <- read_study_results(file.path(fixture, "results.csv"))
saliva <- read_saliva(file.path(fixture, "saliva.csv"))
merged <- merge_saliva(results, saliva)
plot_saliva_curve(merged, ci = NULL)
```

plot_sampling_timeline

Plot an individual sampling timeline

Description

Actual sampling events are coloured by compliance and shaped by timestamp provenance. When `show_expected = TRUE`, hollow grey circles show the static protocol targets, hollow blue diamonds show targets updated from preceding recorded relative samples, and arrows show the signed deviation from the app-updated target. Missing relative scans break the adaptive target chain. Missing events and scheduled/recorded sample mismatches remain visible.

Usage

```
plot_sampling_timeline(data, participant, day, show_expected = TRUE)
```

Arguments

data	Canonical results or sample events.
participant	Participant identifier.
day	Canonical day identifier.
show_expected	Whether expected sampling times are shown.

Value

A ggplot object.

Examples

```
fixture <- system.file("extdata", "parity", "v1.0.0", package = "carwatch")
results <- read_study_results(file.path(fixture, "results.csv"))
plot_sampling_timeline(results, participant = "VP01", day = "D1")
```

plot_timing_deviation *Plot signed sampling-time deviations*

Description

Negative values indicate early collection and positive values late collection. Boxes summarize each group while jittered points retain every recorded observation.

Usage

```
plot_timing_deviation(data, by = "sample_position")
```

Arguments

data	Canonical results or sample events.
by	Grouping variable. Defaults to sample_position.

Value

A ggplot object.

Examples

```
fixture <- system.file("extdata", "parity", "v1.0.0", package = "carwatch")
results <- read_study_results(file.path(fixture, "results.csv"))
plot_timing_deviation(results)
```

`read_conversion_report`*Read a conversion-issue report CSV*

Description

Read a conversion-issue report CSV

Usage

```
read_conversion_report(path)
```

Arguments

path CSV path.

Value

A validated editable issue tibble.

`read_manual_diary`*Read a manual measurement diary*

Description

Read a manual measurement diary

Usage

```
read_manual_diary(path)
```

Arguments

path Wide CSV diary.

Value

A tibble keyed by participant and canonical day.

read_raw_logs	<i>Read CARWatch raw log files</i>
---------------	------------------------------------

Description

Read CARWatch raw log files

Usage

```
read_raw_logs(
  path,
  tz = "Europe/Berlin",
  errors = c("raise", "warn", "ignore", "error")
)
```

Arguments

path	CSV file, ZIP archive, directory, or vector of paths.
tz	IANA timezone for Unix timestamps.
errors	Invalid-payload handling: "error", "warn", or "ignore".

Value

A tibble of immutable raw events.

Examples

```
fixture <- system.file("extdata", "parity", "v1.0.0", package = "carwatch")
raw_logs <- read_raw_logs(file.path(fixture, "raw", "VP01"))
head(raw_logs)
```

read_raw_logs_from_participant_dirs	<i>Read raw logs from one explicitly mapped folder per participant</i>
-------------------------------------	--

Description

Read raw logs from one explicitly mapped folder per participant

Usage

```
read_raw_logs_from_participant_dirs(
  participant_dirs,
  tz = "Europe/Berlin",
  errors = c("raise", "warn", "ignore", "error"),
  create_report = FALSE
)
```

Arguments

participant_dirs Named paths, one folder per participant.
tz IANA study timezone.
errors Invalid-payload handling.
create_report Whether to return the source audit.

Value

Raw events, or a raw-events/source-audit list.

Examples

```
fixture <- system.file("extdata", "parity", "v1.0.0", package = "carwatch")  
folders <- c(VP01 = file.path(fixture, "raw", "VP01"))  
imported <- read_raw_logs_from_participant_dirs(folders, create_report = TRUE)  
head(imported$source_audit)
```

read_saliva	<i>Read a BioPsyKit-style saliva CSV</i>
-------------	--

Description

Read a BioPsyKit-style saliva CSV

Usage

```
read_saliva(path, saliva_type = "cortisol")
```

Arguments

path CSV containing exactly participant, sample, and one biomarker.
saliva_type Biomarker column name.

Value

A tibble.

```
read_study_manager_export
```

Read a flat CARWatch Study Manager export

Description

Read a flat CARWatch Study Manager export

Usage

```
read_study_manager_export(path, tz = "Europe/Berlin")
```

Arguments

path	Path to a Study Manager export.
tz	IANA study timezone.

Value

Complete carwatch_results.

```
read_study_results
```

Read canonical CARWatch Study Results

Description

Reads the three-header CSV produced by either carwatch-python or write_study_results(). The returned object preserves opaque identifiers, time zones, and column placement.

Usage

```
read_study_results(path, tz = "Europe/Berlin", simple = FALSE)
```

Arguments

path	Path to a three-header Study Results CSV.
tz	IANA study timezone.
simple	Return a display-only subset.

Value

A carwatch_results object.

Examples

```
fixture <- system.file("extdata", "parity", "v1.0.0", package = "carwatch")
results <- read_study_results(file.path(fixture, "results.csv"))
as_study_days(results)
```

`saliva_feature_wide_to_long`*Convert wide saliva features to an analysis-friendly long table*

Description

Convert wide saliva features to an analysis-friendly long table

Usage

```
saliva_feature_wide_to_long(data, saliva_type = "cortisol")
```

Arguments

<code>data</code>	Feature table returned by <code>compute_features()</code> .
<code>saliva_type</code>	Biomarker prefix to select.

Value

A tibble with `saliva_feature` and biomarker value columns.

`sample_times_datetime_to_minute`*Convert clock-time sample columns to minutes from the first sample*

Description

Convert clock-time sample columns to minutes from the first sample

Usage

```
sample_times_datetime_to_minute(data, sample_cols = NULL)
```

Arguments

<code>data</code>	A data frame whose selected columns contain HH:MM or HH:MM:SS values.
<code>sample_cols</code>	Sample-time columns; all columns are used by default.

Value

A tibble of numeric minute offsets.

slope	<i>Compute the slope between two saliva samples</i>
-------	---

Description

Compute the slope between two saliva samples

Usage

```
slope(
  data,
  sample_labels = NULL,
  sample_idx = NULL,
  saliva_type = "cortisol",
  sample_times = NULL
)
```

Arguments

data	Saliva data or canonical results.
sample_labels	Two sample labels.
sample_idx	Two one-based sample indices.
saliva_type	Measurement column name(s).
sample_times	Sampling-time vector or column name.

Value

Group-level slope values.

standard_features	<i>Summarize descriptive saliva features</i>
-------------------	--

Description

Summarize descriptive saliva features

Usage

```
standard_features(
  data,
  saliva_type = "cortisol",
  group_cols = NULL,
  keep_index = TRUE
)
```

Arguments

data	Saliva data or canonical results.
saliva_type	Measurement column name(s).
group_cols	Grouping columns.
keep_index	Retained for API compatibility.

Value

Descriptive saliva features.

summarize_compliance	<i>Summarize sampling compliance</i>
----------------------	--------------------------------------

Description

Summarize sampling compliance

Usage

```
summarize_compliance(data, group_by = "sample_position")
```

Arguments

data	Canonical results or a sample-event tibble.
group_by	Grouping column(s), or NULL for a cohort total.

Value

A tibble with assessed, compliant, non-compliant, and unassessed counts.

Examples

```
fixture <- system.file("extdata", "parity", "v1.0.0", package = "carwatch")
results <- read_study_results(file.path(fixture, "results.csv"))
summarize_compliance(results)
```

summarize_protocol	<i>Summarize the reconstructed study protocol</i>
--------------------	---

Description

Summarize the reconstructed study protocol

Usage

```
summarize_protocol(  
  raw_logs,  
  protocol_manifest = NULL,  
  errors = c("raise", "warn", "error")  
)
```

Arguments

raw_logs	Immutable events returned by read_raw_logs() .
protocol_manifest	Optional ordered registration configuration.
errors	Unresolved-issue handling.

Value

A registration-level tibble.

summarize_source_audit	<i>Compactly summarize a source audit</i>
------------------------	---

Description

Compactly summarize a source audit

Usage

```
summarize_source_audit(source_audit)
```

Arguments

source_audit	Source audit returned by read_raw_logs_from_participant_dirs() .
--------------	--

Value

A one-column tibble with three values named `raw_log_import`: the number of selected logs, the total raw events in selected logs, and the number of distinct participants, in that order.

`write_conversion_report`*Write an editable conversion report*

Description

Write an editable conversion report

Usage

```
write_conversion_report(report, path)
```

Arguments

<code>report</code>	A conversion report list or its issue table.
<code>path</code>	Destination CSV path.

Value

path, invisibly.

`write_study_results` *Write canonical CARWatch Study Results*

Description

Write canonical CARWatch Study Results

Usage

```
write_study_results(data, path)
```

Arguments

<code>data</code>	Complete carwatch_results object.
<code>path</code>	Destination CSV path.

Value

path, invisibly. The function writes the canonical Study Results CSV and returns its absolute path.

Examples

```
fixture <- system.file("extdata", "parity", "v1.0.0", package = "carwatch")
results <- read_study_results(file.path(fixture, "results.csv"))
output <- tempfile(fileext = ".csv")
write_study_results(results, output)
restored <- read_study_results(output)
```

Index

as_sample_events, [3](#)
as_study_days, [3](#)
auc, [4](#)

compute_features, [4](#)
compute_features(), [25](#)
compute_features_from_carwatch, [5](#)
conversion_report_editor, [6](#)
convert_raw_logs, [7](#)
convert_raw_logs(), [6](#)

drop_non_compliant_samples, [8](#)

extract_registration_schedule, [9](#)

find_non_compliant_samples, [10](#)
find_sampling_anomalies, [10](#)

generate_synthetic_study_data, [11](#)

initial_value, [12](#)
interactive_sampling_timeline, [13](#)

max_increase, [13](#)
max_value, [14](#)
mean_se, [14](#)
merge_saliva, [15](#)

new_carwatch_results, [16](#)
new_sampling_compliance_checker, [17](#)

plot_compliance_overview, [17](#)
plot_saliva_curve, [18](#)
plot_sampling_timeline, [19](#)
plot_sampling_timeline(), [13](#)
plot_timing_deviation, [20](#)

read_conversion_report, [21](#)
read_manual_diary, [21](#)
read_raw_logs, [22](#)
read_raw_logs(), [8](#), [9](#), [28](#)

read_raw_logs_from_participant_dirs, [22](#)
read_saliva, [23](#)
read_study_manager_export, [24](#)
read_study_results, [24](#)

saliva_feature_wide_to_long, [25](#)
sample_times_datetime_to_minute, [25](#)
shiny::runApp(), [7](#), [13](#)
slope, [26](#)
standard_features, [26](#)
summarize_compliance, [27](#)
summarize_protocol, [28](#)
summarize_source_audit, [28](#)

write_conversion_report, [29](#)
write_study_results, [29](#)