

# Package ‘catool’

May 8, 2026

**Type** Package

**Title** Compensation Analysis Tool for Instructor Overload Pay

**Version** 1.0.1

**Date** 2025-06-09

**Description** Calculates equitable overload compensation for college instructors based on institutional policies, enrollment thresholds, and regular teaching load limits. Compensation is awarded only for credit hours that exceed the regular load and meet minimum enrollment criteria. When enrollment is below a specified threshold, pay is prorated accordingly. The package prioritizes compensation from high-enrollment courses, or optionally from low-enrollment courses for fairness, depending on user-defined strategy. Includes tools for flexible policy settings, instructor filtering, and produces clean, audit-ready summary tables suitable for payroll and administrative reporting.

**License** AGPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.1

**Depends** R (>= 3.5.0)

**Imports** dplyr, purrr, tibble, rlang, scales

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

**VignetteBuilder** knitr

**URL** <https://github.com/dawit3000/catool>

**BugReports** <https://github.com/dawit3000/catool/issues>

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Dawit Aberra [aut, cre]

**Maintainer** Dawit Aberra <dawit3000@hotmail.com>

**Repository** CRAN

**Date/Publication** 2025-06-11 11:40:01 UTC

## Contents

filter_schedule . . . . .	2
get_instructor_schedule . . . . .	3
get_subject_schedule . . . . .	4
get_unique_instructors . . . . .	4
ol_comp . . . . .	5
ol_comp_byindex . . . . .	6
ol_comp_summary . . . . .	7
schedule . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

filter_schedule	<i>Filter Course Schedule by College, Department, Program, Subject, and/or Instructor</i>
-----------------	---

---

## Description

Applies one or more filters to a course schedule using pattern matching on instructor, subject, college, department, and program. All matching is case-insensitive and based on partial string matching.

## Usage

```
filter_schedule(
  schedule,
  subject_pattern = NULL,
  instructor_pattern = NULL,
  college_pattern = NULL,
  department_pattern = NULL,
  program_pattern = NULL
)
```

## Arguments

schedule	A data frame containing the course schedule with required columns: INSTRUCTOR, SUBJ. Optional columns include: COLLEGE, DEPARTMENT, PROGRAM.
subject_pattern	Optional regex pattern to match subject codes (e.g., "CSCI", "^MATH").
instructor_pattern	Optional regex pattern to match instructor names (e.g., "Smith", "^Jones").
college_pattern	Optional regex pattern to match college names (e.g., "Science", "Engineering").
department_pattern	Optional regex pattern to match department names (e.g., "Math", "Biology").
program_pattern	Optional regex pattern to match program names (e.g., "Undergraduate", "MBA").

**Value**

A filtered data frame of matching courses.

**Examples**

```

schedule <- data.frame(
  INSTRUCTOR = c("Lee", "Smith", "Jones", "Dawson", "Garcia"),
  SUBJ = c("MATH", "NURS", "CSCI", "ENGL", "COMM"),
  COLLEGE = c("Science", "Nursing", "Science", "Arts and Sciences", "Arts and Communication"),
  DEPARTMENT = c("Math", "Nursing", "CS", "English", "Comm Studies"),
  PROGRAM = c("BS", "BSN", "BS", "BA", "BA"),
  stringsAsFactors = FALSE
)

filter_schedule(schedule, subject_pattern = "^MATH|^STAT")
filter_schedule(schedule, instructor_pattern = "smith")
filter_schedule(schedule, college_pattern = "Science")
filter_schedule(schedule, department_pattern = "Comm")
filter_schedule(schedule, program_pattern = "^BA$")

```

---

get\_instructor\_schedule

*Filter Course Schedule by Instructor (Regex-Friendly, Case-Insensitive)*

---

**Description**

Returns a subset of the course schedule containing courses taught by the specified instructor. Matching is case-insensitive and supports regular expressions, allowing flexible partial or pattern-based matching. If no match is found, a warning is issued and an empty data frame is returned.

**Usage**

```
get_instructor_schedule(instructor_name, schedule_df = schedule)
```

**Arguments**

instructor_name	A character string (or regular expression) used to match against values in the INSTRUCTOR column.
schedule_df	A data frame containing course schedule data with an INSTRUCTOR column. Defaults to schedule if not specified.

**Value**

A data frame of courses assigned to instructors matching the given pattern, sorted by descending enrollment.

**Examples**

```
get_instructor_schedule("smith", schedule_df = schedule) # partial match
get_instructor_schedule("^Smith,", schedule_df = schedule) # regex: starts with Smith
get_instructor_schedule("Robinson|Smith", schedule_df = schedule) # regex: matches either
```

---

get\_subject\_schedule *Filter Course Schedule by Subject Code (Regex-Friendly)*

---

**Description**

Returns a subset of the schedule where the SUBJ (subject code) column matches a given pattern. Matching is case-insensitive and supports regular expressions.

**Usage**

```
get_subject_schedule(subject_pattern, schedule)
```

**Arguments**

subject\_pattern      A character string or regular expression to match subject codes.

schedule              A data frame containing course schedule data with a SUBJ column.

**Value**

A filtered data frame containing only matching subject codes.

**Examples**

```
schedule <- data.frame(SUBJ = c("CSCI", "MATH", "STAT"))
get_subject_schedule("CSCI", schedule)
get_subject_schedule("^MATH|^STAT", schedule)
```

---

get\_unique\_instructors  
*Get Unique Instructor Names*

---

**Description**

Extracts a sorted vector of unique, non-empty instructor names from a schedule data frame.

**Usage**

```
get_unique_instructors(schedule_df)
```

**Arguments**

schedule\_df      A data frame containing an INSTRUCTOR column.

**Value**

A character vector of instructor names, sorted alphabetically.

**Examples**

```
get_unique_instructors(schedule)
```

---

ol_comp	<i>Calculate Overload Compensation for One Instructor</i>
---------	---

---

**Description**

Computes prorated overload pay and qualified credit hours for a single instructor based on course credit hours, enrollment, and institutional overload rules.

**Usage**

```
ol_comp(
  instructor_schedule,
  L = 4,
  U = 9,
  rate_per_cr = 2500/3,
  reg_load = 12,
  favor_institution = TRUE
)
```

**Arguments**

instructor_schedule	A data frame of the instructor's courses, with columns INSTRUCTOR, ENRLD, and HRS.
L	Lower enrollment threshold for overload pay qualification (default = 4).
U	Upper limit of proration; courses with ENRLD > U get full-rate pay (default = 9).
rate_per_cr	Base overload pay per credit hour (default = 2500/3).
reg_load	Regular teaching load in credit hours (default = 12).
favor_institution	Logical: if TRUE (default), prioritizes high-enrollment courses for regular load.

## Details

If `favor_institution = TRUE` (default), the function assigns **high-enrollment** qualified courses to the **regular load first**, resulting in lower compensation because only **low-enrollment** courses are left for overload pay — this favors the institution.

If `favor_institution = FALSE`, the function assigns **low-enrollment** qualified courses to the regular load first, preserving **high-enrollment** courses for compensation — this favors the instructor.

**Note:** This function assumes that `instructor_schedule` is already filtered for one instructor. Use `get_instructor_schedule()` to extract an instructor's schedule using flexible, case-insensitive pattern matching (regex supported, e.g., "smith|jones").

## Value

A tibble with course-level compensation and a human-readable summary block.

---

<code>ol_comp_byindex</code>	<i>Calculate Overload Compensation for One Instructor (by Index)</i>
------------------------------	--

---

## Description

Retrieves an instructor's name by index from the schedule and calculates their overload compensation using `ol_comp()`. Returns a clean, readable course-level compensation summary.

## Usage

```
ol_comp_byindex(
  i,
  schedule_df,
  L = 4,
  U = 9,
  rate_per_cr = 2500/3,
  reg_load = 12,
  favor_institution = TRUE
)
```

## Arguments

<code>i</code>	Integer index of the instructor (as returned by <code>get_unique_instructors()</code> ).
<code>schedule_df</code>	A data frame of the full course schedule containing an <code>INSTRUCTOR</code> column.
<code>L</code>	Lower enrollment threshold for overload pay eligibility (inclusive). Default is 4.
<code>U</code>	Upper enrollment limit for proration (inclusive). Default is 9.
<code>rate_per_cr</code>	Overload pay rate per qualified credit hour. Default is 2500/3.
<code>reg_load</code>	Regular teaching load in credit hours. Default is 12.
<code>favor_institution</code>	Logical: if <code>TRUE</code> (default), favors the institution by prioritizing high-enrollment courses for regular load.

**Details**

If `favor_institution = TRUE` (default), the function assigns high-enrollment courses to the regular load first, minimizing compensation.

If `favor_institution = FALSE`, low-enrollment courses are used toward the regular load first, preserving high-enrollment courses for overload pay.

This function internally uses `get_instructor_schedule()`, which supports flexible, case-insensitive regex matching for instructor names (e.g., `^smith$` or `johnson|williams`).

**Value**

Invisibly returns a tibble with the instructor's course-level overload compensation summary.

**Examples**

```
# Example usage with a schedule dataframe:
# ol_comp_byindex(1, schedule_df = schedule)
```

---

<code>ol_comp_summary</code>	<i>Summarize Overload Compensation</i>
------------------------------	--

---

**Description**

Calculates overload compensation for each instructor based on institutional policy. The output includes course-level payments, qualified credit hours, and a readable instructor-level summary block that follows each instructor's courses.

**Usage**

```
ol_comp_summary(
  schedule_df,
  instructor = NULL,
  L = 4,
  U = 9,
  rate_per_cr = 2500/3,
  reg_load = 12,
  favor_institution = TRUE
)
```

**Arguments**

<code>schedule_df</code>	A data frame containing course schedule information. Must include columns such as <code>INSTRUCTOR</code> , <code>HRS</code> , and <code>ENRLD</code> .
<code>instructor</code>	Optional string. If provided, limits the summary to a single instructor. Default is <code>NULL</code> (includes all instructors).
<code>L</code>	Minimum enrollment required for overload pay eligibility. Default is 4.

U	Upper threshold for proration. Courses with ENRLD > U receive full-rate pay. Default is 9.
rate_per_cr	Overload pay rate per credit hour. Default is 2500/3.
reg_load	Regular teaching load in credit hours. Default is 12.
favor_institution	Logical: if TRUE (default), prioritizes high-enrollment courses for regular load.

### Details

If the `instructor` argument is specified, the function limits the summary to that instructor. Matching is exact and case-sensitive unless pre-filtered using `get_instructor_schedule()`, which supports regex-based, case-insensitive pattern matching (e.g., "smith|jones").

### Value

A tibble combining course-level compensation and a summary section for each instructor.

---

schedule	<i>Sample Schedule Dataset</i>
----------	--------------------------------

---

### Description

This dataset represents a sample class schedule used by several institutions, with slight variations in variable names. Users must ensure that the key variables—`HRS`, `ENRLD`, and `INSTRUCTOR`—are properly named and formatted for the functions in this package to work correctly.

### Usage

```
schedule
```

### Format

An object of class `data.frame` with 665 rows and 15 columns.

### Details

A dataset containing instructor schedules for overload analysis.

The dataset has 665 rows and 15 columns, including:

**HRS** Credit hours of the course.

**ENRLD** Number of students enrolled in the course.

**INSTRUCTOR** Name of the instructor teaching the course (anonymized).

### Source

Included in the package under `data/schedule.rda`

# Index

## \* datasets

schedule, 8

filter\_schedule, 2

get\_instructor\_schedule, 3

get\_subject\_schedule, 4

get\_unique\_instructors, 4

ol\_comp, 5

ol\_comp\_byindex, 6

ol\_comp\_summary, 7

schedule, 8