

# Package ‘pacta.loanbook’

May 9, 2026

**Title** Easily Install and Load PACTA for Banks Packages

**Version** 0.1.1

**Description** PACTA (Paris Agreement Capital Transition Assessment) for Banks is a tool that allows banks to calculate the climate alignment of their corporate lending portfolios. This package is designed to make it easy to install and load multiple PACTA for Banks packages in a single step. It also provides thorough documentation - the PACTA for Banks cookbook at [https://rmi-pacta.github.io/pacta.loanbook/articles/cookbook\\_overview.html](https://rmi-pacta.github.io/pacta.loanbook/articles/cookbook_overview.html) - on how to run a PACTA for Banks analysis. This covers prerequisites for the analysis, the separate steps of running the analysis, the interpretation of PACTA for Banks results, and advanced use cases.

**License** MIT + file LICENSE

**URL** <https://rmi-pacta.github.io/pacta.loanbook/>,  
<https://github.com/RMI-PACTA/pacta.loanbook>

**BugReports** <https://github.com/RMI-PACTA/pacta.loanbook/issues>

**Depends** R (>= 4.1)

**Imports** cli, dplyr, ggrepel, magrittr, purrr, r2dii.analysis (>= 0.5.1), r2dii.data (>= 0.6.0), r2dii.match (>= 0.4.1), r2dii.plot (>= 0.5.2), rlang, rstudioapi, scales, tibble, tidyselect

**Suggests** covr, DiagrammeR, ggplot2, gt (>= 0.11.0), knitr (>= 1.42), readr, readxl, rmarkdown (>= 2.19), roxygen2 (>= 4.1.0), spelling, testthat (>= 3.2.2), tidy, writexl

**Encoding** UTF-8

**Language** en-US

**RoxygenNote** 7.3.2

**Config/Needs/website** rmi-pacta/pacta.pkgdown.rmitemplate, rmarkdown

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Jacob Kastl [aut, cre, ctr] (ORCID: <https://orcid.org/0009-0000-8281-8129>),  
 Jackson Hoffart [aut, ctr] (ORCID: <https://orcid.org/0000-0002-8600-5042>),  
 CJ Yetman [aut, ctr] (ORCID: <https://orcid.org/0000-0001-5099-9500>),  
 RMI [cph, fnd] (ROR: <https://ror.org/03anfar33>)

**Maintainer** Jacob Kastl <jacob.kastl@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-06-24 09:50:02 UTC

## Contents

abcd_demo . . . . .	3
co2_intensity_scenario_demo . . . . .	4
crucial_lbk . . . . .	5
data_dictionary . . . . .	6
gics_classification . . . . .	6
increasing_or_decreasing . . . . .	7
isic_classification . . . . .	8
iso_codes . . . . .	9
loanbook_demo . . . . .	9
market_share_demo . . . . .	10
match_name . . . . .	11
nace_classification . . . . .	14
naics_classification . . . . .	15
overwrite_demo . . . . .	16
pacta_loanbook_conflicts . . . . .	16
pacta_loanbook_deps . . . . .	17
pacta_loanbook_logo . . . . .	18
pacta_loanbook_packages . . . . .	19
pacta_loanbook_sitrep . . . . .	19
pacta_loanbook_update . . . . .	20
palette_colours . . . . .	21
plot_emission_intensity . . . . .	22
plot_techmix . . . . .	22
plot_trajectory . . . . .	23
prep_emission_intensity . . . . .	24
prep_techmix . . . . .	25
prep_trajectory . . . . .	27
prioritize . . . . .	28
prioritize_level . . . . .	30
psic_classification . . . . .	31
qplot_emission_intensity . . . . .	32
qplot_techmix . . . . .	33
qplot_trajectory . . . . .	34
recode_metric_techmix . . . . .	35
recode_metric_trajectory . . . . .	36

region_isos . . . . .	37
region_isos_demo . . . . .	37
scale_colour_r2dii . . . . .	38
scale_colour_r2dii_sector . . . . .	39
scale_colour_r2dii_tech . . . . .	40
scale_fill_r2dii . . . . .	41
scale_fill_r2dii_sector . . . . .	42
scale_fill_r2dii_tech . . . . .	43
scenario_colours . . . . .	44
scenario_demo_2020 . . . . .	45
sda_demo . . . . .	46
sector_classifications . . . . .	46
sector_colours . . . . .	47
sic_classification . . . . .	48
spell_out_technology . . . . .	49
target_market_share . . . . .	50
target_sda . . . . .	51
technology_colours . . . . .	53
theme_2dii . . . . .	54
to_title . . . . .	55
<b>Index</b>	<b>57</b>

---

abcd\_demo

*An asset-based company dataset for demonstration*


---

## Description

Fake data about physical assets (e.g. wind turbine power plant capacities), aggregated to company-level. These data are used to assess the climate alignment of financial portfolios. It imitates data from market-intelligence databases.

Demo datasets are synthetic because most financial data is strictly private; they help to demonstrate and test the implementation in R of 'PACTA' (<https://www.transitionmonitor.com/>).

## Usage

```
abcd_demo
```

## Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 4972 rows and 12 columns.

## Definitions

- `company_id` (character): The id of the company owning the asset created by the data provider.,
- `* emission_factor` (double): Company level emission factor of the technology.,
- `* emission_factor_unit` (character): The units that the emission factor is measured in.,
- `* is_ultimate_owner` (logical): Flag if company is the ultimate parent in our database.,
- `* lei` (character): The legal entity identifier of the company owning the asset.,
- `* name_company` (character): The name of the company owning the asset.,
- `* plant_location` (character): Country where asset is located.,
- `* production` (double): Company level production of the technology.,
- `* production_unit` (character): The units that production is measured in.,
- `* sector` (character): Sector to which the asset belongs.,
- `* technology` (character): Technology implemented by the asset.,
- `* year` (integer): Year at which the production value is predicted.

## See Also

Other demo data: [co2\\_intensity\\_scenario\\_demo](#), [loanbook\\_demo](#), [market\\_share\\_demo](#), [overwrite\\_demo](#), [region\\_isos\\_demo](#), [scenario\\_demo\\_2020](#), [sda\\_demo](#)

## Examples

```
head(abcd_demo)
```

---

```
co2_intensity_scenario_demo
```

*A prepared CO<sub>2</sub> intensity climate scenario dataset for demonstration*

---

## Description

Fake CO<sub>2</sub> intensity climate scenario dataset, prepared for the software PACTA (Paris Agreement Capital Transition Assessment). It imitates climate scenario data (e.g. from the International Energy Agency (IEA)) including the change through time in production across industrial sectors.

Demo datasets are synthetic because most financial data is strictly private; they help to demonstrate and test the implementation in R of 'PACTA' (<https://www.transitionmonitor.com/>).

## Usage

```
co2_intensity_scenario_demo
```

## Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 22 rows and 7 columns.

## Definitions

- `emission_factor` (double): The target sector level emissions factor that the scenario prescribes., \* `emission_factor_unit` (character): The units that the emissions factor is measured in., \* `region` (character): The region to which the pathway is relevant., \* `scenario` (character): The name of the scenario., \* `scenario_source` (character): The source publication from which the scenario was taken., \* `sector` (character): The sector to which the scenario prescribes a pathway., \* `year` (integer): The year at which the pathway value is prescribed.

## See Also

Other demo data: [abcd\\_demo](#), [loanbook\\_demo](#), [market\\_share\\_demo](#), [overwrite\\_demo](#), [region\\_isos\\_demo](#), [scenario\\_demo\\_2020](#), [sda\\_demo](#)

## Examples

```
head(co2_intensity_scenario_demo)
```

---

crucial\_lbk

*Crucial* loanbook columns for `match_name()`

---

## Description

This is a helper to select the minimum loanbook columns you need to run `match_name()`. Using more columns may use too much time and memory.

## Usage

```
crucial_lbk()
```

## Value

A character vector.

## See Also

Other matching functions: [match\\_name\(\)](#), [prioritize\(\)](#), [prioritize\\_level\(\)](#)

## Examples

```
crucial_lbk()
```

---

data_dictionary	<i>Data Dictionary</i>
-----------------	------------------------

---

**Description**

A table of column names and descriptions of data frames used or exported by the functions in this package.

**Usage**

```
data_dictionary
```

**Format**

```
data_dictionary:  
dataset Name of the dataset  
column Name of the column  
typeof Type of the column  
definition Definition of the column
```

**Examples**

```
data_dictionary
```

---

gics_classification	<i>Dataset to bridge (translate) common sector-classification codes</i>
---------------------	---

---

**Description**

This dataset serves as a translation key between common sector-classification systems and sectors relevant to the 'PACTA' tool (<https://www.transitionmonitor.com/>).

**Usage**

```
gics_classification
```

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 282 rows and 5 columns.

## Definitions

- `borderline` (logical): Flag indicating if PACTA sector and classification code are a borderline match. The value `TRUE` indicates that the match is uncertain between the PACTA sector and the classification. The value `FALSE` indicates that the match is certainly perfect or the classification is certainly out of PACTA's scope., \* `code` (character): Original GICS code., \* `description` (character): Original GICS description., \* `sector` (character): Associated PACTA sector., \* `version` (character): Column identifying to which GICS version the code belongs.

## Details

Classification datasets help to standardize sector classification codes from the wild to a relevant subset including 'power', 'oil and gas', 'coal', 'automotive', 'aviation', 'concrete', 'steel', and 'shipping'.

## See Also

Other datasets: [increasing\\_or\\_decreasing](#), [isic\\_classification](#), [iso\\_codes](#), [nace\\_classification](#), [naics\\_classification](#), [palette\\_colours](#), [psic\\_classification](#), [region\\_isos](#), [scenario\\_colours](#), [sector\\_classifications](#), [sector\\_colours](#), [sic\\_classification](#), [technology\\_colours](#)

## Examples

```
head(gics_classification)
```

---

```
increasing_or_decreasing
```

*Determine if a technology is increasing or decreasing*

---

## Description

This dataset provides a simple lookup table to determine if a technology is meant to increase or decrease to align with a scenario that predicts a less than 2 degree temperature rise.

## Usage

```
increasing_or_decreasing
```

## Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 20 rows and 3 columns.

## Definitions

- `increasing_or_decreasing` (character): If the technology is increasing or decreasing, as defined by the Paris-aligned IEA scenarios., \* `sector` (character): The sector to which the technology belongs., \* `technology` (character): The technology sub-category within the sector.

## See Also

Other datasets: [gics\\_classification](#), [isic\\_classification](#), [iso\\_codes](#), [nace\\_classification](#), [naics\\_classification](#), [palette\\_colours](#), [psic\\_classification](#), [region\\_isos](#), [scenario\\_colours](#), [sector\\_classifications](#), [sector\\_colours](#), [sic\\_classification](#), [technology\\_colours](#)

## Examples

```
head(increasing_or_decreasing)
```

---

isic\_classification     *Dataset to bridge (translate) common sector-classification codes*

---

## Description

This dataset serves as a translation key between common sector-classification systems and sectors relevant to the 'PACTA' tool (<https://www.transitionmonitor.com/>).

## Usage

```
isic_classification
```

## Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 830 rows and 6 columns.

## Definitions

- `borderline` (logical): Flag indicating if PACTA sector and classification code are a borderline match. The value `TRUE` indicates that the match is uncertain between the PACTA sector and the classification. The value `FALSE` indicates that the match is certainly perfect or the classification is certainly out of PACTA's scope., \* `code` (character): ISIC Rev 5 code with top-level letter prepended., \* `description` (character): Original ISIC Rev 5 title., \* `original_code` (character): Original ISIC Rev 5 code., \* `revision` (character): Column identifying to which ISIC revision the code belongs., \* `sector` (character): Associated PACTA sector.

## Details

Classification datasets help to standardize sector classification codes from the wild to a relevant subset including 'power', 'oil and gas', 'coal', 'automotive', 'aviation', 'concrete', 'steel', and 'shipping'.

## See Also

Other datasets: [gics\\_classification](#), [increasing\\_or\\_decreasing](#), [iso\\_codes](#), [nace\\_classification](#), [naics\\_classification](#), [palette\\_colours](#), [psic\\_classification](#), [region\\_isos](#), [scenario\\_colours](#), [sector\\_classifications](#), [sector\\_colours](#), [sic\\_classification](#), [technology\\_colours](#)

**Examples**

```
head(isic_classification)
```

---

iso_codes	<i>Countries and codes</i>
-----------	----------------------------

---

**Description**

This dataset maps countries to codes.

For information about the ISO standard for country codes see <https://www.iso.org/iso-3166-country-codes.html>.

**Usage**

```
iso_codes
```

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 286 rows and 2 columns.

**Definitions**

- `country` (character): Country name., \* `country_iso` (character): Corresponding ISO code.

**See Also**

Other datasets: [gics\\_classification](#), [increasing\\_or\\_decreasing](#), [isic\\_classification](#), [nace\\_classification](#), [naics\\_classification](#), [palette\\_colours](#), [psic\\_classification](#), [region\\_isos](#), [scenario\\_colours](#), [sector\\_classifications](#), [sector\\_colours](#), [sic\\_classification](#), [technology\\_colours](#)

**Examples**

```
head(iso_codes)
```

---

loanbook_demo	<i>A loanbook dataset for demonstration</i>
---------------	---

---

**Description**

Fake financial portfolio.

Demo datasets are synthetic because most financial data is strictly private; they help to demonstrate and test the implementation in R of 'PACTA' (<https://www.transitionmonitor.com/>).

**Usage**

```
loanbook_demo
```

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 283 rows and 13 columns.

**Definitions**

- `id_direct_loantaker` (character): Borrower identifier unique to each borrower/sector combination in `loanbook`., \* `id_loan` (character): Unique loan identifier., \* `id_ultimate_parent` (character): Ultimate parent identifier unique to each ultimate parent/sector combination., \* `isin_direct_loantaker` (logical): Optional input: providing the isin identifier of the direct loan taker to improve the matching coverage., \* `lei_direct_loantaker` (logical): Optional input: providing the lei (legal entity identifier) of the direct loan taker to improve the matching coverage., \* `loan_size_credit_limit` (double): Total credit limit or exposure at default., \* `loan_size_credit_limit_currency` (character): Currency corresponding to credit limit., \* `loan_size_outstanding` (double): Amount drawn by borrower from total credit limit., \* `loan_size_outstanding_currency` (character): Currency corresponding to outstandings., \* `name_direct_loantaker` (character): Name of the company directly taking the loan., \* `name_ultimate_parent` (character): Name of the ultimate parent company to which the borrower belongs. Can be the same as borrower., \* `sector_classification_direct_loantaker` (double): Sector classification code of the direct loantaker., \* `sector_classification_system` (character): Name of the sector classification standard being used.

**See Also**

Other demo data: [abcd\\_demo](#), [co2\\_intensity\\_scenario\\_demo](#), [market\\_share\\_demo](#), [overwrite\\_demo](#), [region\\_isos\\_demo](#), [scenario\\_demo\\_2020](#), [sda\\_demo](#)

**Examples**

```
head(loanbook_demo)
```

---

market_share_demo	<i>An example of a market_share_demo-like dataset</i>
-------------------	---

---

**Description**

Dataset imitating the output of `r2dii.analysis::target_market_share()`.

**Usage**

```
market_share_demo
```

**Format**

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 802 rows and 10 columns.

**See Also**

Other demo data: [abcd\\_demo](#), [co2\\_intensity\\_scenario\\_demo](#), [loanbook\\_demo](#), [overwrite\\_demo](#), [region\\_isos\\_demo](#), [scenario\\_demo\\_2020](#), [sda\\_demo](#)

**Examples**

```
market_share_demo
```

---

match_name	<i>Match a loanbook to asset-based company data (abcd) by the name_* columns</i>
------------	--

---

**Description**

match\_name() scores the match between names in a loanbook dataset (columns can be name\_direct\_loantaker, name\_intermediate\_parent\* and name\_ultimate\_parent) with names in an asset-based company data (column name\_company). The raw names are first internally transformed, and aliases are assigned. The similarity between aliases in each of the loanbook and abcd is scored using [stringdist::stringsim\(\)](#).

**Usage**

```
match_name(
  loanbook,
  abcd,
  by_sector = TRUE,
  min_score = 0.8,
  method = "jw",
  p = 0.1,
  overwrite = NULL,
  join_id = NULL,
  sector_classification = default_sector_classification(),
  ...
)
```

**Arguments**

loanbook, abcd	data frames structured like <a href="#">r2dii.data::loanbook_demo</a> and <a href="#">r2dii.data::abcd_demo</a> .
by_sector	Should names only be compared if companies belong to the same sector?
min_score	A number between 0-1, to set the minimum score threshold. A score of 1 is a perfect match.
method	Method for distance calculation. One of c("osa", "lv", "dl", "hamming", "lcs", "qgram", "cosine", "jaccard", "jw", "soundex"). See <a href="#">stringdist::stringdist-metrics</a> .
p	Prefix factor for Jaro-Winkler distance. The valid range for p is $0 \leq p \leq 0.25$ . If p=0 (default), the Jaro-distance is returned. Applies only to method='jw'.

overwrite	A data frame used to overwrite the sector and/or name columns of a particular direct loantaker or ultimate parent. To overwrite only sector, the value in the name column should be NA and vice-versa. This file can be used to manually match loanbook companies to abcd.
join_id	A join specification passed to <code>dplyr::inner_join()</code> . If a character string, it assumes identical join columns between loanbook and abcd. If a named character vector, it uses the name as the join column of loanbook and the value as the join column of abcd.
sector_classification	A data frame containing sector classifications in the same format as <code>r2dii.data::sector_classification</code> . The default value is <code>r2dii.data::sector_classifications</code> .
...	Arguments passed on to <code>stringdist::stringsim()</code> .

### Value

A data frame with the same groups (if any) and columns as loanbook, and the additional columns:

- `id_2dii` - an id used internally by `match_name()` to distinguish companies
- `level` - the level of granularity that the loan was matched at (e.g `direct_loantaker` or `ultimate_parent`)
- `sector` - the sector of the loanbook company
- `sector_abcd` - the sector of the abcd company
- `name` - the name of the loanbook company
- `name_abcd` - the name of the abcd company
- `score` - the score of the match (manually set this to 1 prior to calling `prioritize()` to validate the match)
- `source` - determines the source of the match. (equal to loanbook unless the match is from `overwrite`)

The returned rows depend on the argument `min_value` and the result of the column score for each loan: \* If any row has score equal to 1, `match_name()` returns all rows where score equals 1, dropping all other rows. \* If no row has score equal to 1, `match_name()` returns all rows where score is equal to or greater than `min_score`. \* If there is no match the output is a 0-row tibble with the expected column names – for type stability.

### Assigning aliases

The transformation process used to compare names between loanbook and abcd datasets applies best practices commonly used in name matching algorithms:

- Remove special characters.
- Replace language specific characters.
- Abbreviate certain names to reduce their importance in the matching.
- Spell out numbers to increase their importance.

## Handling grouped data

This function ignores but preserves existing groups.

## See Also

Other matching functions: [crucial\\_lbk\(\)](#), [prioritize\(\)](#), [prioritize\\_level\(\)](#)

## Examples

```
library(r2dii.data)
library(tibble)

# Small data for examples
loanbook <- head(loanbook_demo, 50)
abcd <- head(abcd_demo, 50)

match_name(loanbook, abcd)

match_name(loanbook, abcd, min_score = 0.9)

# match on LEI
loanbook <- tibble(
  sector_classification_system = "NACE",
  sector_classification_direct_loantaker = "D35.11",
  id_ultimate_parent = "UP15",
  name_ultimate_parent = "Won't fuzzy match",
  id_direct_loantaker = "C294",
  name_direct_loantaker = "Won't fuzzy match",
  lei_direct_loantaker = "LEI123"
)

abcd <- tibble(
  name_company = "alpine knits india pvt. limited",
  sector = "power",
  lei = "LEI123"
)

match_name(loanbook, abcd, join_id = c(lei_direct_loantaker = "lei"))

# Use your own `sector_classifications`
your_classifications <- tibble(
  sector = "power",
  borderline = FALSE,
  code = "D35.11",
  code_system = "XYZ"
)

loanbook <- tibble(
  sector_classification_system = "XYZ",
  sector_classification_direct_loantaker = "D35.11",
  id_ultimate_parent = "UP15",
  name_ultimate_parent = "Alpine Knits India Pvt. Limited",
```

```

id_direct_loantaker = "C294",
name_direct_loantaker = "Yuamen Xinneng Thermal Power Co Ltd"
)

abcd <- tibble(
  name_company = "alpine knits india pvt. limited",
  sector = "power"
)

match_name(loanbook, abcd, sector_classification = your_classifications)

```

---

nace\_classification     *Dataset to bridge (translate) common sector-classification codes*

---

### Description

This dataset serves as a translation key between common sector-classification systems and sectors relevant to the 'PACTA' tool (<https://www.transitionmonitor.com/>).

### Usage

```
nace_classification
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1047 rows and 6 columns.

### Definitions

- `borderline` (logical): Flag indicating if PACTA sector and classification code are a borderline match. The value `TRUE` indicates that the match is uncertain between the PACTA sector and the classification. The value `FALSE` indicates that the match is certainly perfect or the classification is certainly out of PACTA's scope., \* `code` (character): NACE version 2.1 code with top-level letter prepended., \* `description` (character): Original NACE version 2.1 description., \* `original_code` (character): Original NACE version 2.1 code., \* `sector` (character): Associated PACTA sector., \* `version` (character): Column identifying to which NACE version the code belongs.

### Details

Classification datasets help to standardize sector classification codes from the wild to a relevant subset including 'power', 'oil and gas', 'coal', 'automotive', 'aviation', 'concrete', 'steel', and 'shipping'.

### See Also

Other datasets: [gics\\_classification](#), [increasing\\_or\\_decreasing](#), [isic\\_classification](#), [iso\\_codes](#), [naics\\_classification](#), [palette\\_colours](#), [psic\\_classification](#), [region\\_isos](#), [scenario\\_colours](#), [sector\\_classifications](#), [sector\\_colours](#), [sic\\_classification](#), [technology\\_colours](#)

## Examples

```
head(nace_classification)
```

---

naics\_classification *Dataset to bridge (translate) common sector-classification codes*

---

## Description

This dataset serves as a translation key between common sector-classification systems and sectors relevant to the 'PACTA' tool (<https://www.transitionmonitor.com/>).

## Usage

```
naics_classification
```

## Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 2125 rows and 5 columns.

## Definitions

- `borderline` (logical): Flag indicating if PACTA sector and classification code are a borderline match. The value `TRUE` indicates that the match is uncertain between the PACTA sector and the classification. The value `FALSE` indicates that the match is certainly perfect or the classification is certainly out of PACTA's scope..
- `code` (character): Six-digit NAICS code..
- `description` (character): Original NAICS sector title..
- `sector` (character): Associated PACTA sector..
- `version` (character): Column identifying which year the classification was published in..

## Details

Classification datasets help to standardize sector classification codes from the wild to a relevant subset including 'power', 'oil and gas', 'coal', 'automotive', 'aviation', 'concrete', 'steel', and 'shipping'.

## See Also

Other datasets: [gics\\_classification](#), [increasing\\_or\\_decreasing](#), [isic\\_classification](#), [iso\\_codes](#), [nace\\_classification](#), [palette\\_colours](#), [psic\\_classification](#), [region\\_isos](#), [scenario\\_colours](#), [sector\\_classifications](#), [sector\\_colours](#), [sic\\_classification](#), [technology\\_colours](#)

## Examples

```
head(naics_classification)
```

---

overwrite_demo	<i>A demonstration dataset used to overwrite specific entity names or sectors</i>
----------------	---

---

### Description

Fake dataset used to manually link loanbook entities to mismatched asset level entities.

Demo datasets are synthetic because most financial data is strictly private; they help to demonstrate and test the implementation in R of 'PACTA' (<https://www.transitionmonitor.com/>).

### Usage

```
overwrite_demo
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 2 rows and 5 columns.

### Definitions

- `id_2dii` (character): IDs of the entities to overwrite., \* `level` (character): Which level should be overwritten (e.g. `direct_loantaker` or `ultimate_parent`)., \* `name` (character): Overwrite name (if only overwriting sector, type NA)., \* `sector` (character): Overwrite sector (if only overwriting name, type NA)., \* `source` (character): What is the source of this information (leave as "manual" for now, may remove this flag later).

### See Also

Other demo data: [abcd\\_demo](#), [co2\\_intensity\\_scenario\\_demo](#), [loanbook\\_demo](#), [market\\_share\\_demo](#), [region\\_isos\\_demo](#), [scenario\\_demo\\_2020](#), [sda\\_demo](#)

### Examples

```
head(overwrite_demo)
```

---

pacta_loanbook_conflicts	<i>Conflicts between the {pacta_loanbook} and other packages</i>
--------------------------	--

---

### Description

This function lists all the conflicts between packages in the `{pacta_loanbook}` and other packages that you have loaded.

**Usage**

```
pacta_loanbook_conflicts(only = NULL)
```

**Arguments**

`only` Set this to a character vector to restrict to conflicts only with these packages.

**Details**

There are four conflicts that are deliberately ignored: `intersect`, `union`, `setequal`, and `setdiff` from `dplyr`. These functions make the base equivalents generic, so shouldn't negatively affect any existing code.

**Value**

a `pacta_loanbook_conflicts` classed list which will print a list of conflicts to the console in interactive sessions, or `NULL` if no conflicts are found.

**See Also**

Other utility functions: [pacta\\_loanbook\\_deps\(\)](#), [pacta\\_loanbook\\_log\(\)](#), [pacta\\_loanbook\\_packages\(\)](#), [pacta\\_loanbook\\_sitrep\(\)](#), [pacta\\_loanbook\\_update\(\)](#)

**Examples**

```
pacta_loanbook_conflicts()
```

---

```
pacta_loanbook_deps
```

*List all {pacta.loanbook} dependencies*

---

**Description**

List all {pacta.loanbook} dependencies

**Usage**

```
pacta_loanbook_deps(recursive = FALSE, repos = getOption("repos"))
```

**Arguments**

`recursive` If `TRUE`, will also list all dependencies of {pacta.loanbook} packages.  
`repos` The repositories to use to check for updates. Defaults to `getOption("repos")`.

**Value**

a tibble containing the local and CRAN versions of dependent packages.

**See Also**

Other utility functions: [pacta\\_loanbook\\_conflicts\(\)](#), [pacta\\_loanbook\\_logo\(\)](#), [pacta\\_loanbook\\_packages\(\)](#), [pacta\\_loanbook\\_sitrep\(\)](#), [pacta\\_loanbook\\_update\(\)](#)

**Examples**

```
pacta_loanbook_deps(repos = "https://cran.r-project.org")
```

---

`pacta_loanbook_logo`    *The {pacta\_loanbook} logo, using ASCII or Unicode characters*

---

**Description**

Use `cli::ansi_strip()` to get rid of the colors.

**Usage**

```
pacta_loanbook_logo(unicode = cli::is_utf8_output())
```

**Arguments**

`unicode`            Whether to use Unicode symbols. Default is TRUE on UTF-8 platforms.

**Value**

a `pacta_loanbook_logo` classed `cli_ansi_string` which will print the PACTA logo in the console in interactive sessions.

**See Also**

Other utility functions: [pacta\\_loanbook\\_conflicts\(\)](#), [pacta\\_loanbook\\_deps\(\)](#), [pacta\\_loanbook\\_packages\(\)](#), [pacta\\_loanbook\\_sitrep\(\)](#), [pacta\\_loanbook\\_update\(\)](#)

**Examples**

```
pacta_loanbook_logo()
```

---

pacta\_loanbook\_packages

*List all packages in {pacta.loanbook}*

---

**Description**

List all packages in {pacta.loanbook}

**Usage**

```
pacta_loanbook_packages(include_self = TRUE)
```

**Arguments**

include\_self    Include {pacta.loanbook} in the list?

**Value**

a character vector containing the names of packages imported by {pacta.loanbook}.

**See Also**

Other utility functions: [pacta\\_loanbook\\_conflicts\(\)](#), [pacta\\_loanbook\\_deps\(\)](#), [pacta\\_loanbook\\_logo\(\)](#), [pacta\\_loanbook\\_sitrep\(\)](#), [pacta\\_loanbook\\_update\(\)](#)

**Examples**

```
pacta_loanbook_packages()
```

---

pacta\_loanbook\_sitrep    *Get a situation report on {pacta.loanbook}*

---

**Description**

This function gives a quick overview of the versions of R and RStudio as well as the {pacta.loanbook} package. It's primarily designed to help you get a quick idea of what's going on when you're helping someone else debug a problem.

**Usage**

```
pacta_loanbook_sitrep(repos = getOption("repos"))
```

**Arguments**

repos                    The repositories to use to check for updates. Defaults to `getOption("repos")`.

**Value**

returns NULL invisibly. The function is called for its side effect of printing a situation report of {pacta\_loanbook} and its core packages.

**See Also**

Other utility functions: [pacta\\_loanbook\\_conflicts\(\)](#), [pacta\\_loanbook\\_deps\(\)](#), [pacta\\_loanbook\\_logo\(\)](#), [pacta\\_loanbook\\_packages\(\)](#), [pacta\\_loanbook\\_update\(\)](#)

**Examples**

```
pacta_loanbook_sitrep(repos = "https://cran.r-project.org")
```

---

pacta\_loanbook\_update *Update {pacta\_loanbook} packages*

---

**Description**

This will check to see if all {pacta\_loanbook} packages (and optionally, their dependencies) are up-to-date, and will install after an interactive confirmation.

**Usage**

```
pacta_loanbook_update(recursive = FALSE, repos = getOption("repos"))
```

**Arguments**

**recursive**        If TRUE, will also list all dependencies of {pacta\_loanbook} packages.  
**repos**            The repositories to use to check for updates. Defaults to `getOption("repos")`.

**Value**

returns NULL invisibly. The function is called for its side effect of printing the status of locally installed, relevant packages.

**See Also**

Other utility functions: [pacta\\_loanbook\\_conflicts\(\)](#), [pacta\\_loanbook\\_deps\(\)](#), [pacta\\_loanbook\\_logo\(\)](#), [pacta\\_loanbook\\_packages\(\)](#), [pacta\\_loanbook\\_sitrep\(\)](#)

**Examples**

```
pacta_loanbook_update(repos = "https://cran.r-project.org")
```

---

palette_colours	<i>Colour datasets</i>
-----------------	------------------------

---

## Description

All datasets have at least two columns:

- label: Text label of the colour.
- hex: Hex code of the colour.

## Usage

```
palette_colours
```

## Format

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

## Details

In `scenario_colours`, colours are ordered from red to green to be used in trajectory charts.

## See Also

Other datasets: [gics\\_classification](#), [increasing\\_or\\_decreasing](#), [isic\\_classification](#), [iso\\_codes](#), [nace\\_classification](#), [naics\\_classification](#), [psic\\_classification](#), [region\\_isos](#), [scenario\\_colours](#), [sector\\_classifications](#), [sector\\_colours](#), [sic\\_classification](#), [technology\\_colours](#)

## Examples

```
palette_colours
```

```
scenario_colours
```

```
sector_colours
```

```
technology_colours
```

---

plot\_emission\_intensity  
*Create an emission intensity plot*

---

**Description**

Create an emission intensity plot

**Usage**

```
plot_emission_intensity(data)
```

**Arguments**

data                    A data frame like the output of prep\_emission\_intensity().

**Value**

An object of class "ggplot".

**See Also**

Other plotting functions: [plot\\_techmix\(\)](#), [plot\\_trajectory\(\)](#), [prep\\_emission\\_intensity\(\)](#), [prep\\_techmix\(\)](#), [prep\\_trajectory\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_techmix\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_colour\\_r2dii\\_tech\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\\_tech\(\)](#), [spell\\_out\\_technology\(\)](#), [theme\\_2dii\(\)](#), [to\\_title\(\)](#)

**Examples**

```
# plot with `qplot_emission_intensity()` parameters
data <- subset(sda_demo, sector == "cement" & region == "global")
data <- prep_emission_intensity(data, span_5yr = TRUE, convert_label = to_title)

plot_emission_intensity(data)
```

---

plot\_techmix                    *Create a techmix plot*

---

**Description**

Create a techmix plot

**Usage**

```
plot_techmix(data)
```

**Arguments**

data                    A data frame like the output of prep\_techmix().

**Value**

An object of class "ggplot".

**See Also**

Other plotting functions: [plot\\_emission\\_intensity\(\)](#), [plot\\_trajectory\(\)](#), [prep\\_emission\\_intensity\(\)](#), [prep\\_techmix\(\)](#), [prep\\_trajectory\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_techmix\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_colour\\_r2dii\\_tech\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\\_tech\(\)](#), [spell\\_out\\_technology\(\)](#), [theme\\_2dii\(\)](#), [to\\_title\(\)](#)

**Examples**

```
# plot with `qplot_techmix()` parameters
data <- subset(
  market_share_demo,
  scenario_source == "demo_2020" &
  sector == "power" &
  region == "global" &
  metric %in% c("projected", "corporate_economy", "target_sds")
)
data <- prep_techmix(
  data,
  span_5yr = TRUE,
  convert_label = recode_metric_techmix,
  convert_tech_label = spell_out_technology
)

plot_techmix(data)
```

---

plot\_trajectory                    *Create a trajectory plot*

---

**Description**

Create a trajectory plot

**Usage**

```
plot_trajectory(data, center_y = FALSE, perc_y_scale = FALSE)
```

**Arguments**

data	A data frame like the outputs of <code>prep_trajectory()</code> . <ul style="list-style-type: none"> <li>• (Optional) If present, the column label is used for data labels.</li> </ul>
center_y	Logical. Use TRUE to center the y-axis around start value (the default behavior of <code>qplot_trajectory()</code> ), or use FALSE to not center.
perc_y_scale	Logical. FALSE defaults to using no label conversion. Use TRUE to convert labels on y-axis to percentage using <code>scales::percent</code> (the default behavior of <code>qplot_trajectory()</code> ).

**Value**

An object of class "ggplot".

**See Also**

Other plotting functions: [plot\\_emission\\_intensity\(\)](#), [plot\\_techmix\(\)](#), [prep\\_emission\\_intensity\(\)](#), [prep\\_techmix\(\)](#), [prep\\_trajectory\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_techmix\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_colour\\_r2dii\\_tech\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\\_tech\(\)](#), [spell\\_out\\_technology\(\)](#), [theme\\_2dii\(\)](#), [to\\_title\(\)](#)

**Examples**

```
# plot with `qplot_trajectory()` parameters
data <- subset(
  market_share_demo,
  sector == "power" &
  technology == "renewables" &
  region == "global" &
  scenario_source == "demo_2020"
)
data <- prep_trajectory(data)

plot_trajectory(
  data,
  center_y = TRUE,
  perc_y_scale = TRUE
)
```

---

```
prep_emission_intensity
```

*Prepare data for a emission intensity plot*

---

**Description**

Prepare data for a emission intensity plot

**Usage**

```
prep_emission_intensity(data, convert_label = identity, span_5yr = FALSE)
```

**Arguments**

data	A data frame. Requirements: <ul style="list-style-type: none"> <li>• The structure must be like <a href="#">sda_demo</a>.</li> <li>• The column sector must have a single value (e.g. "cement").</li> <li>• (Optional) If present, the column label is used for data labels.</li> </ul>
convert_label	A symbol. The unquoted name of a function to apply to y-axis labels. For example: <ul style="list-style-type: none"> <li>• To convert labels to uppercase use <code>convert_label = toupper</code>.</li> <li>• To get the default behavior of <code>qplot_emission_intensity</code> use <code>convert_label = to_title</code>.</li> </ul>
span_5yr	Logical. Use TRUE to restrict the time span to 5 years from the start year (the default behavior of <code>qplot_emission_intensity()</code> ), or use FALSE to impose no restriction.

**Value**

A data-frame ready to be plotted using `plot_emission_intensity()`.

**See Also**

Other plotting functions: [plot\\_emission\\_intensity\(\)](#), [plot\\_techmix\(\)](#), [plot\\_trajectory\(\)](#), [prep\\_techmix\(\)](#), [prep\\_trajectory\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_techmix\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_colour\\_r2dii\\_tech\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\\_tech\(\)](#), [spell\\_out\\_technology\(\)](#), [theme\\_2dii\(\)](#), [to\\_title\(\)](#)

**Examples**

```
# `data` must meet documented "Requirements"
data <- subset(sda_demo, sector == "cement" & region == "global")
prep_emission_intensity(data)
```

---

```
prep_techmix
```

---

*Prepare data for plotting technology mix*

---

**Description**

Prepare data for plotting technology mix

**Usage**

```
prep_techmix(
  data,
  convert_label = identity,
  span_5yr = FALSE,
  convert_tech_label = identity
)
```

**Arguments**

**data** A data frame. Requirements:

- The structure must be like [market\\_share\\_demo](#).
- The following columns must have a single value: sector, region, scenario\_source.
- The column `metric` must have a portfolio (e.g. "projected"), a benchmark (e.g. "corporate\_economy"), and a single scenario (e.g. "target\_sds").
- (Optional) If present, the column `label` is used for data labels.
- (Optional) If present, the column `label_tech` is used for technology labels.

**convert\_label** A symbol. The unquoted name of a function to apply to y-axis labels. For example:

- To convert labels to uppercase use `convert_label = toupper`.
- To get the default behavior of `qplot_techmix` use `convert_label = recode_metric_techmix`.

**span\_5yr** Logical. Use TRUE to restrict the time span to 5 years from the start year (the default behavior of `qplot_techmix()`), or use FALSE to impose no restriction.

**convert\_tech\_label** A symbol. The unquoted name of a function to apply to technology legend labels. For example, to convert labels to uppercase use `convert_tech_label = toupper`. To get the default behavior of `qplot_techmix()` use `convert_tech_label = spell_out_technology`.

**Value**

A data-frame ready to be plotted using `plot_techmix()`.

**See Also**

Other plotting functions: [plot\\_emission\\_intensity\(\)](#), [plot\\_techmix\(\)](#), [plot\\_trajectory\(\)](#), [prep\\_emission\\_intensity\(\)](#), [prep\\_trajectory\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_techmix\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_colour\\_r2dii\\_tech\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\\_tech\(\)](#), [spell\\_out\\_technology\(\)](#), [theme\\_2dii\(\)](#), [to\\_title\(\)](#)

**Examples**

```
# `data` must meet documented "Requirements"
data <- subset(
  market_share_demo,
  scenario_source == "demo_2020" &
```

```

    sector == "power" &
    region == "global" &
    metric %in% c("projected", "corporate_economy", "target_sds")
  )
prep_techmix(data)

```

---

```

prep_trajectory      Prepare data for a trajectory plot

```

---

### Description

Prepare data for a trajectory plot

### Usage

```

prep_trajectory(
  data,
  convert_label = identity,
  span_5yr = FALSE,
  value_col = "percentage_of_initial_production_by_scope"
)

```

### Arguments

data	A data frame. Requirements: <ul style="list-style-type: none"> <li>• The structure must be like <a href="#">market_share_demo</a>.</li> <li>• The following columns must have a single value: sector, technology, region, scenario_source.</li> <li>• (Optional) If present, the column label is used for data labels.</li> </ul>
convert_label	A symbol. The unquoted name of a function to apply to y-axis labels. For example: <ul style="list-style-type: none"> <li>• To convert labels to uppercase use <code>convert_label = toupper</code>.</li> <li>• To get the default behavior of <code>qplot_trajectory</code> use <code>convert_label = recode_metric_trajectory</code>.</li> </ul>
span_5yr	Logical. Use TRUE to restrict the time span to 5 years from the start year (the default behavior of <code>qplot_trajectory()</code> ), or use FALSE to impose no restriction.
value_col	Character. Name of the column to be used as a value to be plotted.

### Value

A data-frame ready to be plotted using `plot_trajectory()`.

**See Also**

Other plotting functions: [plot\\_emission\\_intensity\(\)](#), [plot\\_techmix\(\)](#), [plot\\_trajectory\(\)](#), [prep\\_emission\\_intensity\(\)](#), [prep\\_techmix\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_techmix\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_colour\\_r2dii\\_tech\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\\_tech\(\)](#), [spell\\_out\\_technology\(\)](#), [theme\\_2dii\(\)](#), [to\\_title\(\)](#)

**Examples**

```
# `data` must meet documented "Requirements"
data <- subset(
  market_share_demo,
  sector == "power" &
  technology == "renewablecap" &
  region == "global" &
  scenario_source == "demo_2020"
)

prep_trajectory(data)
```

---

prioritize

*Pick rows where score is 1 and level per loan is of highest priority*

---

**Description**

When multiple perfect matches are found per loan (e.g. a match at `direct_loantaker` level and `ultimate_parent` level), we must prioritize the desired match. By default, the highest priority is the most granular match (i.e. `direct_loantaker`).

**Usage**

```
prioritize(data, priority = NULL)
```

**Arguments**

<code>data</code>	A data frame like the validated output of <a href="#">match_name()</a> . See <i>Details</i> on how to validate data.
<code>priority</code>	One of: <ul style="list-style-type: none"> <li>• <code>NULL</code>: defaults to the default level priority as returned by <a href="#">prioritize_level()</a>.</li> <li>• A character vector giving a custom priority.</li> <li>• A function to apply to the output of <a href="#">prioritize_level()</a>, e.g. <code>rev</code>.</li> <li>• A quosure-style lambda function, e.g. <code>~ rev(.x)</code>.</li> </ul>

## Details

**How to validate** data Write the output of `match_name()` into a .csv file with:

```
# Writing to current working directory
matched %>%
  readr::write_csv("matched.csv")
```

Compare, edit, and save the data manually:

- Open *matched.csv* with any spreadsheet editor (Excel, Google Sheets, etc.).
- Compare the columns `name` and `name_abcd` manually to determine if the match is valid. Other information can be used in conjunction with just the names to ensure the two entities match (sector, internal information on the company structure, etc.)
- Edit the data:
  - If you are happy with the match, set the `score` value to 1.
  - Otherwise set or leave the `score` value to anything other than 1.
- Save the edited file as, say, *valid\_matches.csv*.

Re-read the edited file (validated) with:

```
# Reading from current working directory
valid_matches <- readr::read_csv("valid_matches.csv")
```

## Value

A data frame with a single row per loan, where `score` is 1 and `priority_level` is highest.

## Handling grouped data

This function ignores but preserves existing groups.

## See Also

Other matching functions: [crucial\\_lbk\(\)](#), [match\\_name\(\)](#), [prioritize\\_level\(\)](#)

## Examples

```
library(dplyr)

# styler: off
matched <- tribble(
  ~sector, ~sector_abcd, ~score, ~id_loan, ~level,
  "coal", "coal", 1, "aa", "ultimate_parent",
  "coal", "coal", 1, "aa", "direct_loantaker",
  "coal", "coal", 1, "bb", "intermediate_parent",
  "coal", "coal", 1, "bb", "ultimate_parent",
)
# styler: on
```

```

prioritize_level(matched)

# Using default priority
prioritize(matched)

# Using the reverse of the default priority
prioritize(matched, priority = rev)

# Same
prioritize(matched, priority = ~ rev(.x))

# Using a custom priority
bad_idea <- c("intermediate_parent", "ultimate_parent", "direct_loantaker")

prioritize(matched, priority = bad_idea)

```

---

prioritize_level	<i>Arrange unique level values in default order of priority</i>
------------------	---

---

### Description

Arrange unique level values in default order of priority

### Usage

```
prioritize_level(data)
```

### Arguments

data            A data frame, commonly the output of `match_name()`.

### Value

A character vector of the default level priority per loan.

### See Also

Other matching functions: `crucial_lbk()`, `match_name()`, `prioritize()`

### Examples

```

matched <- tibble::tibble(
  level = c(
    "intermediate_parent_1",
    "direct_loantaker",
    "direct_loantaker",
    "direct_loantaker",
    "ultimate_parent",
    "intermediate_parent_2"
  )
)

```

```
)  
prioritize_level(matched)
```

---

psic\_classification     *Dataset to bridge (translate) common sector-classification codes*

---

## Description

This dataset serves as a translation key between common sector-classification systems and sectors relevant to the 'PACTA' tool (<https://www.transitionmonitor.com/>).

## Usage

```
psic_classification
```

## Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1271 rows and 5 columns.

## Definitions

- `borderline` (logical): Flag indicating if PACTA sector and classification code are a borderline match. The value `TRUE` indicates that the match is uncertain between the PACTA sector and the classification. The value `FALSE` indicates that the match is certainly perfect or the classification is certainly out of PACTA's scope..
- `code` (character): Formatted PSIC classification code..
- `description` (character): Original PSIC classification sector name..
- `sector` (character): Associated PACTA sector..
- `version` (character): Column identifying which year the classification was published in..

## Details

Classification datasets help to standardize sector classification codes from the wild to a relevant subset including 'power', 'oil and gas', 'coal', 'automotive', 'aviation', 'concrete', 'steel', and 'shipping'.

## See Also

Other datasets: [gics\\_classification](#), [increasing\\_or\\_decreasing](#), [isic\\_classification](#), [iso\\_codes](#), [nace\\_classification](#), [naics\\_classification](#), [palette\\_colours](#), [region\\_isos](#), [scenario\\_colours](#), [sector\\_classifications](#), [sector\\_colours](#), [sic\\_classification](#), [technology\\_colours](#)

## Examples

```
head(psic_classification)
```

---

`qplot_emission_intensity`*Create a quick emission intensity plot*

---

## Description

Compared to `plot_emission_intensity()` this function:

- is restricted to plotting future as 5 years from the start year,
- outputs formatted labels, based on emission metric column,
- outputs a title,
- outputs formatted axis labels.

## Usage

```
qplot_emission_intensity(data)
```

## Arguments

`data` A data frame like the output of `prep_emission_intensity()`.

## Value

An object of class "ggplot".

## See Also

Other plotting functions: `plot_emission_intensity()`, `plot_techmix()`, `plot_trajectory()`, `prep_emission_intensity()`, `prep_techmix()`, `prep_trajectory()`, `qplot_techmix()`, `qplot_trajectory()`, `recode_metric_techmix()`, `recode_metric_trajectory()`, `scale_colour_r2dii()`, `scale_colour_r2dii_sector()`, `scale_colour_r2dii_tech()`, `scale_fill_r2dii()`, `scale_fill_r2dii_sector()`, `scale_fill_r2dii_tech()`, `spell_out_technology()`, `theme_2dii()`, `to_title()`

## Examples

```
# `data` must meet documented "Requirements"
data <- subset(sda_demo, sector == "cement" & region == "global")

qplot_emission_intensity(data)
```

---

qplot_techmix	<i>Create a quick techmix plot</i>
---------------	------------------------------------

---

### Description

Compared to [plot\\_techmix\(\)](#) this function:

- is restricted to plotting future as 5 years from the start year,
- outputs pretty bar labels, based on metric column,
- outputs pretty legend labels, based on technology column,
- outputs a title.

### Usage

```
qplot_techmix(data)
```

### Arguments

`data` A data frame like the output of [prep\\_techmix\(\)](#).

### Value

An object of class "ggplot".

### See Also

Other plotting functions: [plot\\_emission\\_intensity\(\)](#), [plot\\_techmix\(\)](#), [plot\\_trajectory\(\)](#), [prep\\_emission\\_intensity\(\)](#), [prep\\_techmix\(\)](#), [prep\\_trajectory\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_colour\\_r2dii\\_tech\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\\_tech\(\)](#), [spell\\_out\\_technology\(\)](#), [theme\\_2dii\(\)](#), [to\\_title\(\)](#)

### Examples

```
# `data` must meet documented "Requirements"
data <- subset(
  market_share_demo,
  sector == "power" &
  region == "global" &
  scenario_source == "demo_2020" &
  metric %in% c("projected", "corporate_economy", "target_sds")
)

qplot_techmix(data)
```

---

qplot\_trajectory      *Create a quick trajectory plot*

---

### Description

Compared to `plot_trajectory()` this function:

- is restricted to plotting only 5 years from the start year,
- outputs pretty legend labels, based on the column holding metrics,
- outputs a title,
- outputs a subtitle,
- outputs informative axis labels in sentence case.

### Usage

```
qplot_trajectory(data)
```

### Arguments

`data`                    A data frame like the outputs of `prep_trajectory()`.

- (Optional) If present, the column label is used for data labels.

### Value

An object of class "ggplot".

### See Also

Other plotting functions: `plot_emission_intensity()`, `plot_techmix()`, `plot_trajectory()`, `prep_emission_intensity()`, `prep_techmix()`, `prep_trajectory()`, `qplot_emission_intensity()`, `qplot_techmix()`, `recode_metric_techmix()`, `recode_metric_trajectory()`, `scale_colour_r2dii()`, `scale_colour_r2dii_sector()`, `scale_colour_r2dii_tech()`, `scale_fill_r2dii()`, `scale_fill_r2dii_sector()`, `scale_fill_r2dii_tech()`, `spell_out_technology()`, `theme_2dii()`, `to_title()`

### Examples

```
# `data` must meet documented "Requirements"
data <- subset(
  market_share_demo,
  sector == "power" &
  technology == "renewables" &
  region == "global" &
  scenario_source == "demo_2020"
)

qplot_trajectory(data)
```

---

recode\_metric\_techmix *Replicate labels produced with qplot\_\*() functions*

---

### Description

- `to_title()` converts labels like `qplot_emission_intensity()`.
- `recode_metric_trajectory()` converts labels like `qplot_trajectory()`.
- `recode_metric_techmix()` converts labels like `qplot_techmix()`.
- `spell_out_technology()` converts technology labels like `qplot_techmix()`.

### Usage

```
recode_metric_techmix(x)
```

### Arguments

x                    A character vector.

### Value

A character vector.

### See Also

Other plotting functions: `plot_emission_intensity()`, `plot_techmix()`, `plot_trajectory()`, `prep_emission_intensity()`, `prep_techmix()`, `prep_trajectory()`, `qplot_emission_intensity()`, `qplot_techmix()`, `qplot_trajectory()`, `recode_metric_trajectory()`, `scale_colour_r2dii()`, `scale_colour_r2dii_sector()`, `scale_colour_r2dii_tech()`, `scale_fill_r2dii()`, `scale_fill_r2dii_sector()`, `scale_fill_r2dii_tech()`, `spell_out_technology()`, `theme_2dii()`, `to_title()`

### Examples

```
to_title(c("a.string", "another_STRING"))

metric <- c("projected", "corporate_economy", "target_xyz", "else")
recode_metric_trajectory(metric)

recode_metric_techmix(metric)

spell_out_technology(c("gas", "ice", "coalcap", "hdv"))
```

---

recode\_metric\_trajectory

*Replicate labels produced with `qplot_*()` functions*

---

### Description

- `to_title()` converts labels like `qplot_emission_intensity()`.
- `recode_metric_trajectory()` converts labels like `qplot_trajectory()`.
- `recode_metric_techmix()` converts labels like `qplot_techmix()`.
- `spell_out_technology()` converts technology labels like `qplot_techmix()`.

### Usage

```
recode_metric_trajectory(x)
```

### Arguments

`x`                    A character vector.

### Value

A character vector.

### See Also

Other plotting functions: `plot_emission_intensity()`, `plot_techmix()`, `plot_trajectory()`, `prep_emission_intensity()`, `prep_techmix()`, `prep_trajectory()`, `qplot_emission_intensity()`, `qplot_techmix()`, `qplot_trajectory()`, `recode_metric_techmix()`, `scale_colour_r2dii()`, `scale_colour_r2dii_sector()`, `scale_colour_r2dii_tech()`, `scale_fill_r2dii()`, `scale_fill_r2dii_sector()`, `scale_fill_r2dii_tech()`, `spell_out_technology()`, `theme_2dii()`, `to_title()`

### Examples

```
to_title(c("a.string", "another_STRING"))

metric <- c("projected", "corporate_economy", "target_xyz", "else")
recode_metric_trajectory(metric)

recode_metric_techmix(metric)

spell_out_technology(c("gas", "ice", "coalcap", "hdv"))
```

---

`region_isos`*A dataset outlining various region definitions*

---

### Description

This dataset maps codes representing countries to regions.

For information about the ISO standard for country codes see <https://www.iso.org/iso-3166-country-codes.html>.

### Usage

```
region_isos
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 9262 rows and 3 columns.

### Definitions

- `isos` (character): Countries in region, defined by iso code., \* `region` (character): Benchmark region name., \* `source` (character): Source publication from which the regions are defined.

### See Also

Other datasets: [gics\\_classification](#), [increasing\\_or\\_decreasing](#), [isic\\_classification](#), [iso\\_codes](#), [nace\\_classification](#), [naics\\_classification](#), [palette\\_colours](#), [psic\\_classification](#), [scenario\\_colours](#), [sector\\_classifications](#), [sector\\_colours](#), [sic\\_classification](#), [technology\\_colours](#)

### Examples

```
head(region_isos)
```

---

`region_isos_demo`*A dataset outlining various region definitions*

---

### Description

This dataset maps codes representing countries to regions. It is similar to but smaller than [region\\_isos](#).

Demo datasets are synthetic because most financial data is strictly private; they help to demonstrate and test the implementation in R of 'PACTA' (<https://www.transitionmonitor.com/>).

For information about the ISO standard for country codes see <https://www.iso.org/iso-3166-country-codes.html>.

**Usage**

```
region_isos_demo
```

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 358 rows and 3 columns.

**Definitions**

- `isos` (character): Countries in region, defined by iso code., \* `region` (character): Benchmark region name., \* `source` (character): Source publication from which the regions are defined.

**See Also**

Other demo data: [abcd\\_demo](#), [co2\\_intensity\\_scenario\\_demo](#), [loanbook\\_demo](#), [market\\_share\\_demo](#), [overwrite\\_demo](#), [scenario\\_demo\\_2020](#), [sda\\_demo](#)

**Examples**

```
region_isos_demo
```

---

`scale_colour_r2dii`      *Custom PACTA colour and fill scales*

---

**Description**

A custom discrete colour and fill scales with colours from the PACTA palette.

**Usage**

```
scale_colour_r2dii(colour_labels = NULL, ...)
```

**Arguments**

`colour_labels` A character vector. Specifies colour labels to use and their order. Run `unique(r2dii.plot:::palette_c)` to see available colours. Similar to `value` parameter in `ggplot2::scale_colour_manual()`.

`...` Other parameters passed on to `ggplot2::discrete_scale()`.

**Value**

An object of class "ScaleDiscrete".

**See Also**

Other plotting functions: [plot\\_emission\\_intensity\(\)](#), [plot\\_techmix\(\)](#), [plot\\_trajectory\(\)](#), [prep\\_emission\\_intensity\(\)](#), [prep\\_techmix\(\)](#), [prep\\_trajectory\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_techmix\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_colour\\_r2dii\\_tech\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\\_tech\(\)](#), [spell\\_out\\_technology\(\)](#), [theme\\_2dii\(\)](#), [to\\_title\(\)](#)

**Examples**

```
library(ggplot2, warn.conflicts = FALSE)

ggplot(mpg) +
  geom_point(aes(displ, hwy, color = class)) +
  scale_colour_r2dii()

ggplot(mpg) +
  geom_histogram(aes(cyl, fill = class), position = "dodge", bins = 5) +
  scale_fill_r2dii()
```

---

scale\_colour\_r2dii\_sector

*Custom PACTA sector colour and fill scales*

---

**Description**

A custom discrete colour and fill scales with colours from the PACTA sector palette.

**Usage**

```
scale_colour_r2dii_sector(sectors = NULL, ...)
```

**Arguments**

sectors	A character vector. Specifies sector colours to use and their order. Run <code>unique(r2dii.plot:::sector_c)</code> to see available labels. Similar to value parameter in <code>ggplot2::scale_colour_manual()</code> .
...	Other parameters passed on to <code>ggplot2::discrete_scale()</code> .

**Value**

An object of class "ScaleDiscrete".

**See Also**

Other plotting functions: `plot_emission_intensity()`, `plot_techmix()`, `plot_trajectory()`, `prep_emission_intensity()`, `prep_techmix()`, `prep_trajectory()`, `qplot_emission_intensity()`, `qplot_techmix()`, `qplot_trajectory()`, `recode_metric_techmix()`, `recode_metric_trajectory()`, `scale_colour_r2dii()`, `scale_colour_r2dii_tech()`, `scale_fill_r2dii()`, `scale_fill_r2dii_sector()`, `scale_fill_r2dii_tech()`, `spell_out_technology()`, `theme_2dii()`, `to_title()`

**Examples**

```
library(ggplot2, warn.conflicts = FALSE)

ggplot(mpg) +
  geom_point(aes(displ, hwy, color = class)) +
  scale_colour_r2dii_sector()
```

```
ggplot(mpg) +
  geom_histogram(aes(cyl, fill = class), position = "dodge", bins = 5) +
  scale_fill_r2dii_sector()
```

---

scale\_colour\_r2dii\_tech

*Custom PACTA technology colour and fill scales*

---

## Description

A custom discrete colour and fill scales with colours from the PACTA technology palette.

## Usage

```
scale_colour_r2dii_tech(sector, technologies = NULL, ...)
```

## Arguments

sector	A string. Sector name specifying a colour palette. Run <code>unique(r2dii.plot::technology_colours\$sector)</code> to see available sectors.
technologies	A character vector. Specifies technologies to use as colours and their order. Run <code>unique(r2dii.plot::technology_colours\$technology)</code> to see available technologies (pay attention if they match the sector). Similar to value parameter in <code>ggplot2::scale_colour_manual()</code> .
...	Other parameters passed on to <code>ggplot2::discrete_scale()</code> .

## Value

An object of class "ScaleDiscrete".

## See Also

Other plotting functions: [plot\\_emission\\_intensity\(\)](#), [plot\\_techmix\(\)](#), [plot\\_trajectory\(\)](#), [prep\\_emission\\_intensity\(\)](#), [prep\\_techmix\(\)](#), [prep\\_trajectory\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_techmix\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\\_tech\(\)](#), [spell\\_out\\_technology\(\)](#), [theme\\_2dii\(\)](#), [to\\_title\(\)](#)

## Examples

```
library(ggplot2, warn.conflicts = FALSE)
```

```
ggplot(mpg) +
  geom_point(aes(displ, hwy, color = class)) +
  scale_colour_r2dii_tech("automotive")
```

```
ggplot(mpg) +
  geom_histogram(aes(cyl, fill = class), position = "dodge", bins = 5) +
  scale_fill_r2dii_tech("automotive")
```

---

scale_fill_r2dii	<i>Custom PACTA colour and fill scales</i>
------------------	--

---

## Description

A custom discrete colour and fill scales with colours from the PACTA palette.

## Usage

```
scale_fill_r2dii(colour_labels = NULL, ...)
```

## Arguments

`colour_labels` A character vector. Specifies colour labels to use and their order. Run `unique(r2dii.plot::palette_c)` to see available colours. Similar to `value` parameter in `ggplot2::scale_colour_manual()`.

`...` Other parameters passed on to `ggplot2::discrete_scale()`.

## Value

An object of class "ScaleDiscrete".

## See Also

Other plotting functions: `plot_emission_intensity()`, `plot_techmix()`, `plot_trajectory()`, `prep_emission_intensity()`, `prep_techmix()`, `prep_trajectory()`, `qplot_emission_intensity()`, `qplot_techmix()`, `qplot_trajectory()`, `recode_metric_techmix()`, `recode_metric_trajectory()`, `scale_colour_r2dii()`, `scale_colour_r2dii_sector()`, `scale_colour_r2dii_tech()`, `scale_fill_r2dii_sector()`, `scale_fill_r2dii_tech()`, `spell_out_technology()`, `theme_2dii()`, `to_title()`

## Examples

```
library(ggplot2, warn.conflicts = FALSE)

ggplot(mpg) +
  geom_point(aes(displ, hwy, color = class)) +
  scale_colour_r2dii()

ggplot(mpg) +
  geom_histogram(aes(cyl, fill = class), position = "dodge", bins = 5) +
  scale_fill_r2dii()
```

---

`scale_fill_r2dii_sector`*Custom PACTA sector colour and fill scales*

---

## Description

A custom discrete colour and fill scales with colours from the PACTA sector palette.

## Usage

```
scale_fill_r2dii_sector(sectors = NULL, ...)
```

## Arguments

<code>sectors</code>	A character vector. Specifies sector colours to use and their order. Run <code>unique(r2dii.plot:::sector_c)</code> to see available labels. Similar to <code>value</code> parameter in <code>ggplot2::scale_colour_manual()</code> .
<code>...</code>	Other parameters passed on to <code>ggplot2::discrete_scale()</code> .

## Value

An object of class "ScaleDiscrete".

## See Also

Other plotting functions: `plot_emission_intensity()`, `plot_techmix()`, `plot_trajectory()`, `prep_emission_intensity()`, `prep_techmix()`, `prep_trajectory()`, `qplot_emission_intensity()`, `qplot_techmix()`, `qplot_trajectory()`, `recode_metric_techmix()`, `recode_metric_trajectory()`, `scale_colour_r2dii()`, `scale_colour_r2dii_sector()`, `scale_colour_r2dii_tech()`, `scale_fill_r2dii()`, `scale_fill_r2dii_tech()`, `spell_out_technology()`, `theme_2dii()`, `to_title()`

## Examples

```
library(ggplot2, warn.conflicts = FALSE)

ggplot(mpg) +
  geom_point(aes(displ, hwy, color = class)) +
  scale_colour_r2dii_sector()

ggplot(mpg) +
  geom_histogram(aes(cyl, fill = class), position = "dodge", bins = 5) +
  scale_fill_r2dii_sector()
```

---

scale\_fill\_r2dii\_tech *Custom PACTA technology colour and fill scales*

---

### Description

A custom discrete colour and fill scales with colours from the PACTA technology palette.

### Usage

```
scale_fill_r2dii_tech(sector, technologies = NULL, ...)
```

### Arguments

sector	A string. Sector name specifying a colour palette. Run <code>unique(r2dii.plot::technology_colours\$sector)</code> to see available sectors.
technologies	A character vector. Specifies technologies to use as colours and their order. Run <code>unique(r2dii.plot::technology_colours\$technology)</code> to see available technologies (pay attention if they match the sector). Similar to value parameter in <code>ggplot2::scale_colour_manual()</code> .
...	Other parameters passed on to <code>ggplot2::discrete_scale()</code> .

### Value

An object of class "ScaleDiscrete".

### See Also

Other plotting functions: [plot\\_emission\\_intensity\(\)](#), [plot\\_techmix\(\)](#), [plot\\_trajectory\(\)](#), [prep\\_emission\\_intensity\(\)](#), [prep\\_techmix\(\)](#), [prep\\_trajectory\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_techmix\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_colour\\_r2dii\\_tech\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [spell\\_out\\_technology\(\)](#), [theme\\_2dii\(\)](#), [to\\_title\(\)](#)

### Examples

```
library(ggplot2, warn.conflicts = FALSE)

ggplot(mpg) +
  geom_point(aes(displ, hwy, color = class)) +
  scale_colour_r2dii_tech("automotive")

ggplot(mpg) +
  geom_histogram(aes(cyl, fill = class), position = "dodge", bins = 5) +
  scale_fill_r2dii_tech("automotive")
```

---

scenario_colours	<i>Colour datasets</i>
------------------	------------------------

---

## Description

All datasets have at least two columns:

- label: Text label of the colour.
- hex: Hex code of the colour.

## Usage

```
scenario_colours
```

## Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 5 rows and 2 columns.

## Details

In `scenario_colours`, colours are ordered from red to green to be used in trajectory charts.

## See Also

Other datasets: [gics\\_classification](#), [increasing\\_or\\_decreasing](#), [isic\\_classification](#), [iso\\_codes](#), [nace\\_classification](#), [naics\\_classification](#), [palette\\_colours](#), [psic\\_classification](#), [region\\_isos](#), [sector\\_classifications](#), [sector\\_colours](#), [sic\\_classification](#), [technology\\_colours](#)

## Examples

```
palette_colours
```

```
scenario_colours
```

```
sector_colours
```

```
technology_colours
```

---

scenario\_demo\_2020     *A prepared climate scenario dataset for demonstration*

---

## Description

Fake climate scenario dataset, prepared for the software PACTA (Paris Agreement Capital Transition Assessment). It imitates climate scenario data (e.g. from the International Energy Agency (IEA)) including the change through time in production across industrial sectors.

Demo datasets are synthetic because most financial data is strictly private; they help to demonstrate and test the implementation in R of 'PACTA' (<https://www.transitionmonitor.com/>).

## Usage

```
scenario_demo_2020
```

## Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1512 rows and 8 columns.

## Definitions

- `region` (character): The region to which the pathway is relevant., \* `scenario` (character): The name of the scenario., \* `scenario_source` (character): The source publication from which the scenario was taken., \* `sector` (character): The sector to which the scenario prescribes a pathway., \* `smsp` (double): Sector market share percentage of the pathway calculated in 2020., \* `technology` (character): The technology within the sector to which the scenario prescribes a pathway., \* `tmsr` (double): Technology market share ratio of the pathway calculated in 2020., \* `year` (integer): The year at which the pathway value is prescribed.

## See Also

Other demo data: [abcd\\_demo](#), [co2\\_intensity\\_scenario\\_demo](#), [loanbook\\_demo](#), [market\\_share\\_demo](#), [overwrite\\_demo](#), [region\\_isos\\_demo](#), [sda\\_demo](#)

## Examples

```
head(scenario_demo_2020)
```

---

`sda_demo`*An example of an sda\_demo-like dataset*

---

**Description**

Dataset imitating the output of `r2dii.analysis::target_sda()`.

**Usage**`sda_demo`**Format**

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 110 rows and 6 columns.

**Source**

<https://github.com/RMI-PACTA/r2dii.plot/issues/55>.

**See Also**

Other demo data: [abcd\\_demo](#), [co2\\_intensity\\_scenario\\_demo](#), [loanbook\\_demo](#), [market\\_share\\_demo](#), [overwrite\\_demo](#), [region\\_isos\\_demo](#), [scenario\\_demo\\_2020](#)

**Examples**`sda_demo`

---

`sector_classifications`*A view of available sector classification datasets*

---

**Description**

This dataset lists all sector classification code standards used by 'PACTA' (<https://www.transitionmonitor.com/>).

**Usage**`sector_classifications`**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 6559 rows and 4 columns.

**Definitions**

- `borderline` (character): Flag indicating if 2dii sector and classification code are a borderline match. The value `TRUE` indicates that the match is uncertain between the 2dii sector and the classification. The value `FALSE` indicates that the match is certainly perfect or the classification is certainly out of 2dii's scope., \* `code` (character): Formatted code., \* `code_system` (character): Code system., \* `sector` (character): Associated 2dii sector.

**Details**

Classification datasets help to standardize sector classification codes from the wild to a relevant subset including 'power', 'oil and gas', 'coal', 'automotive', 'aviation', 'concrete', 'steel', and 'shipping'.

**See Also**

Other datasets: [gics\\_classification](#), [increasing\\_or\\_decreasing](#), [isic\\_classification](#), [iso\\_codes](#), [nace\\_classification](#), [naics\\_classification](#), [palette\\_colours](#), [psic\\_classification](#), [region\\_isos](#), [scenario\\_colours](#), [sector\\_colours](#), [sic\\_classification](#), [technology\\_colours](#)

**Examples**

```
head(sector_classifications)
```

---

sector_colours	<i>Colour datasets</i>
----------------	------------------------

---

**Description**

All datasets have at least two columns:

- `label`: Text label of the colour.
- `hex`: Hex code of the colour.

**Usage**

```
sector_colours
```

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 8 rows and 2 columns.

**Details**

In `scenario_colours`, colours are ordered from red to green to be used in trajectory charts.

**See Also**

Other datasets: [gics\\_classification](#), [increasing\\_or\\_decreasing](#), [isic\\_classification](#), [iso\\_codes](#), [nace\\_classification](#), [naics\\_classification](#), [palette\\_colours](#), [psic\\_classification](#), [region\\_isos](#), [scenario\\_colours](#), [sector\\_classifications](#), [sic\\_classification](#), [technology\\_colours](#)

**Examples**

```
palette_colours
```

```
scenario_colours
```

```
sector_colours
```

```
technology_colours
```

---

sic_classification	<i>Dataset to bridge (translate) common sector-classification codes</i>
--------------------	---

---

**Description**

This dataset serves as a translation key between common sector-classification systems and sectors relevant to the 'PACTA' tool (<https://www.transitionmonitor.com/>).

**Usage**

```
sic_classification
```

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1005 rows and 5 columns.

**Definitions**

- `borderline` (character): Flag indicating if PACTA sector and classification code are a borderline match. The value `TRUE` indicates that the match is uncertain between the PACTA sector and the classification. The value `FALSE` indicates that the match is certainly perfect or the classification is certainly out of PACTA's scope., \* `code` (character): Original SIC code., \* `description` (character): Original SIC description., \* `sector` (character): Associated PACTA sector., \* `version` (character): Column identifying to which SIC version the code belongs.

**Details**

Classification datasets help to standardize sector classification codes from the wild to a relevant subset including 'power', 'oil and gas', 'coal', 'automotive', 'aviation', 'concrete', 'steel', and 'shipping'.

**See Also**

Other datasets: [gics\\_classification](#), [increasing\\_or\\_decreasing](#), [isic\\_classification](#), [iso\\_codes](#), [nace\\_classification](#), [naics\\_classification](#), [palette\\_colours](#), [psic\\_classification](#), [region\\_isos](#), [scenario\\_colours](#), [sector\\_classifications](#), [sector\\_colours](#), [technology\\_colours](#)

**Examples**

```
head(sic_classification)
```

---

spell\_out\_technology *Replicate labels produced with qplot\_\*() functions*

---

**Description**

- `to_title()` converts labels like `qplot_emission_intensity()`.
- `recode_metric_trajectory()` converts labels like `qplot_trajectory()`.
- `recode_metric_techmix()` converts labels like `qplot_techmix()`.
- `spell_out_technology()` converts technology labels like `qplot_techmix()`.

**Usage**

```
spell_out_technology(x)
```

**Arguments**

x                    A character vector.

**Value**

A character vector.

**See Also**

Other plotting functions: [plot\\_emission\\_intensity\(\)](#), [plot\\_techmix\(\)](#), [plot\\_trajectory\(\)](#), [prep\\_emission\\_intensity\(\)](#), [prep\\_techmix\(\)](#), [prep\\_trajectory\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_techmix\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_colour\\_r2dii\\_tech\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\\_tech\(\)](#), [theme\\_2dii\(\)](#), [to\\_title\(\)](#)

**Examples**

```
to_title(c("a.string", "another_STRING"))

metric <- c("projected", "corporate_economy", "target_xyz", "else")
recode_metric_trajectory(metric)

recode_metric_techmix(metric)

spell_out_technology(c("gas", "ice", "coalcap", "hdv"))
```

---

target\_market\_share    *Add targets for production, using the market share approach*

---

### Description

This function calculates the portfolio-level production targets, as calculated using the market share approach applied to each relevant climate production forecast.

### Usage

```
target_market_share(
  data,
  abcd,
  scenario,
  region_isos = r2dii.data::region_isos,
  use_credit_limit = FALSE,
  by_company = FALSE,
  weight_production = TRUE,
  increasing_or_decreasing = r2dii.data::increasing_or_decreasing
)
```

### Arguments

data	A "data.frame" like the output of <code>r2dii.match::prioritize</code> .
abcd	An asset level data frame like <code>r2dii.data::abcd_demo</code> .
scenario	A scenario data frame like <code>r2dii.data::scenario_demo_2020</code> .
region_isos	A data frame like <code>r2dii.data::region_isos</code> (default).
use_credit_limit	Logical vector of length 1. FALSE defaults to using the column <code>loan_size_outstanding</code> . Set to TRUE to use the column <code>loan_size_credit_limit</code> instead.
by_company	Logical vector of length 1. FALSE defaults to outputting <code>production_value</code> at the portfolio-level. Set to TRUE to output <code>production_value</code> at the company-level.
weight_production	Logical vector of length 1. TRUE defaults to outputting production, weighted by relative loan-size. Set to FALSE to output the unweighted production values.
increasing_or_decreasing	A data frame like <code>r2dii.data::increasing_or_decreasing</code> .

### Value

A tibble including the summarized columns `metric`, `production`, `technology_share`, `percentage_of_initial_production` and `scope`. If `by_company = TRUE`, the output will also have the column `name_abcd`.

### Handling grouped data

This function ignores existing groups and outputs ungrouped data.

**See Also**

Other analysis functions: [target\\_sda\(\)](#)

**Examples**

```
library(r2dii.data)
library(r2dii.match)

loanbook <- head(loanbook_demo, 100)
abcd <- head(abcd_demo, 100)

matched <- loanbook %>%
  match_name(abcd) %>%
  prioritize()

# Calculate targets at portfolio level
matched %>%
  target_market_share(
    abcd = abcd,
    scenario = scenario_demo_2020,
    region_isos = region_isos_demo
  )

# Calculate targets at company level
matched %>%
  target_market_share(
    abcd = abcd,
    scenario = scenario_demo_2020,
    region_isos = region_isos_demo,
    by_company = TRUE
  )

matched %>%
  target_market_share(
    abcd = abcd,
    scenario = scenario_demo_2020,
    region_isos = region_isos_demo,
    # Calculate unweighted targets
    weight_production = FALSE
  )
```

---

target\_sda

*Add targets for CO<sub>2</sub> emissions per unit production at the portfolio level, using the SDA approach*

---

## Description

This function calculates targets of  $CO_2$  emissions per unit production at the portfolio-level, otherwise referred to as "emissions factors". It uses the [sectoral-decarbonization approach \(SDA\)](#) to calculate these targets.

## Usage

```
target_sda(  
  data,  
  abcd,  
  co2_intensity_scenario,  
  use_credit_limit = FALSE,  
  by_company = FALSE,  
  region_isos = r2dii.data::region_isos  
)
```

## Arguments

data	A dataframe like the output of <code>r2dii.match::prioritize()</code> .
abcd	An asset-level data frame like <code>r2dii.data::abcd_demo</code> .
co2_intensity_scenario	A scenario data frame like <code>r2dii.data::co2_intensity_scenario_demo</code> .
use_credit_limit	Logical vector of length 1. FALSE defaults to using the column <code>loan_size_outstanding</code> . Set to TRUE to instead use the column <code>loan_size_credit_limit</code> .
by_company	Logical vector of length 1. FALSE defaults to outputting <code>weighted_production_value</code> at the portfolio-level. Set to TRUE to output <code>weighted_production_value</code> at the company-level.
region_isos	A data frame like <code>r2dii.data::region_isos</code> (default).

## Value

A tibble including the summarized columns `emission_factor_metric` and `emission_factor_value`. If `by_company = TRUE`, the output will also have the column `name_abcd`.

## Handling grouped data

This function ignores existing groups and outputs ungrouped data.

## See Also

Other analysis functions: [target\\_market\\_share\(\)](#)

## Examples

```
library(r2dii.match)  
library(r2dii.data)
```

```
loanbook <- head(loanbook_demo, 150)
abcd <- head(abcd_demo, 100)

matched <- loanbook %>%
  match_name(abcd) %>%
  prioritize()

# Calculate targets at portfolio level
matched %>%
  target_sda(
    abcd = abcd,
    co2_intensity_scenario = co2_intensity_scenario_demo,
    region_isos = region_isos_demo
  )

# Calculate targets at company level
matched %>%
  target_sda(
    abcd = abcd,
    co2_intensity_scenario = co2_intensity_scenario_demo,
    region_isos = region_isos_demo,
    by_company = TRUE
  )
```

---

technology\_colours      *Colour datasets*

---

## Description

All datasets have at least two columns:

- label: Text label of the colour.
- hex: Hex code of the colour.

## Usage

```
technology_colours
```

## Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 18 rows and 3 columns.

## Details

In `scenario_colours`, colours are ordered from red to green to be used in trajectory charts.

**See Also**

Other datasets: [gics\\_classification](#), [increasing\\_or\\_decreasing](#), [isic\\_classification](#), [iso\\_codes](#), [nace\\_classification](#), [naics\\_classification](#), [palette\\_colours](#), [psic\\_classification](#), [region\\_isos](#), [scenario\\_colours](#), [sector\\_classifications](#), [sector\\_colours](#), [sic\\_classification](#)

**Examples**

```
palette_colours
scenario_colours
sector_colours
technology_colours
```

---

theme\_2dii

*Complete theme*

---

**Description**

A ggplot theme which can be applied to all graphs to appear according to the PACTA plotting aesthetics.

**Usage**

```
theme_2dii(
  base_size = 12,
  base_family = "Helvetica",
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)
```

**Arguments**

base\_size      base font size, given in pts.  
base\_family    base font family  
base\_line\_size base size for line elements  
base\_rect\_size base size for rect elements

**Value**

An object of class "theme", "gg".

**See Also**

Other plotting functions: [plot\\_emission\\_intensity\(\)](#), [plot\\_techmix\(\)](#), [plot\\_trajectory\(\)](#), [prep\\_emission\\_intensity\(\)](#), [prep\\_techmix\(\)](#), [prep\\_trajectory\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_techmix\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_colour\\_r2dii\\_tech\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\\_tech\(\)](#), [spell\\_out\\_technology\(\)](#), [to\\_title\(\)](#)

**Examples**

```
library(ggplot2, warn.conflicts = FALSE)

ggplot(mtcars) +
  geom_histogram(aes(mpg), bins = 10) +
  theme_2dii()
```

---

to\_title

*Replicate labels produced with qplot\_\*() functions*

---

**Description**

- [to\\_title\(\)](#) converts labels like [qplot\\_emission\\_intensity\(\)](#).
- [recode\\_metric\\_trajectory\(\)](#) converts labels like [qplot\\_trajectory\(\)](#).
- [recode\\_metric\\_techmix\(\)](#) converts labels like [qplot\\_techmix\(\)](#).
- [spell\\_out\\_technology\(\)](#) converts technology labels like [qplot\\_techmix\(\)](#).

**Usage**

```
to_title(x)
```

**Arguments**

x                    A character vector.

**Value**

A character vector.

**See Also**

Other plotting functions: [plot\\_emission\\_intensity\(\)](#), [plot\\_techmix\(\)](#), [plot\\_trajectory\(\)](#), [prep\\_emission\\_intensity\(\)](#), [prep\\_techmix\(\)](#), [prep\\_trajectory\(\)](#), [qplot\\_emission\\_intensity\(\)](#), [qplot\\_techmix\(\)](#), [qplot\\_trajectory\(\)](#), [recode\\_metric\\_techmix\(\)](#), [recode\\_metric\\_trajectory\(\)](#), [scale\\_colour\\_r2dii\(\)](#), [scale\\_colour\\_r2dii\\_sector\(\)](#), [scale\\_colour\\_r2dii\\_tech\(\)](#), [scale\\_fill\\_r2dii\(\)](#), [scale\\_fill\\_r2dii\\_sector\(\)](#), [scale\\_fill\\_r2dii\\_tech\(\)](#), [spell\\_out\\_technology\(\)](#), [theme\\_2dii\(\)](#)

**Examples**

```
to_title(c("a.string", "another_STRING"))

metric <- c("projected", "corporate_economy", "target_xyz", "else")
recode_metric_trajectory(metric)

recode_metric_techmix(metric)

spell_out_technology(c("gas", "ice", "coalcap", "hdv"))
```

# Index

- \* **analysis functions**
    - target\_market\_share, 50
    - target\_sda, 51
  - \* **data dictionary**
    - data\_dictionary, 6
  - \* **datasets**
    - abcd\_demo, 3
    - co2\_intensity\_scenario\_demo, 4
    - data\_dictionary, 6
    - gics\_classification, 6
    - increasing\_or\_decreasing, 7
    - isic\_classification, 8
    - iso\_codes, 9
    - loanbook\_demo, 9
    - market\_share\_demo, 10
    - nace\_classification, 14
    - naics\_classification, 15
    - overwrite\_demo, 16
    - palette\_colours, 21
    - psic\_classification, 31
    - region\_isos, 37
    - region\_isos\_demo, 37
    - scenario\_colours, 44
    - scenario\_demo\_2020, 45
    - sda\_demo, 46
    - sector\_classifications, 46
    - sector\_colours, 47
    - sic\_classification, 48
    - technology\_colours, 53
  - \* **demo data**
    - abcd\_demo, 3
    - co2\_intensity\_scenario\_demo, 4
    - loanbook\_demo, 9
    - market\_share\_demo, 10
    - overwrite\_demo, 16
    - region\_isos\_demo, 37
    - scenario\_demo\_2020, 45
    - sda\_demo, 46
  - \* **matching functions**
    - crucial\_lbk, 5
    - match\_name, 11
    - prioritize, 28
    - prioritize\_level, 30
  - \* **plotting functions**
    - plot\_emission\_intensity, 22
    - plot\_techmix, 22
    - plot\_trajectory, 23
    - prep\_emission\_intensity, 24
    - prep\_techmix, 25
    - prep\_trajectory, 27
    - qplot\_emission\_intensity, 32
    - qplot\_techmix, 33
    - qplot\_trajectory, 34
    - recode\_metric\_techmix, 35
    - recode\_metric\_trajectory, 36
    - scale\_colour\_r2dii, 38
    - scale\_colour\_r2dii\_sector, 39
    - scale\_colour\_r2dii\_tech, 40
    - scale\_fill\_r2dii, 41
    - scale\_fill\_r2dii\_sector, 42
    - scale\_fill\_r2dii\_tech, 43
    - spell\_out\_technology, 49
    - theme\_2dii, 54
    - to\_title, 55
  - \* **utility functions**
    - pacta\_loanbook\_conflicts, 16
    - pacta\_loanbook\_deps, 17
    - pacta\_loanbook\_logo, 18
    - pacta\_loanbook\_packages, 19
    - pacta\_loanbook\_sitrep, 19
    - pacta\_loanbook\_update, 20
- abcd\_demo, 3, 5, 10, 11, 16, 38, 45, 46
- cli::ansi\_strip(), 18
- co2\_intensity\_scenario\_demo, 4, 4, 10, 11, 16, 38, 45, 46
- crucial\_lbk, 5, 13, 29, 30

- data\_dictionary, 6
- dplyr::inner\_join(), 12
- ggplot2::discrete\_scale(), 38–43
- ggplot2::scale\_colour\_manual(), 38–43
- gics\_classification, 6, 8, 9, 14, 15, 21, 31, 37, 44, 47–49, 54
- increasing\_or\_decreasing, 7, 7, 8, 9, 14, 15, 21, 31, 37, 44, 47–49, 54
- isic\_classification, 7, 8, 8, 9, 14, 15, 21, 31, 37, 44, 47–49, 54
- iso\_codes, 7, 8, 9, 14, 15, 21, 31, 37, 44, 47–49, 54
- loanbook\_demo, 4, 5, 9, 11, 16, 38, 45, 46
- market\_share\_demo, 4, 5, 10, 10, 16, 26, 27, 38, 45, 46
- match\_name, 5, 11, 29, 30
- match\_name(), 28, 30
- nace\_classification, 7–9, 14, 15, 21, 31, 37, 44, 47–49, 54
- naics\_classification, 7–9, 14, 15, 21, 31, 37, 44, 47–49, 54
- overwrite\_demo, 4, 5, 10, 11, 16, 38, 45, 46
- pacta\_loanbook\_conflicts, 16, 18–20
- pacta\_loanbook\_deps, 17, 17, 18–20
- pacta\_loanbook\_logo, 17, 18, 18, 19, 20
- pacta\_loanbook\_packages, 17, 18, 19, 20
- pacta\_loanbook\_sitrep, 17–19, 19, 20
- pacta\_loanbook\_update, 17–20, 20
- palette\_colours, 7–9, 14, 15, 21, 31, 37, 44, 47–49, 54
- plot\_emission\_intensity, 22, 23–26, 28, 32–36, 38–43, 49, 55
- plot\_emission\_intensity(), 32
- plot\_techmix, 22, 22, 24–26, 28, 32–36, 38–43, 49, 55
- plot\_techmix(), 33
- plot\_trajectory, 22, 23, 23, 25, 26, 28, 32–36, 38–43, 49, 55
- plot\_trajectory(), 34
- prep\_emission\_intensity, 22–24, 24, 26, 28, 32–36, 38–43, 49, 55
- prep\_techmix, 22–25, 25, 28, 32–36, 38–43, 49, 55
- prep\_trajectory, 22–26, 27, 32–36, 38–43, 49, 55
- prioritize, 5, 13, 28, 30
- prioritize\_level, 5, 13, 29, 30
- prioritize\_level(), 28
- psic\_classification, 7–9, 14, 15, 21, 31, 37, 44, 47–49, 54
- qplot\_emission\_intensity, 22–26, 28, 32, 33–36, 38–43, 49, 55
- qplot\_emission\_intensity(), 35, 36, 49, 55
- qplot\_techmix, 22–26, 28, 32, 33, 34–36, 38–43, 49, 55
- qplot\_techmix(), 35, 36, 49, 55
- qplot\_trajectory, 22–26, 28, 32, 33, 34, 35, 36, 38–43, 49, 55
- qplot\_trajectory(), 35, 36, 49, 55
- r2dii.analysis::target\_market\_share(), 10
- r2dii.analysis::target\_sda(), 46
- r2dii.data::abcd\_demo, 11, 50, 52
- r2dii.data::co2\_intensity\_scenario\_demo, 52
- r2dii.data::increasing\_or\_decreasing, 50
- r2dii.data::loanbook\_demo, 11
- r2dii.data::region\_isos, 50, 52
- r2dii.data::scenario\_demo\_2020, 50
- recode\_metric\_techmix, 22–26, 28, 32–34, 35, 36, 38–43, 49, 55
- recode\_metric\_trajectory, 22–26, 28, 32–35, 36, 38–43, 49, 55
- region\_isos, 7–9, 14, 15, 21, 31, 37, 37, 44, 47–49, 54
- region\_isos\_demo, 4, 5, 10, 11, 16, 37, 45, 46
- scale\_colour\_r2dii, 22–26, 28, 32–36, 38, 39–43, 49, 55
- scale\_colour\_r2dii\_sector, 22–26, 28, 32–36, 38, 39, 40–43, 49, 55
- scale\_colour\_r2dii\_tech, 22–26, 28, 32–36, 38, 39, 40, 41–43, 49, 55
- scale\_fill\_r2dii, 22–26, 28, 32–36, 38–40, 41, 42, 43, 49, 55
- scale\_fill\_r2dii\_sector, 22–26, 28, 32–36, 38–41, 42, 43, 49, 55

scale\_fill\_r2dii\_tech, 22–26, 28, 32–36,  
38–42, 43, 49, 55  
scenario\_colours, 7–9, 14, 15, 21, 31, 37,  
44, 47–49, 54  
scenario\_demo\_2020, 4, 5, 10, 11, 16, 38, 45,  
46  
sda\_demo, 4, 5, 10, 11, 16, 25, 38, 45, 46  
sector\_classifications, 7–9, 14, 15, 21,  
31, 37, 44, 46, 48, 49, 54  
sector\_colours, 7–9, 14, 15, 21, 31, 37, 44,  
47, 47, 49, 54  
sic\_classification, 7–9, 14, 15, 21, 31, 37,  
44, 47, 48, 48, 54  
spell\_out\_technology, 22–26, 28, 32–36,  
38–43, 49, 55  
stringdist::stringdist-metrics, 11  
stringdist::stringsim(), 11, 12  
  
target\_market\_share, 50, 52  
target\_sda, 51, 51  
technology\_colours, 7–9, 14, 15, 21, 31, 37,  
44, 47–49, 53  
theme\_2dii, 22–26, 28, 32–36, 38–43, 49, 54,  
55  
to\_title, 22–26, 28, 32–36, 38–43, 49, 55, 55