

Package ‘pcatR’

September 3, 2026

Type Package

Title Analyze and Visualize Pragmatic Context Assessment Tool Data

Version 1.0.1

Description Provides a reproducible workflow for importing, validating, classifying, summarizing, visualizing, and reporting responses to the 14-item Pragmatic Context Assessment Tool (pCAT). The package preserves the instrument's two-part response structure, supports original and updated Consolidated Framework for Implementation Research (CFIR) mappings, describes team agreement and disagreement, compares repeated assessments, and creates implementation action-planning outputs. It does not calculate or claim a validated total pCAT scale score. The instrument is described by Robinson and Damschroder (2023) <[doi:10.1186/s43058-022-00380-5](https://doi.org/10.1186/s43058-022-00380-5)>; updated CFIR mappings are from Domlyn et al. (2026) <[doi:10.1186/s43058-026-00956-5](https://doi.org/10.1186/s43058-026-00956-5)>.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports ggplot2 (>= 3.4.0), grDevices, grid, stats, utils

Suggests covr, knitr, pkgdown, rmarkdown, shiny, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/Needs/check rcmdcheck

Config/Needs/coverage covr

Config/Needs/website pkgdown

RoxygenNote 7.3.2

NeedsCompilation no

URL <https://github.com/JaeManP/pcatR>, <https://jaemanp.github.io/pcatR/>

BugReports <https://github.com/JaeManP/pcatR/issues>

Author Lilac Li [aut, cph] (URL:
https://www.unk.edu/academics/management/lilac-li.php),
Jae Man Park [cre, cph] (URL:
https://www.linkedin.com/in/jae-man-park/)

Maintainer Jae Man Park <jaemanblp2@gmail.com>

Repository CRAN

Date/Publication 2026-09-03 11:40:02 UTC

Contents

pcatR-package	2
pcat_analyse	3
pcat_app	5
pcat_classify	5
pcat_items	6
pcat_palette	7
pcat_self_test	10
pcat_standardize	10
pcat_summarise	12
pcat_user_guide	15
pcat_validate	15
pcat_write_analysis	17
Index	20

pcatR-package	<i>pcatR: Analyze Pragmatic Context Assessment Tool Data</i>
---------------	--

Description

Tools for importing, validating, classifying, summarizing, comparing, and visualizing responses to the 14-item Pragmatic Context Assessment Tool (pCAT). The package preserves the separate direction and effect responses and does not calculate or claim a validated total scale score.

Details

The package supports the original CFIR mapping, the updated 2022 CFIR mapping, team agreement diagnostics, repeated assessments, barrier-focused action-plan tables, and an optional Shiny analysis dashboard.

See Also

[pcat_items](#), [pcat_validate](#), [pcat_classify](#), and [pcat_summarise](#).

pcat_analyse	<i>Run the standard pCAT analysis workflow</i>
--------------	--

Description

Validates and classifies responses, creates item-level summaries and consensus diagnostics, and optionally creates a barrier-focused action plan.

Usage

```
pcat_analyse(
  data,
  group_vars = NULL,
  respondent_id = "respondent_id",
  key_cols = NULL,
  require_complete = FALSE,
  neutral_effect = c("flag", "allow", "set_missing"),
  validation_action = c("none", "warn", "error"),
  strict = FALSE,
  suppress_below = NULL,
  agreement_threshold = 0.6,
  polarization_min = 0.2,
  minimum_n = 2L,
  include_action_plan = TRUE,
  barrier_threshold = 0.5,
  strong_barrier_threshold = 0.2,
  include_approximate = FALSE
)

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

Arguments

<code>data</code>	Long-format pCAT response data.
<code>group_vars</code>	Optional character vector of grouping columns, such as <code>site_id</code> and <code>timepoint</code> .
<code>respondent_id</code>	Name of the respondent identifier column.
<code>key_cols</code>	Columns defining one assessment for validation. When omitted, available project, site, team, timepoint, assessment date, and respondent identifiers are used.
<code>require_complete</code>	Logical; require exactly one response to every pCAT item 1–14 within each assessment.
<code>neutral_effect</code>	How to handle a neutral direction paired with any recorded effect: flag it, allow it, or set the effect to missing.
<code>validation_action</code>	Whether validation findings generate no condition, a warning, or an error.

<code>strict</code>	Logical; treat warnings as invalid when setting the validation object's valid field.
<code>suppress_below</code>	Optional positive minimum respondent count. Numeric summary results and the derived modal classification below this threshold are replaced with missing values.
<code>agreement_threshold</code>	One finite number from zero through one giving the minimum dominant-side share used for a consensus label.
<code>polarization_min</code>	One finite number from zero through one giving the minimum barrier and facilitator shares required to label an item polarized.
<code>minimum_n</code>	Minimum number of valid direction responses required for a consensus label.
<code>include_action_plan</code>	Logical; create a barrier-focused action-plan table.
<code>barrier_threshold</code>	One finite number from zero through one giving the minimum barrier prevalence for action-plan inclusion.
<code>strong_barrier_threshold</code>	One finite number from zero through one giving the minimum strong-barrier prevalence for action-plan inclusion.
<code>include_approximate</code>	Logical; include the source table's non-identical Relative Priority to Relative Advantage approximation.
<code>x</code>	A <code>pcat_analysis</code> object.
<code>...</code>	Additional arguments, currently ignored by the print method.

Details

The result retains validation findings and all denominators needed for transparent reporting. It does not calculate an overall pCAT scale score. Small-cell suppression hides numeric analytic measures and the derived modal classification in summary tables; respondent-level classified data are retained in the result and must be governed separately.

Value

`pcat_analyse()` returns a list of class `pcat_analysis` with components `validation`, `classified`, `summary`, `consensus`, `action_plan`, and `settings`. The print method returns `x` invisibly.

Examples

```
analysis <- pcat_analyse(
  pcat_example_data(),
  group_vars = c("site_id", "timepoint"),
  require_complete = TRUE,
  validation_action = "none"
)
analysis
head(analysis$summary)
```

`pcat_app`*Launch the optional pCAT analysis dashboard*

Description

Runs a bundled Shiny interface for uploading a non-sensitive long-format CSV, validating and classifying responses, viewing issues and a profile plot, and exporting classified data.

Usage

```
pcat_app()
```

Details

The application is an analysis interface, not a secure survey-collection platform. Do not upload protected health information or sensitive direct identifiers.

Value

The value returned by `shiny::runApp()`.

Examples

```
if (interactive()) {  
  if (requireNamespace("shiny", quietly = TRUE)) {  
    pcat_app()  
  }  
}
```

`pcat_classify`*Classify pCAT responses*

Description

Combines direction and effect responses into strong barrier, weak barrier, neutral, weak facilitator, and strong facilitator categories. It retains explicit states for invalid responses and missing effect responses.

Usage

```
pcat_classify(  
  data,  
  validation_action = c("none", "warn", "error"),  
  attach_items = TRUE  
)
```

Arguments

`data` Long-format response data or a `pcat_validation` object.

`validation_action` Condition behavior used when raw data must first be validated.

`attach_items` Logical; attach item wording and CFIR metadata.

Details

Strength categories are assigned only when both direction and effect validation are valid. A neutral row with an invalid effect remains directionally neutral but has an invalid strength and no complete five-category classification.

The numeric `pcat_display_code` ranges from -2 to +2 and is supplied only for descriptive plotting and transition displays. It is not a validated scale score and should not be summed into an overall pCAT score.

Value

A data frame of class `pcat_classified`.

Examples

```
classified <- pcat_classify(pcat_example_data())
head(classified)
```

`pcat_items`

Instrument metadata, CFIR mappings, and included example data

Description

Returns the 14-item pCAT dictionary, a long-form mapping table, response-code definitions, synthetic example data, or source-derived candidate ERIC strategy links.

Usage

```
pcat_items(cfir_version = c("both", "original", "2022"))

pcat_construct_map(
  cfir_version = c("both", "original", "2022"),
  include_secondary = TRUE
)

pcat_response_options()

pcat_example_data()

pcat_strategy_candidates(
```

```
construct = NULL,  
tier = c("all", "1", "2"),  
include_approximate = FALSE  
)
```

Arguments

cfir_version	Mapping version to return: both source versions, the original CFIR, or the updated 2022 CFIR.
include_secondary	Logical; retain secondary item-to-construct links. The 2026 supplementary mapping gives item 10 a secondary Funding link.
construct	Optional original-CFIR or source-table construct name used to filter candidate strategies.
tier	Candidate strategy endorsement tier: all, tier 1, or tier 2.
include_approximate	Logical; include the source table’s non-identical Relative Priority to Relative Advantage approximation.

Details

pcat_items() keeps one primary updated-CFIR construct per item for item-level reporting. pcat_construct_map() exposes every retained item-construct link. Candidate strategies are prompts for local deliberation, not automatically selected or validated prescriptions.

Value

A data frame. pcat_items() has one row per pCAT item; pcat_construct_map() has one row per item-construct link.

Examples

```
pcat_items("both")  
pcat_construct_map("2022")  
pcat_response_options()  
head(pcat_example_data())  
pcat_strategy_candidates("Available Resources", tier = "1")
```

pcat_palette	Default pCAT plotting palette
--------------	-------------------------------

Description

Creates a publication-quality diverging bar chart. Barriers are plotted to the left, facilitators to the right, and the neutral share is printed at the center. Percentages use complete five-category responses as the denominator.

Usage

```
pcat_palette()

plot_pcat_profile(
  data,
  group_vars = NULL,
  label = c("cfir_original_construct", "cfir_2022_construct", "item_text", "item_id"),
  label_width = 42L,
  facet_ncol = NULL,
  facet_label = c("values", "names_values"),
  show_neutral = TRUE,
  show_n = TRUE,
  base_size = 11,
  label_size = 8.5,
  legend_position = "bottom",
  title = "pCAT barrier-facilitator profile",
  subtitle = NULL,
  caption = NULL,
  palette = pcat_palette()
)

plot_pcat_heatmap(
  data,
  respondent_id = "respondent_id",
  facet_vars = NULL,
  facet_ncol = NULL,
  facet_label = c("values", "names_values"),
  base_size = 11,
  title = "pCAT respondent-by-item profile",
  subtitle = NULL,
  caption = paste("Missing, invalid, effect-incomplete, or absent item records are",
    "shown in light grey."),
  palette = pcat_palette()
)

plot_pcat_change(
  data,
  from = NULL,
  to = NULL,
  respondent_id = "respondent_id",
  facet_vars = NULL,
  display = c("transition", "delta"),
  facet_ncol = NULL,
  facet_label = c("values", "names_values"),
  base_size = 11,
  title = "Paired pCAT change",
  subtitle = NULL,
  caption = NULL,
```



```
    ...  
  )
```

Arguments

<code>data</code>	Output from <code>pcat_change()</code> or classified data.
<code>group_vars</code>	Optional grouping columns used for facets.
<code>label</code>	Item label field.
<code>label_width</code>	Approximate character width used to wrap item labels.
<code>facet_ncol</code>	Number of facet columns.
<code>facet_label</code>	Whether facet strips show values only or names and values.
<code>show_neutral</code>	Print the neutral percentage at the center.
<code>show_n</code>	Include the complete-response denominator in the neutral label.
<code>base_size</code>	Base text size.
<code>label_size</code>	Text size for item labels.
<code>legend_position</code>	Position of the shared legend.
<code>title, subtitle, caption</code>	Optional plot annotations.
<code>palette</code>	Named color vector. See <code>pcat_palette()</code> .
<code>respondent_id</code>	Respondent identifier column in paired output.
<code>facet_vars</code>	Optional grouping columns used for facets.
<code>from, to</code>	Required only when data have not already been paired.
<code>display</code>	Display transition categories or the descriptive delta code.
<code>...</code>	Passed to <code>pcat_change()</code> when pairing is needed.

Value

A named character vector of color values.

A ggplot object.

A ggplot object.

A ggplot object.

pcat_self_test	<i>Run the built-in pcatR self-test</i>
----------------	---

Description

Runs checks of the instrument dictionary, validation, classification, summaries, reshaping, longitudinal comparison, action planning, the standard analysis wrapper, and plotting.

Usage

```
pcat_self_test(verbose = TRUE)
```

Arguments

verbose	Print one line per check.
---------	---------------------------

Value

A data frame with check names and results, invisibly.

pcat_standardize	<i>Prepare, import, and reshape pCAT data</i>
------------------	---

Description

Reads standard pCAT CSV files, standardizes user column names, creates entry templates, writes CSV templates, and converts responses between long and wide layouts.

Usage

```
pcat_read_csv(
  path,
  layout = c("auto", "long", "wide"),
  item_prefix = "item",
  na = c("", "NA")
)
```

```
pcat_standardize(
  data,
  respondent_id = "respondent_id",
  item_id = "item_id",
  direction = "direction",
  effect = "effect",
  project_id = NULL,
  site_id = NULL,
  team_id = NULL,
```

```

    role = NULL,
    timepoint = NULL,
    assessment_date = NULL,
    comment = NULL,
    keep_original = TRUE
  )

pcat_template(
  format = c("long", "wide"),
  n_respondents = 1L,
  include_item_text = TRUE
)

pcat_write_template(
  path,
  format = c("long", "wide"),
  n_respondents = 1L,
  include_item_text = TRUE,
  overwrite = FALSE
)

pcat_wide_to_long(data, item_prefix = "item")

pcat_long_to_wide(data, id_cols = NULL, item_prefix = "item")

```

Arguments

path	Input or output CSV path, depending on the function.
layout	Input layout: auto-detected, long, or wide.
item_prefix	Prefix used before item numbers in wide response columns.
na	Character values interpreted as missing when reading a CSV.
data	A data frame.
respondent_id, item_id, direction, effect	Source column names, supplied as character strings, for the four required fields.
project_id, site_id, team_id, role, timepoint, assessment_date, comment	Optional source column names supplied as character strings.
keep_original	Logical; retain source columns whose names differ from the standard names.
format	Output layout, either long or wide.
n_respondents	Positive number of blank respondent records.
include_item_text	Logical; include item wording in a long entry template.
overwrite	Logical; replace an existing output file.
id_cols	Columns defining one assessment row in wide output.

Details

Standard long data require respondent_id, item_id, direction, and effect. Standard wide response names follow item01_direction, item01_effect, through item 14. Free-text comments should be reviewed for sensitive information before analysis or sharing.

Value

A data frame, except pcat_write_template(), which invisibly returns the normalized output path.

Examples

```
long <- pcat_template("long", n_respondents = 2)
wide <- pcat_long_to_wide(long, id_cols = c("respondent_id", "timepoint"))
pcat_wide_to_long(wide)

template_path <- tempfile("pcat_template_", fileext = ".csv")

pcat_write_template(
  template_path,
  format = "long",
  n_respondents = 2
)

imported <- pcat_read_csv(template_path)
head(imported)
```

pcat_summarise

Summarize, compare, and plan from pCAT data

Description

Creates item-level summaries, team agreement and disagreement diagnostics, paired longitudinal transitions, and barrier-focused action-plan tables.

Usage

```
pcat_summarise(
  data,
  group_vars = NULL,
  respondent_id = "respondent_id",
  suppress_below = NULL
)

pcat_consensus(
  data,
  group_vars = NULL,
```

```

    agreement_threshold = 0.6,
    polarization_min = 0.2,
    minimum_n = 2L
  )

pcat_change(
  data,
  timepoint = "timepoint",
  from,
  to,
  id_cols = NULL
)

pcat_action_plan(
  data,
  group_vars = NULL,
  barrier_threshold = 0.5,
  strong_barrier_threshold = 0.2,
  include_strategy_candidates = TRUE,
  include_approximate = FALSE
)

```

Arguments

<code>data</code>	Raw, validated, classified, or previously summarized pCAT data, as appropriate for the function.
<code>group_vars</code>	Optional character vector of grouping columns such as site and timepoint.
<code>respondent_id</code>	Respondent identifier column name.
<code>suppress_below</code>	Optional positive minimum respondent count. Numeric analytic measures and the derived <code>modal_class</code> below it are replaced with missing values and suppressed is set to TRUE.
<code>agreement_threshold</code>	One finite number from zero through one giving the minimum dominant-side share used for a consensus label.
<code>polarization_min</code>	One finite number from zero through one giving the minimum barrier and facilitator shares required to label an item polarized.
<code>minimum_n</code>	Minimum number of valid direction responses.
<code>timepoint</code>	Timepoint column name.
<code>from, to</code>	Earlier and later timepoint values.
<code>id_cols</code>	Columns identifying paired respondent-item records.
<code>barrier_threshold</code>	One finite number from zero through one giving the minimum barrier prevalence for action-plan inclusion.
<code>strong_barrier_threshold</code>	One finite number from zero through one giving the minimum strong-barrier prevalence for action-plan inclusion.

```
include_strategy_candidates
    Logical; join source-derived candidate ERIC strategies where a direct construct
    link is available.
include_approximate
    Logical; include the non-identical Relative Priority to Relative Advantage ap-
    proximation.
```

Details

Directional percentages (`pct_barrier`, `pct_neutral`, `pct_facilitator`, `pct_effect_missing`, and `pct_complete_class`) use eligible records with a valid direction and denominator `n_valid_direction`. Complete five-category percentages use only eligible records with a complete direction-plus-effect classification and denominator `n_complete_class`. The directional `n_neutral` count can therefore exceed `n_neutral_complete` when a neutral direction has an invalid effect. All five complete-category counts are derived from `pcat_class5` and partition `n_complete_class`. Report both denominators when they differ. `pct_complete_class` is the proportion of valid-direction records with a complete five-category classification.

Means, medians, and change values based on `pcat_display_code` are strictly descriptive. Candidate strategies require local stakeholder review and are not automatic prescriptions. Small-cell suppression hides numeric analytic measures and `modal_class`; it applies only to the returned summary table and does not de-identify source or classified data. When barrier or facilitator percentages are unavailable, `pcat_consensus()` reports `polarized = NA`; unavailable data are not treated as evidence that a cell is nonpolarized.

Value

A data frame with a function-specific class and documented denominator attributes where applicable.

Examples

```
classified <- pcat_classify(pcat_example_data())
summary <- pcat_summarise(
  classified,
  group_vars = c("site_id", "timepoint")
)
pcat_consensus(summary)

pcat_change(
  classified,
  from = "planning",
  to = "mid_implementation"
)

pcat_action_plan(
  summary,
  group_vars = c("site_id", "timepoint")
)
```

pcat_user_guide	<i>Locate or open the pcatR technical user guide</i>
-----------------	--

Description

Locates the technical user guide installed with pcatR. The guide defines the package's data schema, validation rules, denominators, descriptive classification logic, privacy cautions, and reporting expectations.

Usage

```
pcat_user_guide(format = c("pdf", "html"), open = interactive())
```

Arguments

format	Guide format: PDF or HTML.
open	Logical; open the guide with the operating system's default application.

Value

The normalized guide path, invisibly.

Examples

```
path <- pcat_user_guide("html", open = FALSE)
path

if (interactive()) {
  pcat_user_guide("html", open = TRUE)
}
```

pcat_validate	<i>Validate pCAT response data</i>
---------------	------------------------------------

Description

Checks identifiers, item numbers, direction and effect values, duplicate keys, neutral responses with recorded effects, missing effects for barrier or facilitator directions, and optional 14-item completeness.

Usage

```
pcat_validate(
  data,
  key_cols = NULL,
  require_complete = FALSE,
  neutral_effect = c("flag", "allow", "set_missing"),
  action = c("warn", "error", "none"),
  strict = FALSE
)

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

pcat_validation_data(x, valid_rows_only = FALSE)

pcat_validation_issues(x, level = c("all", "row", "assessment"))
```

Arguments

<code>data</code>	Long-format pCAT response data. Required columns are <code>respondent_id</code> , <code>item_id</code> , <code>direction</code> , and <code>effect</code> .
<code>key_cols</code>	Columns defining one assessment. When omitted, available project, site, team, timepoint, assessment date, and respondent identifiers are used.
<code>require_complete</code>	Logical; require exactly one row for every item 1–14 within each assessment.
<code>neutral_effect</code>	How to handle a neutral direction paired with any recorded effect: flag it, allow it, or set the effect to missing.
<code>action</code>	Whether validation findings generate a warning, an error, or no condition.
<code>strict</code>	Logical; treat warnings as invalid when setting the object's <code>valid</code> field.
<code>x</code>	A <code>pcat_validation</code> object.
<code>...</code>	Additional arguments, currently ignored by the print method.
<code>valid_rows_only</code>	Logical; retain only rows without row-level errors.
<code>level</code>	Return all issues, row-level issues, or assessment-completeness issues.

Details

The default behavior flags a neutral response paired with either effect code as a warning because effect strength is not applicable when direction is neutral. Validation findings remain available in structured form even when `action = "none"`.

Value

`pcat_validate()` returns a `pcat_validation` object. The extractor functions return data frames. The print method returns `x` invisibly.

Examples

```
dat <- pcat_example_data()
val <- pcat_validate(dat, require_complete = TRUE, action = "none")
val
head(pcat_validation_data(val))
pcat_validation_issues(val)
```

pcat_write_analysis	<i>Export pCAT analysis results and profile figures</i>
---------------------	---

Description

`pcat_write_analysis()` writes a reproducible directory containing an analysis manifest, validation findings, optional classified responses, item summaries, consensus diagnostics, an optional action-plan worksheet, analysis settings, session information, and an optional profile PDF. `pcat_save_profile_pdf()` places one grouping combination on each page. It validates every page before writing a temporary PDF and replaces the target only after the complete temporary file is nonempty. An existing directory is rejected as an output path before any backup or output operation.

Usage

```
pcat_write_analysis(
  x,
  path,
  overwrite = FALSE,
  include_profile_pdf = TRUE,
  include_classified = TRUE,
  profile_label = c(
    "cfir_original_construct", "cfir_2022_construct", "item_text", "item_id"
  )
)

pcat_save_profile_pdf(
  data,
  path,
  group_vars = NULL,
  overwrite = FALSE,
  width = 11,
  height = 8.5,
  title_prefix = "pCAT profile",
  ...
)
```

Arguments

x A `pcat_analysis` object returned by `pcat_analyse()`.

<code>path</code>	Output directory for <code>pcat_write_analysis()</code> or output PDF file path for <code>pcat_save_profile_pdf()</code> . Existing directories are rejected for the PDF exporter.
<code>overwrite</code>	Logical; replace files previously generated by <code>pcatR</code> in the output directory. Unrelated files are retained.
<code>include_profile_pdf</code>	Logical; include a multi-page profile PDF in the analysis export.
<code>include_classified</code>	Logical; write respondent-item classifications to <code>02_classified_responses.csv</code> . Set to <code>FALSE</code> when a response-level file is unnecessary or should not be distributed.
<code>profile_label</code>	Item-label field used in the exported profile PDF.
<code>data</code>	Raw, validated, or classified pCAT data.
<code>group_vars</code>	Grouping columns defining one PDF page.
<code>width, height</code>	Positive finite PDF page dimensions in inches.
<code>title_prefix</code>	Prefix used in profile page titles.
<code>...</code>	Additional arguments passed to <code>plot_pcat_profile()</code> .

Details

If summary tables were created with `suppress_below`, the classified response file and profile PDF do not automatically inherit that suppression. Review respondent-level and graphical outputs under the applicable privacy and disclosure rules before dissemination. When `overwrite = TRUE`, only known files generated by `pcat_write_analysis()` are removed before the new export is written. If profile-page construction or writing fails, `pcat_save_profile_pdf()` identifies the affected group, removes its temporary file, and leaves any existing target unchanged.

Value

The normalized output directory or file path, invisibly.

Examples

```
analysis <- pcat_analyse(
  pcat_example_data(),
  group_vars = c("site_id", "timepoint"),
  validation_action = "none"
)

analysis_dir <- tempfile("pcat_analysis_")

pcat_write_analysis(
  analysis,
  path = analysis_dir,
  include_profile_pdf = FALSE,
  include_classified = FALSE
)

list.files(analysis_dir)
```

```
profile_path <- tempfile("pcat_profiles_", fileext = ".pdf")

pcat_save_profile_pdf(
  analysis$classified,
  path = profile_path,
  group_vars = c("site_id", "timepoint"),
  label = "item_id"
)
```

Index

`pcat_action_plan` (`pcat_summarise`), 12
`pcat_analyse`, 3
`pcat_app`, 5
`pcat_change` (`pcat_summarise`), 12
`pcat_change()`, 9
`pcat_classify`, 2, 5
`pcat_consensus` (`pcat_summarise`), 12
`pcat_construct_map` (`pcat_items`), 6
`pcat_example_data` (`pcat_items`), 6
`pcat_items`, 2, 6
`pcat_long_to_wide` (`pcat_standardize`), 10
`pcat_palette`, 7
`pcat_palette()`, 9
`pcat_read_csv` (`pcat_standardize`), 10
`pcat_response_options` (`pcat_items`), 6
`pcat_save_profile_pdf`
 (`pcat_write_analysis`), 17
`pcat_self_test`, 10
`pcat_standardize`, 10
`pcat_strategy_candidates` (`pcat_items`), 6
`pcat_summarise`, 2, 12
`pcat_template` (`pcat_standardize`), 10
`pcat_user_guide`, 15
`pcat_validate`, 2, 15
`pcat_validation_data` (`pcat_validate`), 15
`pcat_validation_issues` (`pcat_validate`),
 15
`pcat_wide_to_long` (`pcat_standardize`), 10
`pcat_write_analysis`, 17
`pcat_write_template` (`pcat_standardize`),
 10
`pcatR` (`pcatR`-package), 2
`pcatR`-package, 2
`plot_pcat_change` (`pcat_palette`), 7
`plot_pcat_heatmap` (`pcat_palette`), 7
`plot_pcat_profile` (`pcat_palette`), 7
`print.pcat_analysis` (`pcat_analyse`), 3
`print.pcat_validation` (`pcat_validate`),
 15