

# Package: readmit (via r-universe)

May 28, 2026

**Title** Tools for Working with Hospital Readmissions Data

**Version** 0.0.1.9000

**Description** Contains tools for working with and analyzing hospital readmissions data. The package provides utilities for components of the Hospital Readmissions Reduction Program (HRRP), including program timeline functions, Hospital-Specific Report (HSR) helpers, and general importing tools for the Provider Data Catalog (PDC).

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Imports** dplyr, httr, readr, readxl, rlang, stringr, tibble, tidyr

**Suggests** ggplot2, knitr, purrr, reactable, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**URL** <https://centralstatz.github.io/readmit/>,  
<https://github.com/centralstatz/readmit>

**Depends** R (>= 4.1.0)

**LazyData** true

**Config/Needs/website** rmarkdown

**BugReports** <https://github.com/centralstatz/readmit/issues>

**Config/pak/sysreqs** libicu-dev libssl-dev libx11-dev

**Repository** <https://centralstatz.r-universe.dev>

**Date/Publication** 2026-03-29 19:58:18 UTC

**RemoteUrl** <https://github.com/centralstatz/readmit>

**RemoteRef** HEAD

**RemoteSha** 6405a434fa5871f9599027bbd8c8f6d005f13d8f

## Contents

hrrp_get_dates . . . . .	2
hrrp_keydates . . . . .	3
hsr_coefficients . . . . .	5
hsr_cohort_summary . . . . .	6
hsr_detect_format . . . . .	6
hsr_discharges . . . . .	7
hsr_dual_stays . . . . .	8
hsr_mock_reports . . . . .	9
hsr_parse . . . . .	10
hsr_parser_config . . . . .	10
hsr_payment_summary . . . . .	11
hsr_read . . . . .	12
hsr_readmission_risks . . . . .	13
hsr_resolve_source . . . . .	15
is_hsr . . . . .	15
is_hsr_bundle . . . . .	16
is_hsr_source . . . . .	16
pdv_read . . . . .	17
<b>Index</b>	<b>19</b>

---

hrrp_get_dates	<i>Find relevant dates from the Hospital Readmissions Reduction Program (HRRP)</i>
----------------	--

---

### Description

Identifies key dates (see [hrrp\\_keydates](#)) from the **Hospital Readmissions Reduction Program (HRRP)** that are associated with an input reference date, such as performance and payment periods.

### Usage

```
hrrp_get_dates(ref, period = c("payment", "performance"), discharge = TRUE)
```

### Arguments

ref	A Date object
period	The program period to extract dates for. One of c("payment", "performance").
discharge	Should the ref date be taken as a <i>discharge</i> date? Defaults to TRUE. If FALSE, it's taken to be a penalty/program date.

### Value

A [tibble::tibble](#)

**See Also**[hrrp\\_keydates](#)**Examples**

```
my_date <- as.Date("2022-01-01")

# What are the payment periods for this discharge?
hrrp_get_dates(my_date, period = "payment", discharge = TRUE)

# What performance periods is this discharge included in?
hrrp_get_dates(my_date, period = "performance", discharge = TRUE)

# What is the payment period overlapping this date?
hrrp_get_dates(my_date, period = "payment", discharge = FALSE)

# What is the performance period whose penalty period overlaps this date?
hrrp_get_dates(my_date, period = "performance", discharge = FALSE)

# What is the performance period for current penalty enforcement?
hrrp_get_dates(Sys.Date(), period = "performance", discharge = FALSE)
```

---

hrrp_keydates	<i>Key dates from the Hospital Readmissions Reduction Program (HRRP)</i>
---------------	--

---

**Description**

A collection of datasets giving important dates involved in the **Hospital Readmissions Reduction Program (HRRP)**, including performance periods, payment periods, review and correction periods, claims snapshot dates, and cohort inclusion status for program years since 2019. The data is manually abstracted from **QualityNet** and was most recently updated on 11/21/2025.

All tables contain the ProgramYear, which is the HRRP program (or federal fiscal year) in question. Tables with date ranges contain a StartDate and EndDate contain the beginning and end dates of the respective interval. Other tables contain individual dates or indicators for cohort inclusion (see descriptions).

**Usage**

hrrp\_keydates

hrrp\_performance\_periods

hrrp\_payment\_periods

hrrp\_review\_periods

hrrp\_snapshot\_dates

hrrp\_cohort\_inclusion

### Format

**hrrp\_performance\_periods:**

Start and end dates for discharges evaluated for readmissions. Note some program years have multiple performance intervals and are represented on separate rows of this table.

**hrrp\_payment\_periods:**

Start and end dates for when payment penalties are applied by CMS to hospital reimbursement.

**hrrp\_review\_periods:**

Start and end dates for the review and correction period where hospitals could review discharge-level data and downstream calculations of penalty amounts.

**hrrp\_snapshot\_dates:**

The *as of* date for when claims data were extracted by CMS for evaluation in the program. *Note some discrepancies exist in CMS documentation for these dates (e.g., FY2026) when comparing anticipated versus actual snapshot date. The latter was used in this case.*

**hrrp\_cohort\_inclusion:**

Indicators of whether each cohort was included in the program for the given year. 1 = included, 0 = excluded.

**AMI** Acute Myocardial Infarction

**COPD** Chronic Obstructive Pulmonary Disease

**HF** Heart Failure

**PN** Pneumonia

**CABG** Coronary Artery Bypass Graft

**HK** Total Hip/Knee Replacement

**hrrp\_keydates:**

A combined table that joins all of the above individual tables into a single common table with altered column names.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 11 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 9 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 8 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 9 rows and 2 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 9 rows and 7 columns.

### Source

<https://qualitynet.cms.gov/inpatient/hrrp/resources#tab1>

---

hsr_coefficients	<i>Extract risk model coefficients from a Hospital-Specific Report (HSR)</i>
------------------	--

---

### Description

Extracts the regression coefficients from the parsed HSR coefficients component for a given cohort.

### Usage

```
hsr_coefficients(file, cohort)
```

### Arguments

file	A parsed HSR bundle or a local source bundle path.
cohort	Cohort to extract the coefficients for. One of c("AMI", "COPD", "HF", "PN", "CABG", "HK")

### Value

A `tibble::tibble()` containing the columns:

- Factor: The model term name (as listed in the file)
- Value: The model coefficient value (on the linear predictor scale)

### Examples

```
bundle_path <- file.path(tempdir(), "hsr_bundle")
dir.create(bundle_path, recursive = TRUE)
utils::write.csv(
  data.frame(
    cohort = c("HF", "HF"),
    term = c("AGE", "HOSP_EFFECT"),
    value = c(0.25, -0.10)
  ),
  file.path(bundle_path, "coefficients.csv"),
  row.names = FALSE
)
hsr_coefficients(bundle_path, "HF")
```

---

hsr_cohort_summary	<i>Extract cohort summary information from a Hospital-Specific Report (HSR)</i>
--------------------	---

---

### Description

Extracts cohort summary information from the HSR, including (but not limited to) the discharge/readmission volumes, predicted/expected readmission rates, peer group medians, and DRG ratios.

**Note:** CMS changed the format of Hospital-Specific Reports (HSRs) for FY2026 (see [here](#)). The current HSR functions support formats through FY2025.

### Usage

```
hsr_cohort_summary(file)
```

### Arguments

file                    A parsed HSR bundle, a local source bundle path, or a legacy file path.

### Value

A `tibble::tibble()` containing the full Table 2 parsed from the report.

### Examples

```
# Access a report
my_report <- hsr_mock_reports("FY2025_HRRP_MockHSR.xlsx")

# Extract the cohort summary as a dataset
hsr_cohort_summary(my_report)
```

---

hsr_detect_format	<i>Detect the format of a Hospital-Specific Report (HSR)</i>
-------------------	--

---

### Description

Detects the broad file format of a Hospital-Specific Report (HSR). Phase 1 only recognizes the legacy Excel-based format used through FY2025.

### Usage

```
hsr_detect_format(path)
```

### Arguments

path                    File path to a report.

**Value**

A scalar character string describing the detected format.

**Examples**

```
report <- hsr_mock_reports("FY2025_HRRP_MockHSR.xlsx")
hsr_detect_format(report)
```

---

hsr_discharges	<i>Extract discharge-level data from a Hospital-Specific Report (HSR)</i>
----------------	---

---

**Description**

Extracts discharge-level data for a specific cohort from a parsed HSR bundle.

**Usage**

```
hsr_discharges(  
  file,  
  cohort,  
  discharge_phi = TRUE,  
  risk_factors = FALSE,  
  eligible_only = FALSE  
)
```

**Arguments**

file	A parsed HSR bundle or a local source bundle path.
cohort	Cohort to extract the discharges for. One of c("AMI", "COPD", "HF", "PN", "CABG", "HK")
discharge_phi	Should discharge PHI be included? Defaults to TRUE (see details).
risk_factors	Should readmission risk factors be included? Defaults to FALSE (see details).
eligible_only	Should only eligible discharges be included? Defaults to FALSE (see details).

**Details**

Parsed discharge components must already have any file-structure differences resolved upstream by the parser layer. This helper operates only on the conceptual discharges component in the parsed bundle.

eligible\_only = TRUE is supported when the parsed component contains one or more cohort inclusion columns. The discharge\_phi and risk\_factors arguments currently only support the default behavior (discharge\_phi = TRUE, risk\_factors = FALSE); other combinations fail clearly until richer discharge field metadata is available in parsed bundles.

**Value**

A `tibble::tibble()`

## Examples

```
bundle_path <- file.path(tempdir(), "hsr_bundle")
dir.create(bundle_path, recursive = TRUE)
utils::write.csv(
  data.frame(
    cohort = c("HF", "HF"),
    `ID Number` = c(1001, 1002),
    index_stay = c(1, 1)
  ),
  file.path(bundle_path, "discharges.csv"),
  row.names = FALSE
)
hsr_discharges(bundle_path, "HF")
```

---

hsr_dual_stays	<i>Extract dually-eligible discharges from a Hospital-Specific Report (HSR)</i>
----------------	---

---

## Description

Extracts the discharge-level records from the HSR of patients who were dually-eligible for Medicare and Medicaid benefits (see details).

**Note:** CMS changed the format of Hospital-Specific Reports (HSRs) for FY2026 (see [here](#)). The current HSR functions support formats through FY2025.

## Usage

```
hsr_dual_stays(file)
```

## Arguments

`file` A parsed HSR bundle, a local source bundle path, or a legacy file path.

## Details

In the **Hospital Readmissions Reduction Program (HRRP)**, hospitals' readmission rates are compared against a peer group of "like" hospitals, which determines whether or not they will get penalized financially.

The peer group allocation done by **CMS** is determined by creating hospital groupings based on the share of Medicare beneficiaries who were also eligible for Medicaid benefits, a marker of socioeconomic status in the hospital population. `hsr_dual_stays()` extracts the list of discharges accounting for the numerator of this ratio.

## Value

A `tibble::tibble()`

## Examples

```
# Access a report
my_report <- hsr_mock_reports("FY2025_HRRP_MockHSR.xlsx")

# Extract dually-eligible stays as a dataset
hsr_dual_stays(my_report)
```

---

hsr_mock_reports	Retrieve file location of mock Hospital-Specific Reports (HSR)
------------------	--

---

## Description

Provides the location of mock HSRs downloaded from QualityNet that come with the package that users can import. These files are a representation of what a hospital's real report looks like when received from CMS. They contain mock data for sensitive fields (discharge level data, etc.), but real data for national level results (e.g., model coefficients). Thus, it gives user ability to practice/explore package functions and translate them to their own hospital reports. Files include fiscal years (FY) 2019-2025 and were downloaded on 11/8/2025 from <https://qualitynet.cms.gov/inpatient/hrrp/reports>. File names were changed for better identifiability.

**Note:** CMS changed the format of Hospital-Specific Reports (HSRs) for FY2026 (see [here](#)). The current HSR functions support formats through FY2025.

## Usage

```
hsr_mock_reports(path = NULL)
```

## Arguments

path                    Name of file. If NULL, all files will be listed.

## Details

This function was adapted from [readxl::readxl\\_example\(\)](#).

## Value

A character string or vector of strings

## Examples

```
# Show all available mock reports
hsr_mock_reports()

# Show path to a single report
hsr_mock_reports("FY2025_HRRP_MockHSR.xlsx")

# Use mock report for testing package functions
hsr_payment_summary(hsr_mock_reports("FY2025_HRRP_MockHSR.xlsx"))
```

---

hsr\_parse                      *Parse an HSR source bundle*

---

### Description

Parses a local HSR source bundle into a lightweight conceptual bundle. The parsed bundle tracks component outputs separately from source artifacts and preserves artifact lineage and raw parser metadata.

### Usage

```
hsr_parse(x, spec = hsr_parser_config(), ...)
```

### Arguments

x                      A local HSR source bundle path or a readmit\_hsr\_source object.  
spec                    A parser configuration created by hsr\_parser\_config().  
...                     Reserved for future parser options.

### Value

A readmit\_hsr\_bundle object.

### Examples

```
bundle_path <- file.path(tempdir(), "hsr_bundle")
dir.create(bundle_path, recursive = TRUE)
utils::write.csv(
  data.frame(metric = "payment_adjustment_factor", value = 0.98),
  file.path(bundle_path, "payment_summary.csv"),
  row.names = FALSE
)
hsr_parse(bundle_path, spec = hsr_parser_config())
```

---

hsr\_parser\_config            *Create an HSR parser configuration*

---

### Description

Creates a parser configuration object that controls how local HSR source bundles are interpreted. The package provides built-in defaults for file inclusion, artifact role hints, embedded component markers, and basic column aliases, while allowing users to override selected assumptions when their files differ.

### Usage

```
hsr_parser_config(artifacts = NULL, components = NULL)
```

**Arguments**

artifacts      Optional artifact-level overrides.  
components    Optional component-level overrides.

**Value**

A readmit\_hsr\_spec object.

**Examples**

```
spec <- hsr_parser_config()
spec$artifacts$role_hints$payment_summary <- c("payment", "summary_table")
spec
```

---

hsr\_payment\_summary    *Extract payment summary information from a Hospital-Specific Report (HSR)*

---

**Description**

Extracts payment summary information from the HSR, including (but not limited to) the payment penalty, peer group the hospital was compared against, and dual proportion that determines peer group assignment.

**Note:** CMS changed the format of Hospital-Specific Reports (HSRs) for FY2026 (see [here](#)). The current HSR functions support formats through FY2025.

**Usage**

```
hsr_payment_summary(file)
hsr_count_dual_stays(file)
hsr_count_total_stays(file)
hsr_dual_proportion(file)
hsr_peer_group(file)
hsr_neutrality_modifier(file)
hsr_payment_adjustment_factor(file)
hsr_payment_penalty(file)
```

**Arguments**

`file` A parsed HSR bundle, a local source bundle path, or a legacy file path. For convenience functions, this can also be the pre-parsed payment summary table itself (to minimize file I/O).

**Value**

- `hsr_payment_summary()` returns a `tibble::tibble()` containing the full Table 1 parsed from the report.
- Additional convenience functions extract specific columns from this table, and always return a numeric value.

**Examples**

```
# Access a report
my_report <- hsr_mock_reports("FY2025_HRRP_MockHSR.xlsx")

# Full payment summary table
payment_summary <- hsr_payment_summary(my_report)
payment_summary

# Extract individual components (from file)
hsr_payment_penalty(my_report)

# Or existing extract
hsr_payment_penalty(payment_summary)
```

---

hsr\_read

---

*Parse a Hospital-Specific Report (HSR)*


---

**Description**

Parses a Hospital-Specific Report (HSR) into a lightweight internal object that can be reused by downstream helpers. Phase 1 support is limited to the legacy Excel-based HSR format used through FY2025.

The parsed object preserves current legacy section outputs as much as possible, while also retaining raw parser metadata for debugging and future format support.

If `format` is left `NULL`, `hsr_detect_format()` is used to identify the supported parser to apply.

**Usage**

```
hsr_read(path, format = NULL, validate = TRUE, ...)
```

**Arguments**

path	File path to a report.
format	Optional format override. Defaults to NULL, in which case <code>hsr_detect_format()</code> is used.
validate	Should the parsed object be validated before returning? Defaults to TRUE.
...	Reserved for future parser-specific options.

**Value**

A `readmit_hsr` object.

**Examples**

```
report <- hsr_mock_reports("FY2025_HRRP_MockHSR.xlsx")
parsed <- hsr_read(report)
parsed
```

---

`hsr_readmission_risks` *Compute discharge-level readmission risks in a Hospital-Specific Report (HSR)*

---

**Description**

Computes the *predicted* and *expected* readmission risks for each eligible discharge in the specified cohort from parsed HSR components.

**Usage**

```
hsr_readmission_risks(file, cohort)
```

**Arguments**

file	A parsed HSR bundle or a local source bundle path.
cohort	Cohort to compute readmission risks for. One of <code>c("AMI", "COPD", "HF", "PN", "CABG", "HK")</code>

**Details**

The **readmission measure** is what **CMS** uses to grade performance in the **Hospital Readmissions Reduction Program (HRRP)**.

Individual discharges are assigned an adjusted readmission risk (based on clinical history), which then get aggregated into a hospital-level score and compared to peer groups for penalty determination. Specifically, a random-intercept logistic regression model is built for each cohort ([see methodology](#)) which serves as the basis for two (2) readmission risks assigned to each discharge:

- *Predicted*: Adjusted for patient-specific clinical factors plus the *hospital-specific* effect (random intercept term)

- *Expected*: Adjusted for patient-specific clinical factors plus the *hospital-average* effect

These quantities are then aggregated across all discharges and their ratio is taken to form the *Excess Readmission Ratio (ERR)*, which is then used as the cohort-specific comparison metric. Thus, it is a comparative measure of how likely patients are to be readmitted at *your* hospital versus the *average* hospital, given your hospital's clinical characteristics.

## Value

A `tibble::tibble()` containing the following columns:

- ID Number: The unique discharge identifier (see `hsr_discharges()`)
- Predicted: The predicted readmission risk for the discharge
- Expected: The expected readmission risk for the discharge

## Examples

```
bundle_path <- file.path(tempdir(), "hsr_bundle")
dir.create(bundle_path, recursive = TRUE)
utils::write.csv(
  data.frame(
    cohort = c("HF", "HF"),
    `ID Number` = c(1001, 1002),
    AGE = c(1, 0)
  ),
  file.path(bundle_path, "discharges.csv"),
  row.names = FALSE
)
utils::write.csv(
  data.frame(
    cohort = c("HF", "HF", "HF"),
    term = c("AGE", "HOSP_EFFECT", "AVG_EFFECT"),
    value = c(0.25, -0.10, -0.20)
  ),
  file.path(bundle_path, "coefficients.csv"),
  row.names = FALSE
)

# Compute readmission risks for HF discharges
hf_risks <- hsr_readmission_risks(bundle_path, "HF")
hf_risks

# Compute the ERR from scratch
hf_risks |>
dplyr::summarize(
  Discharges = dplyr::n(),
  Predicted = mean(Predicted),
  Expected = mean(Expected),
  ERR = Predicted / Expected
)
```

---

hsr\_resolve\_source      *Resolve an HSR source bundle*

---

### Description

Resolves a local source bundle for Hospital-Specific Report (HSR) parsing. The returned object inventories source artifacts but does not parse them into conceptual components.

### Usage

```
hsr_resolve_source(x, spec = hsr_parser_config(), ...)
```

### Arguments

x                      Path to a local HSR source bundle.  
spec                    A parser configuration created by `hsr_parser_config()`.  
...                     Reserved for future source-resolution options.

### Value

A `readmit_hsr_source` object.

### Examples

```
bundle_path <- file.path(tempdir(), "hsr_bundle")
dir.create(bundle_path, recursive = TRUE)
utils::write.csv(
  data.frame(metric = "payment_adjustment_factor", value = 0.98),
  file.path(bundle_path, "payment_summary.csv"),
  row.names = FALSE
)
hsr_resolve_source(bundle_path, spec = hsr_parser_config())
```

---

is\_hsr                      *Check whether an object is a parsed HSR*

---

### Description

Check whether an object is a parsed HSR

### Usage

```
is_hsr(x)
```

### Arguments

x                      An object to test.

**Value**

TRUE if x inherits from class `"readmit_hsr"`, otherwise FALSE.

**Examples**

```
report <- hsr_mock_reports("FY2025_HRRP_MockHSR.xlsx")
parsed <- hsr_read(report)
is_hsr(parsed)
is_hsr(report)
```

---

is_hsr_bundle	<i>Check whether an object is a parsed HSR bundle</i>
---------------	---

---

**Description**

Check whether an object is a parsed HSR bundle

**Usage**

```
is_hsr_bundle(x)
```

**Arguments**

x                    An object to test.

**Value**

TRUE if x inherits from class `"readmit_hsr_bundle"`, otherwise FALSE.

---

is_hsr_source	<i>Check whether an object is an HSR source bundle</i>
---------------	--

---

**Description**

Check whether an object is an HSR source bundle

**Usage**

```
is_hsr_source(x)
```

**Arguments**

x                    An object to test.

**Value**

TRUE if x inherits from class `"readmit_hsr_source"`, otherwise FALSE.

---

pdc\_read

*Import datasets from the Provider Data Catalog (PDC)*

---

## Description

Explore and import datasets directly from the [CMS Provider Data Catalog \(PDC\)](#).

- `pdc_topics()`: Retrieves the list of topics (subcategories) that data is available for
- `pdc_datasets()`: Retrieves identifiers, names, descriptions, and other metadata associated with the datasets in the (optionally specified) topics
- `pdc_read()`: Imports a full dataset for the given identifier (`datasetid`). These are found in `pdc_datasets()`.

## Usage

```
pdc_read(datasetid, ...)  
  
pdc_datasets(topics = NULL)  
  
pdc_topics()
```

## Arguments

<code>datasetid</code>	A dataset identifier (e.g., from <code>pdc_datasets()</code> )
<code>...</code>	Additional arguments passed to <code>readr::read_csv()</code>
<code>topics</code>	A topic to list dataset metadata for (e.g., from <code>pdc_topics()</code> )

## Value

A character vector listing available data topics, or a `tibble::tibble()` containing the requested data/metadata.

## Examples

```
# 1. See list of available data topics  
pdc_topics()  
  
# 2. See list of datasets available for a topic  
hospital_data <- pdc_datasets("Hospitals")  
hospital_data  
  
# Find a dataset you want  
hospital_data |>  
  dplyr::filter(  
    stringr::str_detect(  
      title,  
      pattern = "(?i)readmission"    )  
  )
```

```
)  
)
```

```
# 3. Use that data set ID to import  
pd_read("9n3s-kdb3")
```

# Index

## \* datasets

hrrp\_keydates, 3

hrrp\_cohort\_inclusion (hrrp\_keydates), 3

hrrp\_get\_dates, 2

hrrp\_keydates, 2, 3, 3

hrrp\_payment\_periods (hrrp\_keydates), 3

hrrp\_performance\_periods  
(hrrp\_keydates), 3

hrrp\_review\_periods (hrrp\_keydates), 3

hrrp\_snapshot\_dates (hrrp\_keydates), 3

hsr\_coefficients, 5

hsr\_cohort\_summary, 6

hsr\_count\_dual\_stays  
(hsr\_payment\_summary), 11

hsr\_count\_total\_stays  
(hsr\_payment\_summary), 11

hsr\_detect\_format, 6

hsr\_detect\_format(), 13

hsr\_discharges, 7

hsr\_discharges(), 14

hsr\_dual\_proportion  
(hsr\_payment\_summary), 11

hsr\_dual\_stays, 8

hsr\_mock\_reports, 9

hsr\_neutrality\_modifier  
(hsr\_payment\_summary), 11

hsr\_parse, 10

hsr\_parser\_config, 10

hsr\_payment\_adjustment\_factor  
(hsr\_payment\_summary), 11

hsr\_payment\_penalty  
(hsr\_payment\_summary), 11

hsr\_payment\_summary, 11

hsr\_peer\_group (hsr\_payment\_summary), 11

hsr\_read, 12

hsr\_readmission\_risks, 13

hsr\_resolve\_source, 15

is\_hsr, 15

is\_hsr\_bundle, 16

is\_hsr\_source, 16

pdcdatasets (pdc\_read), 17

pdc\_read, 17

pdc\_topics (pdc\_read), 17

readr::read\_csv(), 17

readxl::readxl\_example(), 9

tibble::tibble, 2

tibble::tibble(), 5-8, 12, 14, 17