

Package ‘MineSDG’

August 7, 2026

Title Mining Industry SDG Impact Calculator

Version 0.4.0

Description Provides tools to calculate quantitative scores for the United Nations Sustainable Development Goals (SDGs) for the mining, minerals and metals sector. Retrieves official indicator data from the 'United Nations SDG API', runs trend, stability, benchmarking and convergence diagnostics, maps indicators to mining-sector materiality domains via a bundled ontology, computes site-level Key Performance Indicators (KPIs) aligned with Global Reporting Initiative (GRI) 11, International Council on Mining and Metals (ICMM) Mining Principles and Sustainability Accounting Standards Board (SASB) EM-MM conventions, scores sites on a 0-100 SDG scorecard, and ships an interactive 'shiny' dashboard with demonstration datasets. An Environmental, Social and Governance (ESG) reporting layer generates Global Reporting Initiative (GRI), International Council on Mining and Metals (ICMM) and Business Responsibility and Sustainability Reporting (BRSR) reports from a disclosure bundle interface, with framework mappings shipped as data and rendering to 'HTML', 'PDF', 'Word' and 'Excel' via 'Quarto' and 'openxlsx2'. Official Sustainable Development Goals information and indicator methodology are available from the United Nations Sustainable Development Goals website <<https://sdgs.un.org/goals>>.

URL <https://imanojkumar.github.io/MineSDG/>

BugReports <https://github.com/imanojkumar/MineSDG/issues/>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports httr2 (>= 1.0.0), data.table (>= 1.15.0), dplyr, ggplot2, stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown, shiny, DT, quarto, openxlsx2

VignetteBuilder knitr

Config/testthat/edition 3

SystemRequirements Quarto CLI (≥ 1.4) (optional, for report rendering)

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Manoj Kumar [aut, cre]

Maintainer Manoj Kumar <ekumarmanoj@gmail.com>

Repository CRAN

Date/Publication 2026-08-07 17:20:21 UTC

Contents

analyze_sdg_convergence	3
analyze_sdg_trend	4
as_disclosure_bundle	4
benchmark_sdg_performance	5
brsr_crosswalk	6
bundle_kpi	6
bundle_raw	7
bundle_score	7
calculate_community_investment	8
calculate_energy_intensity	9
calculate_ghg_intensity	9
calculate_land_restoration	10
calculate_safety_performance	11
calculate_waste_intensity	12
calculate_water_efficiency	13
calculate_workforce_diversity	14
compile_site_disclosures	15
compute_sdg_stability	16
demo_mine_sites	16
demo_sdg_country	17
explore_sdg_ontology	17
fetch_sdg_country_data	18
framework_crosswalk	19
generate_brsr_kpis	20
generate_brsr_report	20
generate_brsr_sectionA	21
generate_brsr_sectionB	22
generate_gri_report	22
generate_icmm_report	23
generate_mining_esg_index	24
generate_mining_risk_profile	25
generate_sdg_executive_summary	26
get_sdg_metadata	26
gri_content_index	27
gri_crosswalk	27

icmm_crosswalk	28
icmm_default_params	28
icmm_recommendation_rules	29
list_mining_kpis	29
list_sdg_indicators	30
map_bundle_to_framework	31
map_sdg_to_mining_domain	31
mining_kpi_registry	32
plot_domain_radar	33
plot_mining_esg_index	34
plot_sdg_benchmark	34
plot_sdg_convergence	35
plot_sdg_radar	35
plot_sdg_trend	36
plot_sdg_volatility	36
render_minesdg_report	37
run_minesdg_dashboard	38
score_portfolio_sdg	38
score_site_sdg	39
sdg_mining_ontology	40
summarize_sdg_data	41
validate_sdg_indicator	42
write_report_xlsx	42
Index	44

analyze_sdg_convergence

Analyze SDG Convergence

Description

Tests beta-convergence across countries for an SDG indicator.

Usage

```
analyze_sdg_convergence(data, base_year = NULL, final_year = NULL)
```

Arguments

data	A data.table returned by fetch_sdg_country_data().
base_year	Optional base year. If NULL, earliest year is used.
final_year	Optional final year. If NULL, latest year is used.

Value

A data.table with convergence statistics per indicator.

analyze_sdg_trend	<i>Analyze SDG Indicator Trend</i>
-------------------	------------------------------------

Description

Computes trend metrics for SDG indicator data including absolute change, percentage change, CAGR, and linear trend slope.

Usage

```
analyze_sdg_trend(data, group_by = c("indicator", "country"), min_points = 3)
```

Arguments

data	A data.table returned by fetch_sdg_country_data().
group_by	Character vector of grouping columns. Default c("indicator", "country").
min_points	Minimum number of time points required to compute trend. Default 3.

Details

Supports grouping by indicator, country, or combinations.

Value

A data.table containing trend statistics.

Examples

```
dt <- fetch_sdg_country_data(indicator = "15.3.1", country = "IND")
analyze_sdg_trend(dt)
```

as_disclosure_bundle	<i>Coerce an object to a disclosure bundle</i>
----------------------	--

Description

Generic used by the report generators so that users may pass either a prepared [compile_site_disclosures](#) bundle or a raw site data.frame.

Usage

```
as_disclosure_bundle(x, ...)
```

Arguments

`x` A minesdg_disclosure_bundle or a site data.frame.
`...` Passed on to [compile_site_disclosures](#) for data.frame input.

Value

A minesdg_disclosure_bundle.

Examples

```
b <- as_disclosure_bundle(demo_mine_sites, site_id = "AU-KALG",
                           years = 2023:2024)
class(b)
```

benchmark_sdg_performance

Benchmark SDG Performance

Description

Compares SDG indicator performance against a benchmark (global mean or user-supplied dataset).

Usage

```
benchmark_sdg_performance(
  data,
  benchmark_data = NULL,
  year = NULL,
  higher_is_better = TRUE
)
```

Arguments

`data` A data.table returned by `fetch_sdg_country_data()`.
`benchmark_data` Optional data.table with columns: indicator, year, value. If NULL, benchmark is computed as mean of provided data.
`year` Optional numeric year. If NULL, most recent year is used.
`higher_is_better` Logical. TRUE if higher values indicate better performance. Default TRUE.

Value

A data.table with benchmark comparison statistics.

Examples

```
dt <- fetch_sdg_country_data(goal = 15)
benchmark_sdg_performance(dt)
```

brsr_crosswalk	<i>BRSR disclosure crosswalk</i>
----------------	----------------------------------

Description

Mapping of SEBI Business Responsibility and Sustainability Report Sections A, B and C to MineSDG outputs, including INR-lakh currency transforms for monetary lines.

Usage

```
brsr_crosswalk
```

Format

See [gri_crosswalk](#) for the common schema.

Source

Curated in `data-raw/make_crosswalks.R`; paraphrases referencing the SEBI BRSR format.

bundle_kpi	<i>Read a KPI value from a disclosure bundle</i>
------------	--

Description

Stable accessor used by framework mappers and report templates.

Usage

```
bundle_kpi(bundle, kpi_id, year = NULL)
```

Arguments

- `bundle` A `minesdg_disclosure_bundle`.
- `kpi_id` A single KPI id from [mining_kpi_registry](#).
- `year` Optional year; defaults to the bundle's reporting year.

Value

A single numeric value, or NA if the KPI is unavailable.

Examples

```
b <- compile_site_disclosures(demo_mine_sites, site_id = "CU-ATAC",
                              years = 2024)
bundle_kpi(b, "trifr")
```

bundle_raw	<i>Read a raw input field from a disclosure bundle</i>
------------	--

Description

Raw fields are provenance-tagged copies of validated user input; they are never recomputed by the reporting layer.

Usage

```
bundle_raw(bundle, field, year = NULL)
```

Arguments

- bundle A minesdg_disclosure_bundle.
- field A single raw field name (a numeric column of the input site data, e.g. "energy_gj").
- year Optional year; defaults to the bundle's reporting year.

Value

A single numeric value, or NA if the field is unavailable.

Examples

```
b <- compile_site_disclosures(demo_mine_sites, site_id = "CU-ATAC",
                              years = 2024)
bundle_raw(b, "ghg_scope1_t")
```

bundle_score	<i>Read composite or goal-level scores from a disclosure bundle</i>
--------------	---

Description

Stable accessor for scorecard aggregates held in a disclosure bundle.

Usage

```
bundle_score(bundle, what = "composite", year = NULL)
```

Arguments

bundle	A minesdg_disclosure_bundle.
what	"composite" (default), "grade", or "goal_<n>" (e.g. "goal_8").
year	Optional year; defaults to the bundle's reporting year.

Value

A numeric score (or character grade), or NA.

Examples

```
b <- compile_site_disclosures(demo_mine_sites, site_id = "CU-ATAC",
                              years = 2024)
bundle_score(b)
bundle_score(b, what = "goal_8")
```

calculate_community_investment
<i>Calculate Community Investment Ratio (SDG 1 / SDG 17)</i>

Description

Computes community/social investment as a percentage of revenue (GRI 203-1; ICMM Principle 9), a widely benchmarked measure of shared-value contribution in the mining sector.

Usage

```
calculate_community_investment(community_investment, revenue)
```

Arguments

community_investment	Numeric. Community investment in a currency unit (e.g. million USD).
revenue	Numeric. Revenue in the same currency unit. Must be > 0.

Value

A named list with metric and community_investment_pct.

Examples

```
calculate_community_investment(24.0, 3349)
```

 calculate_energy_intensity

Calculate Energy Intensity and Renewable Share (SDG 7)

Description

Computes energy intensity per tonne of ore processed (GRI 302-3) and, optionally, the renewable share of total energy consumption (GRI 302-1 / SASB EM-MM-130a.1).

Usage

```
calculate_energy_intensity(energy_gj, ore_processed_kt, renewable_gj = NULL)
```

Arguments

energy_gj Numeric. Total energy consumed in gigajoules.
 ore_processed_kt Numeric. Ore processed in kilotonnes. Must be > 0.
 renewable_gj Numeric. Optional. Renewable energy consumed in GJ.

Value

A named list with metric, energy_intensity (GJ/t ore) and, when renewable_gj is supplied, renewable_share_percent.

Examples

```
calculate_energy_intensity(10449110, 18481, renewable_gj = 550000)
```

 calculate_ghg_intensity

Calculate GHG Emissions Intensity (SDG 13)

Description

Computes Scope 1 + Scope 2 greenhouse-gas emissions intensity per tonne of ore processed, following the GHG Protocol Corporate Standard and GRI 305-4 / SASB EM-MM-110a.1 disclosure conventions.

Usage

```
calculate_ghg_intensity(scope1_t, scope2_t, ore_processed_kt)
```

Arguments

scope1_t Numeric. Direct (Scope 1) emissions in tonnes CO2e.
 scope2_t Numeric. Energy-indirect (Scope 2) emissions in tonnes CO2e.
 ore_processed_kt Numeric. Ore processed in kilotonnes. Must be > 0.

Details

Intensity is calculated as:

$$(scope1_t + scope2_t) / (ore_processed_kt)$$

expressed as tCO2e per kilotonne of ore.

Value

A named list with metric, total_emissions_t, ghg_intensity (tCO2e/kt ore), and scope1_share_percent.

Examples

```
calculate_ghg_intensity(636000, 342000, 18481)
```

```
calculate_land_restoration
```

Calculate Land Restoration Rehabilitation Rate

Description

Computes the rehabilitation rate of disturbed mining land in alignment with SDG 15.3 (Land Degradation Neutrality). The function evaluates the percentage of disturbed land that has been restored or revegetated and reports the remaining unrestored area.

Usage

```
calculate_land_restoration(disturbed_area_ha, rehabilitated_area_ha)
```

Arguments

disturbed_area_ha Numeric. Total land disturbed by mining activities (in hectares). Must be a non-negative number.
 rehabilitated_area_ha Numeric. Total land rehabilitated or revegetated (in hectares). Must be a non-negative number.

Details

The rehabilitation rate is calculated as:

$$(rehabilitated_area_ha / disturbed_area_ha) * 100$$

If disturbed_area_ha is 0, the function safely returns 0 percent restored to avoid division-by-zero errors.

Value

A named list containing:

metric Character string. "SDG 15.3 - Land Restoration"

percent_restored Numeric. Rehabilitation rate rounded to 2 decimal places.

unrestored_area_ha Numeric. Remaining disturbed land not yet restored (in hectares).

Examples

```
calculate_land_restoration(100, 40)
```

```
calculate_land_restoration(
  disturbed_area_ha = 250,
  rehabilitated_area_ha = 180
)
```

```
calculate_land_restoration(0, 0)
```

```
calculate_safety_performance
```

Calculate Safety Performance Rates (SDG 8.8)

Description

Computes standard mining safety frequency rates per one million hours worked, following the ICMM safety data reporting convention and GRI 403-9: Total Recordable Injury Frequency Rate (TRIFR), Lost Time Injury Frequency Rate (LTIFR), and Fatality Frequency Rate.

Usage

```
calculate_safety_performance(
  hours_worked,
  recordable_injuries,
  lost_time_injuries,
  fatalities = 0
)
```

Arguments

hours_worked Numeric. Total exposure hours. Must be > 0.
 recordable_injuries Numeric. Count of recordable injuries (includes lost-time injuries).
 lost_time_injuries Numeric. Count of lost-time injuries.
 fatalities Numeric. Count of work-related fatalities. Default 0.

Details

Each rate is calculated as:

$$(events/hours_worked) \times 1,000,000$$

Value

A named list with metric, trifr, ltifr, fatality_rate, and hours_worked_millions.

Examples

```
calculate_safety_performance(  
  hours_worked = 6400000,  
  recordable_injuries = 32,  
  lost_time_injuries = 10  
)
```

calculate_waste_intensity

Calculate Mineral Waste Intensity (SDG 12)

Description

Computes tailings-to-ore and waste-rock (strip) ratios, core circular economy and mineral-waste metrics under GRI 306 / GRI 11.8 and the Global Industry Standard on Tailings Management (GISTM) context.

Usage

```
calculate_waste_intensity(ore_processed_kt, tailings_kt, waste_rock_kt = NULL)
```

Arguments

ore_processed_kt Numeric. Ore processed in kilotonnes. Must be > 0.
 tailings_kt Numeric. Tailings produced in kilotonnes.
 waste_rock_kt Numeric. Optional. Waste rock moved in kilotonnes.

Value

A named list with `metric`, `tailings_ratio` (`t tailings / t ore`) and, when supplied, `waste_rock_ratio` (`strip ratio`) and `total_mineral_waste_kt`.

Examples

```
calculate_waste_intensity(18481, 17750, waste_rock_kt = 45412)
```

```
calculate_water_efficiency
```

Calculate Water Use Efficiency (SDG 6.4)

Description

Computes water use efficiency and recycling rates in alignment with Global Reporting Initiative (GRI 303) standards and SDG Target 6.4.

Usage

```
calculate_water_efficiency(withdrawal_m3, discharge_m3, recycled_m3)
```

Arguments

<code>withdrawal_m3</code>	Numeric. Total water withdrawn from all sources (in cubic meters). Must be a non-negative number.
<code>discharge_m3</code>	Numeric. Total water discharged to all destinations (in cubic meters). Must be a non-negative number.
<code>recycled_m3</code>	Numeric. Total water recycled or reused (in cubic meters). Must be a non-negative number.

Details

The function calculates the "Recycling Rate" as:

$$(recycled_m3 / (withdrawal_m3 + recycled_m3)) * 100$$

And "Net Water Consumption" (GRI 303-5) as:

$$withdrawal_m3 - discharge_m3$$

Value

A named list containing:

`metric` Character string. "SDG 6.4 - Water Efficiency"

`net_consumption_m3` Numeric. Net water consumed (Withdrawal - Discharge).

`recycling_rate_percent` Numeric. Percentage of total water use that is recycled, rounded to 2 decimal places.

Examples

```
# Standard mining operation example
calculate_water_efficiency(
  withdrawal_m3 = 50000,
  discharge_m3 = 10000,
  recycled_m3 = 20000
)

# Zero discharge example (Closed loop)
calculate_water_efficiency(1000, 0, 500)

# Edge case: No water used
calculate_water_efficiency(0, 0, 0)
```

calculate_workforce_diversity

Calculate Workforce Diversity and Localisation (SDG 5 / SDG 8)

Description

Computes female employment share (GRI 405-1) and local employment share (GRI 202-2 / SASB EM-MM-210b) of the site workforce.

Usage

```
calculate_workforce_diversity(
  workforce,
  female_employees,
  local_employees = NULL
)
```

Arguments

workforce	Numeric. Total workforce headcount. Must be > 0.
female_employees	Numeric. Female employee headcount.
local_employees	Numeric. Optional. Employees hired from the local/host community.

Value

A named list with `metric`, `female_employment_pct` and, when supplied, `local_employment_pct`.

Examples

```
calculate_workforce_diversity(3200, 420, local_employees = 1900)
```

 compile_site_disclosures

Compile a site disclosure bundle

Description

Runs the existing MineSDG scoring engine once per reporting year and assembles a framework-neutral disclosure bundle. All framework report generators consume this object, which guarantees that the KPI engine remains the single source of truth and that no calculation is duplicated in the reporting layer.

Usage

```
compile_site_disclosures(site_data, site_id = NULL, years = NULL,
  registry = MineSDG::mining_kpi_registry, narratives = list(),
  entity_meta = list())
```

Arguments

site_data	A data.frame with one row per site-year using the demo_mine_sites column conventions.
site_id	Optional site identifier; required when site_data contains multiple sites.
years	Optional integer vector of years to include. Defaults to all years present for the site. The maximum year is the reporting year.
registry	KPI registry used for scoring; defaults to mining_kpi_registry . Pass a modified registry to apply corporate targets or alternative calibrations.
narratives	Named list of narrative disclosures keyed by narrative slot id (see the source_id column of the crosswalk datasets for source_type == "narrative" rows).
entity_meta	Named list of entity-level metadata (e.g. company, cin, fx_usd_inr). Used by report headers and currency transforms.

Value

An object of class minesdg_disclosure_bundle: a list with elements meta, kpis, scores, goal_scores, raw, deltas, narratives and registry_used.

See Also

[map_bundle_to_framework](#), [generate_gri_report](#)

Examples

```
bundle <- compile_site_disclosures(demo_mine_sites, site_id = "CU-ATAC",
  years = 2022:2024)
bundle
bundle_kpi(bundle, "ghg_intensity")
```

compute_sdg_stability	<i>Compute SDG Stability Metrics</i>
-----------------------	--------------------------------------

Description

Calculates variability and stability statistics for SDG indicator data.

Usage

```
compute_sdg_stability(data, group_by = c("indicator", "country"))
```

Arguments

- data A data.table returned by fetch_sdg_country_data().
- group_by Character vector of grouping columns.

Value

A data.table containing stability metrics.

Examples

```
dt <- fetch_sdg_country_data(indicator = "15.3.1", country = "IND")
compute_sdg_stability(dt)
```

demo_mine_sites	<i>Demo Mine Site Panel Data</i>
-----------------	----------------------------------

Description

A synthetic but realistic panel dataset of six mine sites (copper, gold, iron ore, coal, zinc, bauxite) across 2019-2024 with production, emissions, energy, water, land, safety, workforce and community fields. Designed for tutorials, tests and the bundled Shiny dashboard. All values are simulated; no real operation is represented.

Usage

```
demo_mine_sites
```

Format

A data frame with 36 rows (6 sites x 6 years) and 26 variables including ore_processed_kt, ghg_scope1_t, ghg_scope2_t, energy_gj, water_withdrawal_m3, water_recycled_m3, land_disturbed_ha, land_rehabilitated_ha, hours_worked, recordable_injuries, fatalities, female_employment_pct, local_employment_pct, revenue_musd, community_investment_musd, tailings_kt, and waste_rock_kt.

Source

Simulated via data-raw/make_datasets.R (seed 20260720).

Examples

```
head(demo_mine_sites)
```

demo_sdg_country	<i>Demo SDG Country Data (Offline Snapshot)</i>
------------------	---

Description

A synthetic offline snapshot shaped identically to the output of `fetch_sdg_country_data()`: indicators 15.3.1 (land degradation), 6.4.1 (water-use efficiency) and 8.8.1 (occupational injuries) for India, Australia, Chile and South Africa, 2015-2023. Enables every analytics and risk function to run without network access.

Usage

```
demo_sdg_country
```

Format

A data.table with 108 rows and 5 variables: indicator, country, year, value, unit.

Source

Simulated via data-raw/make_datasets.R; values follow plausible national trajectories but are not official UN statistics.

Examples

```
analyze_sdg_trend(demo_sdg_country[demo_sdg_country$indicator == "6.4.1", ])
```

explore_sdg_ontology	<i>Explore the SDG-to-Mining Ontology</i>
----------------------	---

Description

Queries the bundled `sdg_mining_ontology` by SDG goal or by mining sustainability domain, returning material topics, materiality ratings and disclosure-framework references.

Usage

```
explore_sdg_ontology(domain = NULL, ...)
```

Arguments

domain	Character or numeric. If character, filters by mining sustainability domain (partial, case-insensitive match), e.g. "water" or "biodiversity". If numeric (1-17), it is interpreted as an SDG goal number for backward compatibility.
...	Reserved for backward compatibility. A named argument goal may also be supplied.

Value

A data frame of matching ontology rows.

Examples

```
explore_sdg_ontology(domain = 6)
explore_sdg_ontology(domain = "biodiversity")
explore_sdg_ontology(goal = 6)
```

fetch_sdg_country_data

Fetch Country-Level SDG Data from the United Nations SDG API

Description

Retrieves Sustainable Development Goal (SDG) indicator data from the official 'United Nations SDG API'.

Usage

```
fetch_sdg_country_data(
  goal = NULL,
  indicator = NULL,
  country = NULL,
  year_range = NULL,
  save = FALSE,
  save_path = "./data/sdg_downloads/",
  formats = c("csv", "rds")
)
```

Arguments

goal	Numeric (1-17). Optional SDG goal number.
indicator	Character. SDG indicator code (e.g., "15.3.1").
country	Character. ISO3 country code (e.g., "IND", "AUS").
year_range	Numeric vector of length 2 (start_year, end_year).
save	Logical. If TRUE, saves data to disk. Default FALSE.
save_path	Character. Directory path for saving data. Default: "./data/sdg_downloads/".
formats	Character vector. Any combination of "csv", "rds". Default c("csv", "rds").

Details

At least one of goal or indicator must be provided.

- If indicator is supplied → fetch that specific indicator.
- If only goal is supplied → fetch all indicators under that goal.
- If both are supplied → consistency is validated.

country and year_range act as filters.

Value

A data.table containing SDG indicator data.

framework_crosswalk	<i>Access the framework crosswalk datasets</i>
---------------------	--

Description

Returns the disclosure-level mapping rules that connect reporting framework disclosures to MineSDG outputs. Mappings are shipped as package data; adding a disclosure means editing data, not code.

Usage

```
framework_crosswalk(framework = NULL)
```

Arguments

framework NULL (all frameworks) or one of "gri", "icmm", "brsr".

Value

A data.table with the common crosswalk schema (framework, section, disclosure_id, disclosure_title, requirement, source_type, source_id, transform, unit_out, comparative, mandatory, sdg_goal, notes).

Examples

```
framework_crosswalk("gri")[1:5, c("disclosure_id", "source_type",
                                   "source_id")]
```

generate_brsr_kpis	<i>BRSR Section C: principle-wise KPI tables</i>
--------------------	--

Description

Returns the Section C (principle-wise essential and leadership indicator) view of the BRSR mapping. Values are read from the disclosure bundle; no calculations are duplicated.

Usage

```
generate_brsr_kpis(site_data, site_id = NULL, years = NULL,
  registry = MineSDG::mining_kpi_registry, narratives = list(),
  entity_meta = list())
```

Arguments

site_data A site data.frame or a minesdg_disclosure_bundle.
 site_id, years, registry, narratives, entity_meta
 Passed to [compile_site_disclosures](#).

Value

A data.table of Section C quantitative disclosure rows.

Examples

```
generate_brsr_kpis(demo_mine_sites, site_id = "CO-JHAR",
  years = 2023:2024)
```

generate_brsr_report	<i>Generate a BRSR report</i>
----------------------	-------------------------------

Description

Builds a SEBI Business Responsibility and Sustainability Report covering Section A (general disclosures), Section B (management and process disclosures) and Section C (principle-wise performance KPIs). Monetary values without a supplied exchange rate are flagged partial rather than estimated.

Usage

```
generate_brsr_report(site_data, site_id = NULL, years = NULL,
  registry = MineSDG::mining_kpi_registry, narratives = list(),
  entity_meta = list(), output = NULL,
  format = c("html", "pdf", "docx", "xlsx"))
```

Arguments

site_data A site data.frame or a minesdg_disclosure_bundle.
site_id, years, registry, narratives, entity_meta
 Passed to [compile_site_disclosures](#) when site_data is a data.frame. Supply entity_meta\$fx_usd_inr to convert monetary lines to INR lakh.
output Optional path; when supplied the report is rendered.
format Output format used when output is supplied.

Value

A minesdg_report with extras sectionA, sectionB and sectionC.

See Also

[generate_brsr_sectionA](#), [generate_brsr_kpis](#)

Examples

```
rep <- generate_brsr_report(demo_mine_sites, site_id = "CO-JHAR",
                           years = 2023:2024,
                           entity_meta = list(company = "Demo Mining Ltd",
                                                fx_usd_inr = 83.2))
rep
```

generate_brsr_sectionA

BRSR Section A: general disclosures

Description

Returns the Section A view of the BRSR mapping. All three BRSR section functions are filters over one mapping computation.

Usage

```
generate_brsr_sectionA(site_data, site_id = NULL, years = NULL,
  registry = MineSDG::mining_kpi_registry, narratives = list(),
  entity_meta = list())
```

Arguments

site_data A site data.frame or a minesdg_disclosure_bundle.
site_id, years, registry, narratives, entity_meta
 Passed to [compile_site_disclosures](#).

Value

A data.table of Section A disclosure rows with values and statuses.

Examples

```
generate_brsr_sectionA(demo_mine_sites, site_id = "CO-JHAR",
                      years = 2024,
                      entity_meta = list(company = "Demo Mining Ltd"))
```

```
generate_brsr_sectionB
```

BRSR Section B: management and process disclosures

Description

Returns the Section B (management and process, per NGRBC principle) view of the BRSR mapping.

Usage

```
generate_brsr_sectionB(site_data, site_id = NULL, years = NULL,
                      registry = MineSDG::mining_kpi_registry, narratives = list(),
                      entity_meta = list())
```

Arguments

`site_data` A site data.frame or a minesdg_disclosure_bundle.
`site_id, years, registry, narratives, entity_meta`
 Passed to [compile_site_disclosures](#).

Value

A data.table of Section B disclosure rows.

Examples

```
generate_brsr_sectionB(demo_mine_sites, site_id = "CO-JHAR",
                      years = 2024)
```

```
generate_gri_report
```

Generate a GRI sustainability report

Description

Builds a GRI report covering GRI 302 (Energy), 303 (Water and Effluents), 304 (Biodiversity), 305 (Emissions), 306 (Waste), 403 (Occupational Health and Safety) and 413 (Local Communities). All quantitative disclosures are resolved through the disclosure bundle; no calculation is performed in the reporting layer.

Usage

```
generate_gri_report(site_data, site_id = NULL, years = NULL,
  registry = MineSDG::mining_kpi_registry, narratives = list(),
  entity_meta = list(),
  standards = c(302, 303, 304, 305, 306, 403, 413), output = NULL,
  format = c("html", "pdf", "docx", "xlsx"))
```

Arguments

site_data	A site data.frame (see demo_mine_sites) or a prepared minesdg_disclosure_bundle.
site_id, years, registry, narratives, entity_meta	Passed to compile_site_disclosures when site_data is a data.frame.
standards	Integer vector of GRI topic standards to include; default all supported (302, 303, 304, 305, 306, 403, 413).
output	Optional path. When supplied the report is rendered to format at this location (see render_minesdg_report).
format	Output format used when output is supplied.

Value

A minesdg_report object (invisibly when output is supplied).

See Also

[render_minesdg_report](#), [write_report_xlsx](#)

Examples

```
rep <- generate_gri_report(demo_mine_sites, site_id = "CU-ATAC",
  years = 2022:2024)
rep
gri_content_index(rep)[1:5, ]
```

generate_icmm_report	<i>Generate an ICMM Mining Principles report</i>
----------------------	--

Description

Assesses a site against the ICMM Mining Principles, producing a board scorecard, a traffic-light assessment and data-driven recommendations. Traffic lights reclassify existing MineSDG KPI scores against configurable thresholds; recommendations are evaluated from the [icmm_recommendation_rules](#) dataset.

Usage

```
generate_icmm_report(site_data, site_id = NULL, years = NULL,
  registry = MineSDG::mining_kpi_registry, narratives = list(),
  entity_meta = list(), params = icmm_default_params(), output = NULL,
  format = c("html", "pdf", "docx", "xlsx"))
```

Arguments

`site_data` A site data.frame or a minesdg_disclosure_bundle.

`site_id, years, registry, narratives, entity_meta` Passed to [compile_site_disclosures](#) when `site_data` is a data.frame.

`params` ICMC parameters from [icmm_default_params](#).

`output` Optional path; when supplied the report is rendered.

`format` Output format used when output is supplied.

Value

A minesdg_report with extra elements `$extras$scorecard`, `$extras$traffic_lights` and `$extras$recommendations`.

Examples

```
rep <- generate_icmm_report(demo_mine_sites, site_id = "CU-ATAC",
  years = 2023:2024)
rep$extras$traffic_lights[1:5, ]
rep$extras$recommendations
```

```
generate_mining_esg_index
```

Generate Composite Mining ESG Index

Description

Builds a composite Mining ESG Risk Index by aggregating multiple SDG indicators using flexible weighting schemes.

Usage

```
generate_mining_esg_index(
  data,
  indicators,
  weighting_method = "equal",
  custom_weights = NULL
)
```

Arguments

- data data.table returned by fetch_sdg_country_data().
- indicators Character vector of SDG indicator codes.
- weighting_method Character. One of: "equal", "domain_weighted", "custom".
- custom_weights Named numeric vector of weights (required if weighting_method = "custom").

Value

- A data.table containing:
 - country Country ISO3
 - composite_score Aggregated ESG risk score
 - risk_category Overall risk classification

generate_mining_risk_profile
<i>Generate Mining SDG Risk Profile</i>

Description

Integrates SDG trend, stability, benchmarking, and mining-sector relevance to produce a structured ESG risk assessment.

Usage

generate_mining_risk_profile(data, indicator)

Arguments

- data data.table returned from fetch_sdg_country_data()
- indicator Character. SDG indicator code.

Value

- A structured list containing:
 - indicator SDG indicator
 - domain Mining domain
 - risk_score Numeric weighted risk score
 - risk_category Risk classification
 - executive_summary Narrative explanation

```
generate_sdg_executive_summary
```

Generate Executive Summary for SDG Performance

Description

Creates an executive-ready narrative summary based on SDG trend analysis and optional benchmark comparison.

Usage

```
generate_sdg_executive_summary(trend_data, benchmark_data = NULL, digits = 2)
```

Arguments

`trend_data` Output from `analyze_sdg_trend()`.
`benchmark_data` Optional output from `benchmark_sdg_performance()`.
`digits` Number of digits for rounding values. Default 2.

Value

Character vector containing executive summary text.

Examples

```
dt <- fetch_sdg_country_data(indicator = "15.3.1", country = "IND")
trend <- analyze_sdg_trend(dt)
summary_text <- generate_sdg_executive_summary(trend)
cat(summary_text)
```

```
get_sdg_metadata
```

Get SDG Metadata (Cached)

Description

Retrieves SDG metadata from the UN API and caches it for the current R session. Requires network access; downstream helpers such as [list_sdg_indicators](#) use this cache.

Usage

```
get_sdg_metadata()
```

Value

A data.table of SDG indicator metadata.

<code>gri_content_index</code>	<i>GRI content index</i>
--------------------------------	--------------------------

Description

Returns the GRI content index table (disclosure, title, status, source lineage) that assurance and reporting teams attach to published reports.

Usage

```
gri_content_index(report)
```

Arguments

`report` A minesdg_report produced by [generate_gri_report](#).

Value

A data.frame with columns disclosure_id, disclosure_title, status, source and omission_reason.

Examples

```
rep <- generate_gri_report(demo_mine_sites, site_id = "AU-KALG",
                           years = 2024)
head(gri_content_index(rep))
```

<code>gri_crosswalk</code>	<i>GRI disclosure crosswalk</i>
----------------------------	---------------------------------

Description

Disclosure-level mapping rules connecting GRI 302/303/304/305/306/403/413 disclosures to MineSDG outputs. Framework mappings are data: report generators contain no framework-specific logic.

Usage

```
gri_crosswalk
```

Format

A data.table with one row per disclosure and columns framework, section, disclosure_id, disclosure_title, requirement, source_type (kpi/raw/score/narrative/none), source_id, transform, unit_out, comparative, mandatory, sdg_goal, notes. Carries a crosswalk_version attribute.

Source

Curated in data-raw/make_crosswalks.R; requirement texts are original paraphrases referencing the GRI Standards 2021 and the GRI 11 Mining Sector Standard 2022.

icmm_crosswalk	<i>ICMM Mining Principles crosswalk</i>
----------------	---

Description

Mapping of ICMM Mining Principles and selected performance areas (safety, climate, water, tailings, biodiversity, social) to MineSDG outputs.

Usage

icmm_crosswalk

Format

See [gri_crosswalk](#) for the common schema.

Source

Curated in data-raw/make_crosswalks.R; paraphrases referencing the ICMM Mining Principles (2020).

icmm_default_params	<i>Default parameters for ICMM reporting</i>
---------------------	--

Description

Builds the parameter list consumed by [generate_icmm_report](#).

Usage

icmm_default_params(green = 70, amber = 40)

Arguments

- | | |
|-------|--|
| green | Minimum 0-100 score for a green traffic light (default 70). |
| amber | Minimum score for amber (default 40); anything below is red. |

Value

A named list of ICMM report parameters.

Examples

```
icmm_default_params()
icmm_default_params(green = 75, amber = 50)
```

icmm_recommendation_rules	<i>ICMM recommendation rules</i>
---------------------------	----------------------------------

Description

Data-driven recommendation rules evaluated by [generate_icmm_report](#): when a KPI or mining-domain traffic light matches a trigger, the associated recommendation is emitted.

Usage

```
icmm_recommendation_rules
```

Format

A data.table with columns principle, trigger_type (kpi_light or domain_light), trigger_id, trigger_value, priority, recommendation.

Source

Curated in data-raw/make_crosswalks.R.

list_mining_kpis	<i>List Mining KPIs from the Registry</i>
------------------	---

Description

Returns the bundled [mining_kpi_registry](#), optionally filtered by SDG goal or improvement direction.

Usage

```
list_mining_kpis(sdg_goal = NULL, direction = NULL)
```

Arguments

- sdg_goal Numeric (1-17). Optional filter by primary SDG goal.
- direction Character. Optional: "lower_better" or "higher_better".

Value

A data frame of matching KPI registry rows.

Examples

```
list_mining_kpis(sdg_goal = 8)
list_mining_kpis(direction = "higher_better")
```

list_sdg_indicators	<i>List SDG Indicators and Descriptions</i>
---------------------	---

Description

Retrieves the official list of SDG indicators, including indicator codes, descriptions, goal numbers, and tier classification.

Usage

```
list_sdg_indicators(goal = NULL)
```

Arguments

goal	Numeric (1–17). Optional SDG goal number to filter indicators.
------	--

Details

Uses internally cached metadata via `get_sdg_metadata()`.

Value

A `data.table` containing:

goal SDG goal number

indicator Indicator code

description Indicator description

tier Tier classification

Examples

```
list_sdg_indicators()
list_sdg_indicators(goal = 15)
```

map_bundle_to_framework	<i>Map a disclosure bundle onto a reporting framework</i>
-------------------------	---

Description

Joins a disclosure bundle against a framework’s crosswalk dataset and resolves every disclosure to a value and a status (reported, partial, narrative_provided, narrative_required, not_in_scope). Contains no framework-specific logic: frameworks differ only in their crosswalk data.

Usage

```
map_bundle_to_framework(bundle, framework, include_voluntary = TRUE, ...)
```

Arguments

- bundle A minesdg_disclosure_bundle (or site data.frame, compiled automatically).
- framework One of "gri", "icmm", "brsr".
- include_voluntary Include non-mandatory disclosures (default TRUE).
- ... Passed to [compile_site_disclosures](#) when bundle is a data.frame.

Value

An object of class minesdg_framework_mapping: a list with framework, table (one row per disclosure with value, value_prior, narrative, status), coverage and meta.

Examples

```
b <- compile_site_disclosures(demo_mine_sites, site_id = "CU-ATAC",
                             years = 2023:2024)
m <- map_bundle_to_framework(b, "gri")
m
summary(m)
```

map_sdg_to_mining_domain	<i>Map SDG Indicator to Mining-Sector Domain</i>
--------------------------	--

Description

Classifies an SDG goal or indicator into a mining-sector sustainability domain. Returns structured interpretation including domain classification, relevance score (1–5), and strategic narrative explanation.

Usage

```
map_sdg_to_mining_domain(indicator = NULL, goal = NULL)
```

Arguments

indicator Character. Optional SDG indicator code (e.g., "15.3.1").

goal Numeric (1–17). Optional SDG goal number.

At least one of indicator or goal must be provided.

Value

A structured list containing:

goal SDG goal number

domain Mining sustainability domain

relevance_score Numeric score (1–5)

narrative Mining-sector strategic interpretation

Examples

```
map_sdg_to_mining_domain(indicator = "15.3.1")
map_sdg_to_mining_domain(goal = 13)
```

mining_kpi_registry	<i>Mining KPI Registry</i>
---------------------	----------------------------

Description

A reference registry of 18 site-level mining sustainability KPIs with units, SDG target alignment, improvement direction, and indicative good/poor threshold values used by [score_site_sdg\(\)](#) for 0-100 normalisation. Thresholds are indicative sector reference points; organisations should calibrate them to their commodity and jurisdiction.

Usage

```
mining_kpi_registry
```

Format

A data frame with 18 rows and 9 variables:

kpi_id Machine-readable KPI identifier

kpi_name Human-readable KPI name

unit Measurement unit

sdg_goal Primary SDG goal number
sdg_target Primary SDG target
direction "lower_better" or "higher_better"
good_value Value that scores 100
poor_value Value that scores 0
framework_reference GRI / ICMM / SASB / GISTM citations

Examples

```
list_mining_kpis(sdg_goal = 8)
```

plot_domain_radar	<i>Mining-domain radar chart for a site scorecard</i>
-------------------	---

Description

Rolls goal-level scores up to the ontology's mining domains and plots them as a radar chart. The rollup is an unweighted mean of goal scores per domain – a presentation-level aggregation of values already computed by the scoring engine.

Usage

```
plot_domain_radar(x, year = NULL)
```

Arguments

x A minesdg_site_score or minesdg_disclosure_bundle.
year Year to plot when x is a bundle; defaults to the reporting year.

Value

A ggplot object.

Examples

```
sc <- score_site_sdg(demo_mine_sites[demo_mine_sites$site_id ==  
  "CU-ATAC" & demo_mine_sites$year == 2024, ])  
plot_domain_radar(sc)
```

plot_mining_esg_index *Plot Mining ESG Composite Index*

Description

Creates a ranked horizontal bar chart of composite ESG risk.

Usage

```
plot_mining_esg_index(index_data)
```

Arguments

index_data data.table returned by generate_mining_esg_index()

Value

ggplot object

plot_sdg_benchmark *Plot SDG Benchmark Comparison*

Description

Creates a benchmark comparison bar chart.

Usage

```
plot_sdg_benchmark(benchmark_data)
```

Arguments

benchmark_data Output from benchmark_sdg_performance().

Value

A ggplot object.

plot_sdg_convergence	<i>Plot SDG Convergence</i>
----------------------	-----------------------------

Description

Creates a convergence scatter plot showing initial value vs growth rate.

Usage

```
plot_sdg_convergence(data, base_year = NULL, final_year = NULL)
```

Arguments

data	A data.table returned by fetch_sdg_country_data().
base_year	Optional base year. If NULL, earliest year is used.
final_year	Optional final year. If NULL, latest year is used.

Value

A ggplot object.

plot_sdg_radar	<i>SDG radar chart for a site scorecard</i>
----------------	---

Description

Plots goal-level SDG scores as a radar (polar) chart.

Usage

```
plot_sdg_radar(x, year = NULL)
```

Arguments

x	A minesdg_site_score or minesdg_disclosure_bundle.
year	Year to plot when x is a bundle; defaults to the reporting year.

Value

A ggplot object.

Examples

```
sc <- score_site_sdg(demo_mine_sites[demo_mine_sites$site_id ==  
  "CU-ATAC" & demo_mine_sites$year == 2024, ])  
plot_sdg_radar(sc)
```

plot_sdg_trend	<i>Plot SDG Indicator Trend</i>
----------------	---------------------------------

Description

Creates a publication-ready time-series plot for SDG indicator data.

Usage

```
plot_sdg_trend(data, indicator = NULL, country = NULL)
```

Arguments

data	A data.table returned by fetch_sdg_country_data().
indicator	Optional indicator code to filter.
country	Optional ISO3 country code to filter.

Value

A ggplot object.

plot_sdg_volatility	<i>Plot SDG Volatility</i>
---------------------	----------------------------

Description

Visualizes SDG indicator variability across countries.

Usage

```
plot_sdg_volatility(stability_data)
```

Arguments

stability_data	Output from compute_sdg_stability().
----------------	--------------------------------------

Value

A ggplot object.

render_minesdg_report *Render a MineSDG report to HTML, PDF, DOCX or XLSX*

Description

HTML, PDF and DOCX rendering use the Quarto templates shipped in `inst/quarto/` and require the Quarto CLI plus the **quarto** R package. XLSX output is delegated to [write_report_xlsx](#) and requires **openxlsx2**. Rendering happens in a temporary directory; the installed package is never written to.

Usage

```
render_minesdg_report(report, output_file,  
  format = c("html", "pdf", "docx", "xlsx"), template = NULL,  
  quiet = TRUE)
```

Arguments

<code>report</code>	A <code>minesdg_report</code> object.
<code>output_file</code>	Path of the output document. Its directory must exist.
<code>format</code>	One of "html", "pdf", "docx", "xlsx".
<code>template</code>	Optional path to a user-supplied .qmd template used instead of the packaged one.
<code>quiet</code>	Suppress Quarto output (default TRUE).

Value

The path to the rendered file, invisibly.

Examples

```
rep <- generate_gri_report(demo_mine_sites, site_id = "CU-ATAC",  
  years = 2023:2024)  
out <- render_minesdg_report(rep, file.path(tempdir(), "gri.html"))
```

`run_minesdg_dashboard` *Launch the MineSDG Shiny Dashboard*

Description

Starts the interactive MineSDG dashboard: portfolio SDG scorecards, site KPI deep-dives, ontology exploration, the KPI registry, and CSV upload of your own site data (schema of [demo_mine_sites](#)).

Usage

```
run_minesdg_dashboard(...)
```

Arguments

... Passed on to `shiny::runApp()`, e.g. `port`, `launch.browser`.

Details

Requires the shiny package (and optionally DT for enhanced tables), listed in Suggests.

Value

Invisibly returns the value of `shiny::runApp()` (called for its side effect of launching the app).

Examples

```
## Not run:
run_minesdg_dashboard()

## End(Not run)
```

`score_portfolio_sdg` *Score a portfolio of sites*

Description

Loops the existing [score_site_sdg](#) engine over every site-year in a portfolio and returns a tidy table of composite scores and grades, ready for portfolio dashboards and benchmarking. No new scoring logic is introduced.

Usage

```
score_portfolio_sdg(site_data,
  registry = MineSDG::mining_kpi_registry)
```

Arguments

- site_data A data.frame with one row per site-year (multiple sites allowed), using the [demo_mine_sites](#) conventions.
- registry KPI registry passed through to [score_site_sdg](#).

Value

A data.table with columns site_id, site_name, commodity, country, year, composite_score, grade, ordered by year then descending score.

Examples

```
portfolio <- score_portfolio_sdg(demo_mine_sites)
portfolio[portfolio$year == 2024, ]
```

score_site_sdg	<i>Score a Mine Site Against SDG-Aligned KPI Thresholds</i>
----------------	---

Description

Transforms one year of raw site operational data into a normalised 0-100 SDG scorecard. Each KPI is derived from the raw fields, then linearly rescaled between the poor_value (score 0) and good_value (score 100) reference thresholds in [mining_kpi_registry](#), respecting the KPI's improvement direction. Scores are aggregated to SDG-goal level and into a materiality-weighted composite (weights from [sdg_mining_ontology](#)).

Usage

```
score_site_sdg(site_data, registry = MineSDG::mining_kpi_registry)
```

Arguments

- site_data A one-row data.frame (e.g. one site-year of [demo_mine_sites](#)).
- registry KPI registry data frame. Default [mining_kpi_registry](#). Supply a customised copy to calibrate thresholds to your commodity.

Details

Expected columns in site_data (one row; extra columns are ignored): ore_processed_kt, ghg_scope1_t, ghg_scope2_t, energy_gj, renewable_energy_pct, water_withdrawal_m3, water_recycled_m3, land_disturbed_ha, land_rehabilitated_ha, hours_worked, recordable_injuries, lost_time_injuries, fatalities, female_employment_pct, local_employment_pct, revenue_musd, community_investment_musd, tailings_kt, waste_rock_kt. Missing fields simply drop the corresponding KPIs from the score-card.

Value

An object of class minesdg_site_score: a list with scorecard (data.table: kpi_id, kpi_name, sdg_goal, value, unit, score), goal_scores (data.table: sdg_goal, mining_domain, weight, goal_score), composite_score (0-100), and grade (A-E).

Examples

```
site <- demo_mine_sites[demo_mine_sites$site_id == "CU-ATAC" &
  demo_mine_sites$year == 2024, ]
score_site_sdg(site)
```

sdg_mining_ontology	<i>SDG-to-Mining Ontology</i>
---------------------	-------------------------------

Description

A structured ontology mapping all 17 Sustainable Development Goals to mining-sector sustainability domains, material topics, sector materiality ratings, and disclosure-framework references (GRI 11 Mining Sector Standard, ICMM Mining Principles, SASB EM-MM, SEBI BRSR).

Usage

```
sdg_mining_ontology
```

Format

- A data frame with 17 rows and 10 variables:
- goal** SDG goal number (1-17)
 - goal_name** Official SDG goal name
 - mining_domain** Mining sustainability domain classification
 - materiality** Sector materiality rating (1 = peripheral, 5 = core)
 - material_topic** Mining-specific material topic
 - gri_reference** GRI Standards / GRI 11 sector standard reference
 - icmm_principle** ICMM Mining Principles reference
 - sasb_emm** SASB Metals & Mining (EM-MM) metric reference
 - brsr_principle** SEBI BRSR principle reference (India)
 - example_kpis** Semicolon-separated representative site KPIs

Source

Compiled from public framework documentation: GRI 11 (2024), ICMM Mining Principles, SASB Metals & Mining Standard, SEBI BRSR.

Examples

```
head(sdg_mining_ontology[, c("goal", "mining_domain", "materiality")])
```

summarize_sdg_data	<i>Summarize SDG Indicator Data</i>
--------------------	-------------------------------------

Description

Generates descriptive statistics and data quality diagnostics for SDG indicator datasets retrieved via `fetch_sdg_country_data()`.

Usage

```
summarize_sdg_data(data, group_by = NULL, na.rm = TRUE)
```

Arguments

<code>data</code>	A <code>data.table</code> returned by <code>fetch_sdg_country_data()</code> .
<code>group_by</code>	Character vector of column names to group by. Allowed values are "indicator", "country", and "year". Default is NULL (overall summary).
<code>na.rm</code>	Logical. Should missing values be removed when computing statistics? Default TRUE.

Details

Supports optional grouping by indicator, country, year, or combinations thereof.

Value

A `data.table` containing summary statistics.

Examples

```
dt <- fetch_sdg_country_data(  
  indicator = "15.3.1",  
  country = "IND"  
)  
  
summarize_sdg_data(dt)  
summarize_sdg_data(dt, group_by = "year")  
summarize_sdg_data(dt,  
  group_by = c("indicator", "country"))
```

 validate_sdg_indicator

Validate SDG Indicator Code

Description

Checks whether a provided SDG indicator code exists in the official United Nations SDG indicator database.

Usage

```
validate_sdg_indicator(indicator, goal = NULL)
```

Arguments

indicator	Character. SDG indicator code (e.g., "15.3.1").
goal	Numeric (1–17). Optional SDG goal number for consistency check.

Details

Optionally verifies consistency between indicator and goal.
 Uses internally cached metadata via `get_sdg_metadata()`.

Value

Logical TRUE if valid. Stops with error if invalid.

Examples

```
validate_sdg_indicator("15.3.1")
validate_sdg_indicator("15.3.1", goal = 15)
```

 write_report_xlsx

Export a MineSDG report's tables to an Excel workbook

Description

Writes a workbook with a Cover sheet (metadata and coverage), the disclosure tables, and a Data Annex containing the bundle's KPI and raw-input tables. Requires the **openxlsx2** package.

Usage

```
write_report_xlsx(report, output_file)
```

Arguments

<code>report</code>	A minesdg_report object.
<code>output_file</code>	Path of the .xlsx file to write.

Value

The path to the written file, invisibly.

Examples

```
rep <- generate_gri_report(demo_mine_sites, site_id = "CU-ATAC",
                           years = 2023:2024)
out <- write_report_xlsx(rep, file.path(tempdir(), "gri.xlsx"))
```

Index

* datasets

- brsr_crosswalk, [6](#)
- demo_mine_sites, [16](#)
- demo_sdg_country, [17](#)
- gri_crosswalk, [27](#)
- icmm_crosswalk, [28](#)
- icmm_recommendation_rules, [29](#)
- mining_kpi_registry, [32](#)
- sdg_mining_ontology, [40](#)

analyze_sdg_convergence, [3](#)

analyze_sdg_trend, [4](#)

as_disclosure_bundle, [4](#)

benchmark_sdg_performance, [5](#)

brsr_crosswalk, [6](#)

bundle_kpi, [6](#)

bundle_raw, [7](#)

bundle_score, [7](#)

calculate_community_investment, [8](#)

calculate_energy_intensity, [9](#)

calculate_ghg_intensity, [9](#)

calculate_land_restoration, [10](#)

calculate_safety_performance, [11](#)

calculate_waste_intensity, [12](#)

calculate_water_efficiency, [13](#)

calculate_workforce_diversity, [14](#)

compile_site_disclosures, [4](#), [5](#), [15](#), [20–24](#), [31](#)

compute_sdg_stability, [16](#)

demo_mine_sites, [16](#), [23](#), [38](#), [39](#)

demo_sdg_country, [17](#)

explore_sdg_ontology, [17](#)

fetch_sdg_country_data, [18](#)

fetch_sdg_country_data(), [17](#)

framework_crosswalk, [19](#)

generate_brsr_kpis, [20](#), [21](#)

generate_brsr_report, [20](#)

generate_brsr_sectionA, [21](#), [21](#)

generate_brsr_sectionB, [22](#)

generate_gri_report, [15](#), [22](#), [27](#)

generate_icmm_report, [23](#), [28](#), [29](#)

generate_mining_esg_index, [24](#)

generate_mining_risk_profile, [25](#)

generate_sdg_executive_summary, [26](#)

get_sdg_metadata, [26](#)

gri_content_index, [27](#)

gri_crosswalk, [6](#), [27](#), [28](#)

icmm_crosswalk, [28](#)

icmm_default_params, [24](#), [28](#)

icmm_recommendation_rules, [23](#), [29](#)

list_mining_kpis, [29](#)

list_sdg_indicators, [26](#), [30](#)

map_bundle_to_framework, [15](#), [31](#)

map_sdg_to_mining_domain, [31](#)

mining_kpi_registry, [6](#), [15](#), [29](#), [32](#), [39](#)

plot_domain_radar, [33](#)

plot_mining_esg_index, [34](#)

plot_sdg_benchmark, [34](#)

plot_sdg_convergence, [35](#)

plot_sdg_radar, [35](#)

plot_sdg_trend, [36](#)

plot_sdg_volatility, [36](#)

render_minesdg_report, [23](#), [37](#)

run_minesdg_dashboard, [38](#)

score_portfolio_sdg, [38](#)

score_site_sdg, [38](#), [39](#), [39](#)

score_site_sdg(), [32](#)

sdg_mining_ontology, [17](#), [39](#), [40](#)

shiny::runApp(), [38](#)

summarize_sdg_data, [41](#)

`validate_sdg_indicator`, [42](#)

`write_report_xlsx`, [23](#), [37](#), [42](#)