

Hotspot Extraction

vv0.4.1

Jerónimo Rodríguez Escobar

2025-12-10

Table of contents

| | | |
|----------|---|-----------|
| 1 | Setup and Libraries | 1 |
| 2 | Run Metadata Banner | 2 |
| 3 | Add regional attributes to Grid (countries & biomes) | 2 |
| 4 | Reshape grid data to a service-long table | 3 |
| 4.1 | QA: sanity checks on plt_long | 3 |
| 4.1.1 | Inspect pivot output | 3 |
| 5 | Hotspot extraction workflow (global + subregional) | 4 |
| 5.1 | Hotspot rules & export configuration | 4 |
| 5.2 | Validate hotspot configuration | 5 |
| 5.3 | Export hotspot layers | 6 |
| 6 | Checkpoint recap | 6 |
| 7 | Trimmed change bar plots | 6 |
| 7.1 | Signed means (zeros included, trimmed caps) | 7 |
| 8 | Hotspot violin plots | 12 |

1 Setup and Libraries

Load all core packages (tidyverse, spatial, plotting helpers) and initialize reproducibility settings. This chunk also sources R(paths.R and runs devtools::load_all() so downstream chunks can reuse helper functions.

2 Run Metadata Banner

Print run metadata (analysis version, timestamp, git commit, data root) so exported reports remain traceable.

Published: 2025-12-10 10:11 PST



Run metadata

- Analysis version: v0.4.1
- Rendered: 2025-12-10 10:11 PST
- Git: feat/signed-bars-global-line @ 11166a5
- Data root: /home/jeronimo/data/global_ncp

3 Add regional attributes to Grid (countries & biomes)

Attach country and biome IDs to the 10 km grid (`sf_f`) so downstream aggregation/hotspot steps can group by region. This chunk only runs if the processed GPKG is missing.

We enrich the 10 km grid (`sf_f`) with country and WWF biome IDs so we can aggregate and compare change by subregions (World Bank region, income group, continent, UN region, and biome). We use a point-on-surface join to avoid sliver/overlap issues, keep only the needed fields, and write a single enriched GPKG for downstream grouping, hotspot extraction, and plotting.



What this step does

- Reads country and biome layers from `vectors/`.
- Joins them to the 10 km grid via point-on-surface.
- Writes `processed/10k_change_calc.gpkg` for downstream analysis (pivoting, hotspots, plots). **Inputs:** `sf_f` grid; `vectors/cartographic_ee_ee_r264_correspondence.gpkg`; `vectors/Biome.gpkg` **Output:** `processed/10k_change_calc.gpkg` (grid + regional attributes)

Why point-on-surface? It's robust for odd cell shapes and avoids polygon–polygon sliver issues.

When to bump the version? If the joined attributes change schema/meaning (e.g., new grouping columns), bump **MINOR**; if file name/structure changes in a breaking way, bump **MAJOR**.

4 Reshape grid data to a service-long table

Pivot the processed grid into a tidy `plt_long` table (one row per cell × service) and standardize service labels/factor order. Cached so we only rebuild when inputs change.

Start by turning the wide grid table (one row per 10 km cell with many `*_abs_chg / *_pct_chg` columns) into an **analysis-ready long format**: one row per **cell × service** with two value columns: `abs_chg` and `pct_chg`. This makes it easy to rank, filter, and facet by service in later hotspot steps.

- What this chunk does**
- 1) Reads the processed grid from `processed/10k_change_calc.gpkg` and ensures a unique `fid`.
 - 2) Splits “ID/grouping” columns from change columns.
 - 3) Pivots `*_abs_chg / *_pct_chg` to long, then back to wide as `abs_chg / pct_chg` per service.
 - 4) Cleans obvious issues (drops `Inf/NA` where appropriate) so plots and tests won’t choke.
 - 5) Applies a human-readable service label mapping (e.g., `n_export → N_export`).
 - 6) Sets a canonical facet order (`svc_order`) so plots are consistent across the report.

Inputs: `processed/10k_change_calc.gpkg` (contains `fid`, `c_fid`, service change fields, and sub-regional tags like `region_wb`, `income_grp`, `BIOME`, etc.).

Output: `plt_long` (tidy tibble) with columns

`fid`, `c_fid`, `service`, `abs_chg`, `pct_chg`, <grouping/socio vars>.



Tip

Why long format?

Ranking, percentile cuts, ECDFs, and faceted plots are all simpler and faster when each `service` is a row attribute rather than a separate column.

4.1 QA: sanity checks on `plt_long`

Before ranking/thresholding, a lightweight QA helps catch silent problems (e.g., an empty service, unexpected sparsity, or leftover `Insfs`):

- Confirm `plt_long` exists and peek the structure.
- Count rows per `service` to spot outliers in coverage (e.g., a service that only exists in a few countries).

4.1.1 Inspect pivot output

```
# ----- Quick sanity -----
# stopifnot(exists("plt_long"))
# dplyr::glimpse(head(plt_long, 5), width = 80)
# plt_long |>
#   dplyr::count(service, name = "rows") |>
```

```
#   dplyr::arrange(dplyr::desc(rows)) |>
#   print(n = 50)
```

5 Hotspot extraction workflow (global + subregional)

We identify per-service hotspots using a 5% percentile rule and **direction vectors**: - `loss_services` (e.g., Nature_Access, Pollination, N/Sed_Ret_Ratio, C_Risk_Red_Ratio) → we flag the **lowest** values; - `gain_services` (Sed_export, N_export, C_Risk) → we flag the **highest** values.

We run this **once globally** and then **once per subregion** (World Bank region, income group, continent, UN region, WWF biome). For each run and for each metric (absolute and percent change) we write a compact **GPKG** containing only the hotspot cells, plus a CSV index:

- Output root: `processed/hotspots/`
 - `abs/global/hotspots_global_abs.gpkg`
 - `pct/global/hotspots_global_pct.gpkg`
 - `abs/<group_col>/hotspots_<group_col>_<group_val>_abs.gpkg`
 - `pct/<group_col>/hotspots_<group_col>_<group_val>_pct.gpkg`
- Index: `processed/hotspots/_hotspots_index.csv` (columns: scope, group_col, group_val, metric, n_hot, gpkg).

These files are meant for QGIS/QA and downstream stats (e.g., KS) without recomputing hotspots.

5.1 Hotspot rules & export configuration

The analysis uses a single, central configuration so the hotspot rules are consistent everywhere:

Thresholding: we flag hotspots using the top/bottom tails of the distribution per service. Here we use a percentile cutoff (e.g., 5%) rather than a fixed count.

Direction of concern: services in loss are “worse when they go down” (we keep the lowest tail); services in gain are “worse when they go up” (we keep the highest tail).

Combos (optional): grouped service sets that we count per cell for quick composite summaries.

Export switches: choose whether to write GPKGs and/or the CSV index.

```
HOTS_CFG <- list(
  analysis_name    = "global_NCP_hotspots",
  pct_cutoff      = 0.05,
  threshold_mode  = "percent",
  rule_mode        = "vectors",
  loss = c("Nature_Access", "Pollination", "N_Ret_Ratio", "Sed_Ret_Ratio", "C_Risk_Red_Ratio"),
  gain = c("Sed_export", "N_export", "C_Risk"),
  combos = list(
    deg_combo = c("Nature_Access", "Pollination", "N_export", "Sed_export", "C_Risk"),
    rec_combo = c("Nature_Access", "Pollination", "N_Ret_Ratio", "Sed_Ret_Ratio", "C_Risk_Red_Ratio")))
```

```

),
# centralize the grouping columns here
groupings = c("income_grp", "region_wb", "continent", "region_un", "WWF_biome"),
# IO
write_layers = TRUE,
write_index = TRUE,
out_dir = file.path(data_dir(), "processed", "hotspots")
)

```

::: callout-note
****Hotspot configuration****

- Cutoff: 5% (percent)
- Rule mode: vectors
- Loss services: Nature_Access, Pollination, N_Ret_Ratio, Sed_Ret_Ratio, C_Risk_Red_Ratio
- Gain services: Sed_export, N_export, C_Risk
- Combos: ****deg_combo****: Nature_Access, Pollination, N_export, Sed_export, C_Risk
****rec_com**
...
- Groupings: income_grp, region_wb, continent, region_un, WWF_biome
- ...- Write layers: TRUE | Write index: TRUE
- Output dir: `/home/jeronimo/data/global_ncp/processed/hotspots`

:::

5.2 Validate hotspot configuration

```

# stopifnot(exists("plt_long"))
# svc_all <- unique(plt_long$service)
# if (length(intersect(HOTS_CFG$loss, HOTS_CFG$gain)) > 0) {
#   stop("A service appears in BOTH `loss` and `gain`. Fix HOTS_CFG.")
# }
# miss_loss <- setdiff(HOTS_CFG$loss, svc_all)
# miss_gain <- setdiff(HOTS_CFG$gain, svc_all)
# if (length(miss_loss) > 0 || length(miss_gain) > 0) {
#   warning("Services in HOTS_CFG not found in `plt_long$service`:\n",
#         if (length(miss_loss)) paste0(" - missing loss: ", paste(miss_loss, collapse=", ")),
#         if (length(miss_gain)) paste0(" - missing gain: ", paste(miss_gain, collapse=", ")))
# }

```

5.3 Export hotspot layers

i Note

Hotspot export module

- Computes hotspot cells once (global + by subregion) for ABS and PCT change.
- Writes compact GPKGs for mapping/QA and maintains `_hotspots_index.csv`.
- Prereqs: `plt_long` in memory, `HOTS_CFG` defined (loss/gain/combos/etc.).

6 Checkpoint recap

Wired the project with a metadata banner (version, git, data root) for reproducibility.

Enriched the 10 km grid with country/biome tags (documented chunk, eval: false) and standardized the working input at processed/10k_change_calc.gpkg.

Reshaped to analysis-ready long format (`plt_long`), cleaned basic issues, harmonized service labels, and set a canonical facet order.

Centralized hotspot rules in `HOTS_CFG` (loss/gain, combos, cutoff, groupings, IO).

Exported hotspots once (global + by subregion, for abs/pct change) to compact GPKGs under processed/hotspots/, and wrote a manifest: processed/hotspots/_hotspots_index.csv.

Why this structure? Heavy work (ranking/thresholding/joining) is done once. The manifest gives us traceability and fast loading for downstream steps (bar plots, violins, KS tests) without re-computation.

7 Trimmed change bar plots

i Note

How to read these bars

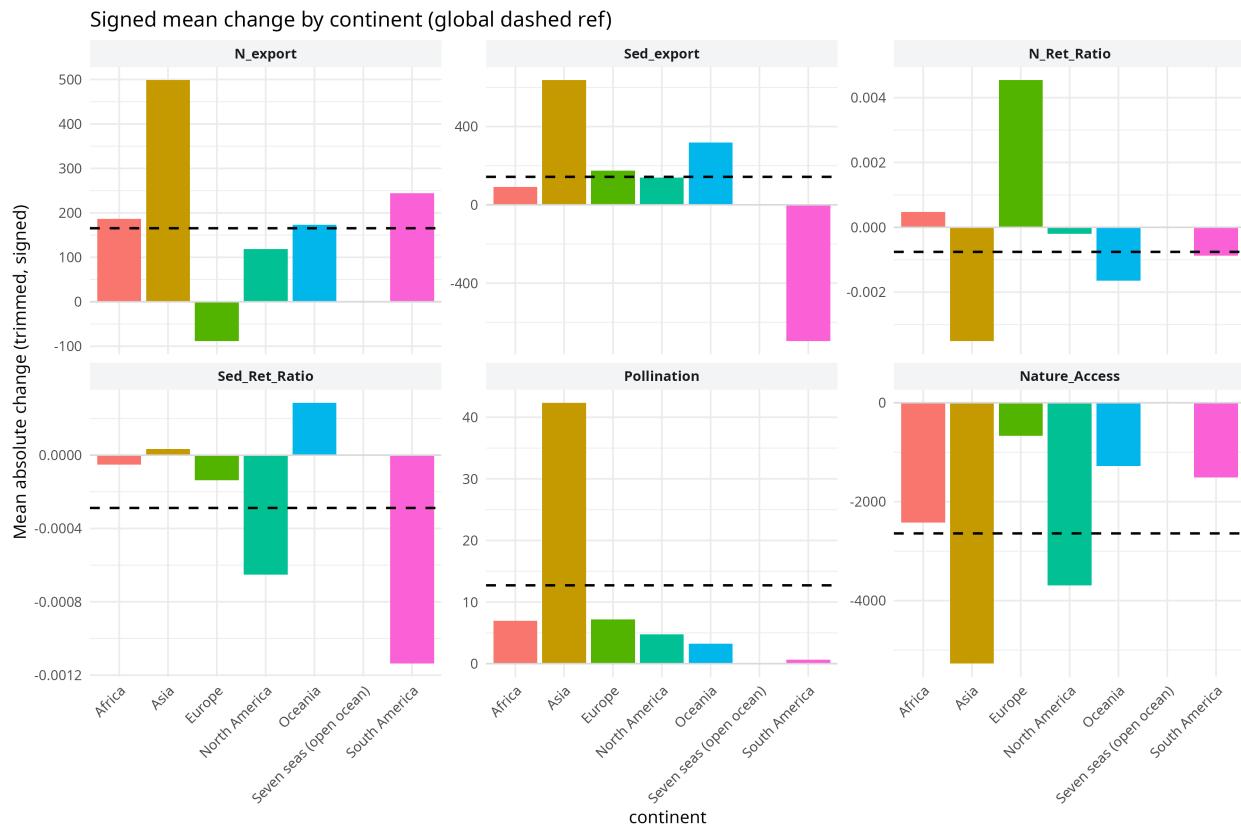
- Each bar shows the **trimmed mean absolute change** ($|\Delta|$) per service within each group; facet axes are free.
- Bars are **always positive** by design: height = **magnitude of change**, not direction.
- Direction-of-concern used elsewhere in the analysis:
 - Worse when **up Sed_export, N_export, C_Risk**.
 - Worse when **down Nature_Access, Pollination, N_Ret_Ratio, Sed_Ret_Ratio, C_Risk_Red_Ratio**.
- These bars answer “**where is change largest?**”. See hotspot maps/violins for **up vs. down** patterns.

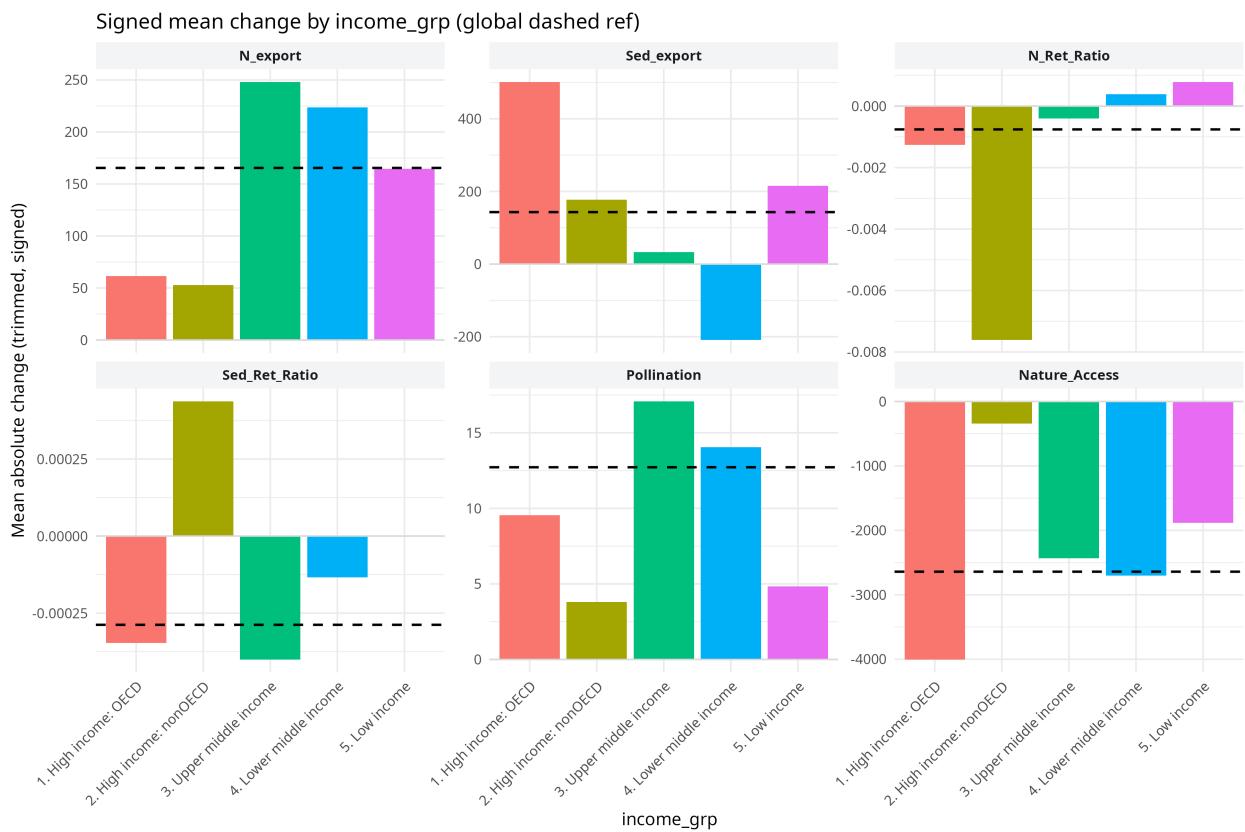
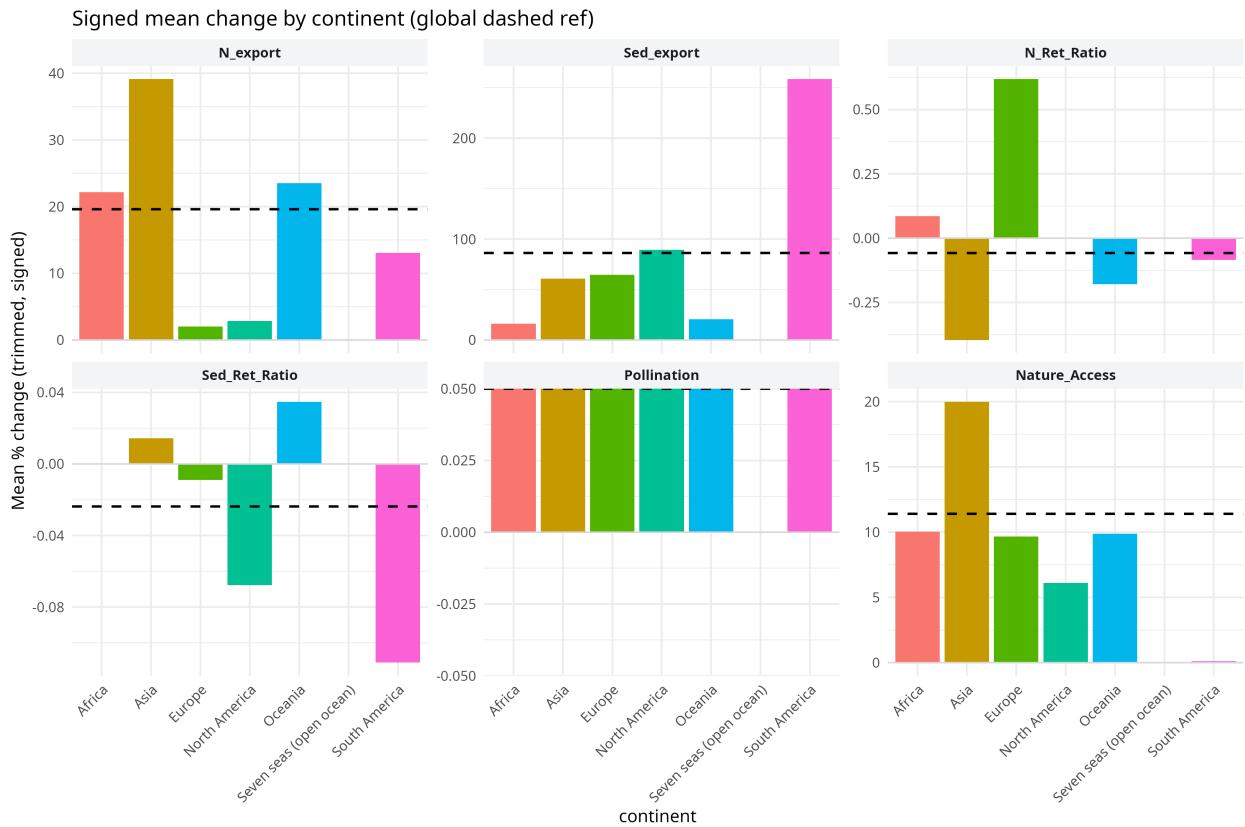
Short answer: your current barplots use all grid cells (the full 10-km population), not just hotspots. They summarize trimmed means per subregion/global from plt_long via aggregate_change_simple(), with cut_q=0.999 to cap outliers and (optionally) drop_zeros=TRUE. That's why every bar is positive—those bars are the magnitude of change, not the direction.

Outputs: PNGs land in outputs/plots/{abs|pct}/<group_col>/bars_*.png plus a flat copy in outputs/plots/latest/bars/ for embedding here. The legacy global-only single-bar chart was removed to keep the gallery focused on grouped views.

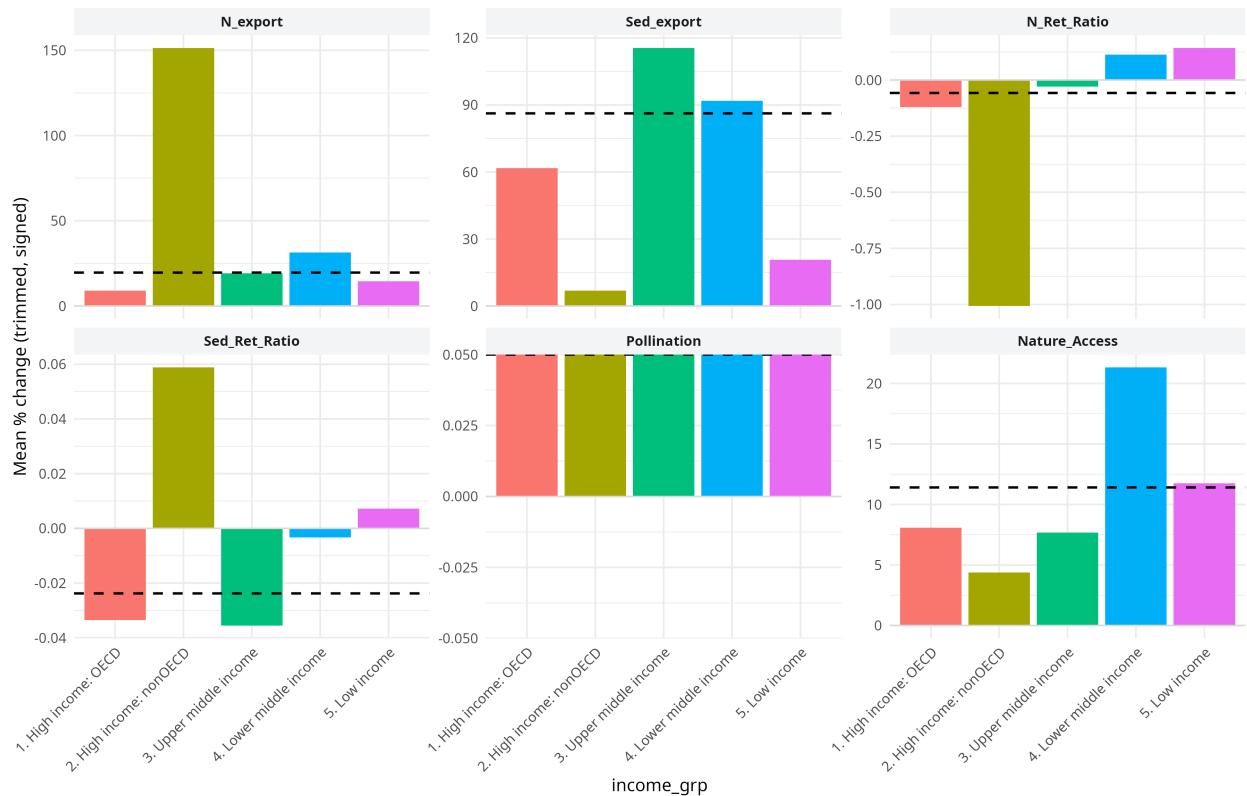
Generate trimmed-mean bar plots (abs & pct) for each grouping. Plots save to outputs/plots/... and are embedded below.

7.1 Signed means (zeros included, trimmed caps)

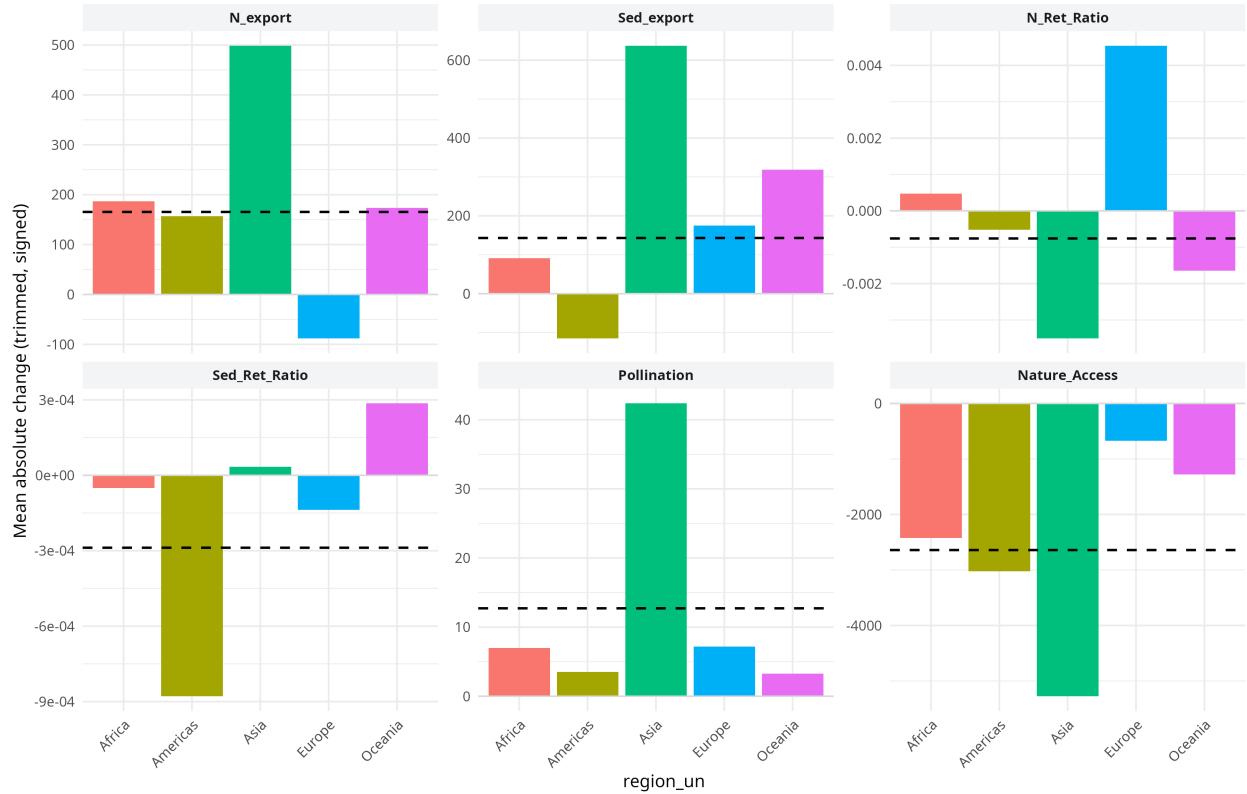




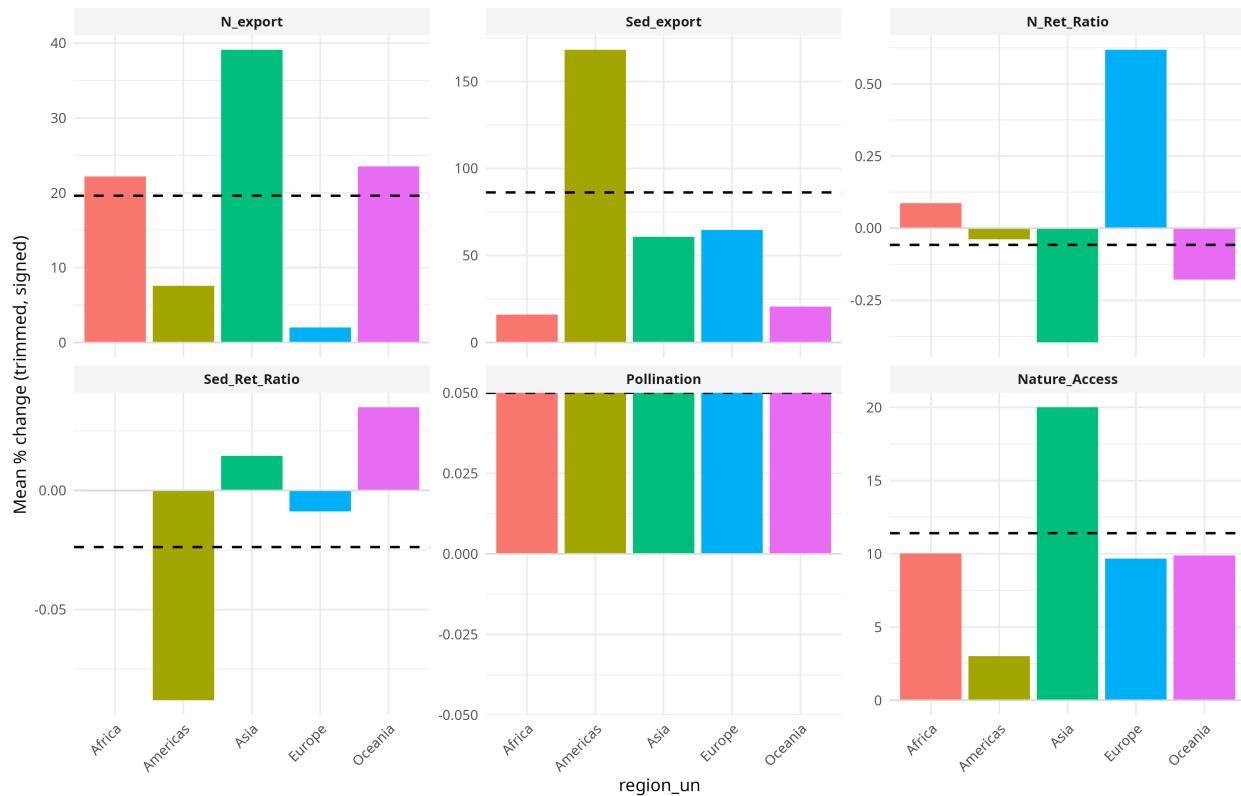
Signed mean change by income_grp (global dashed ref)



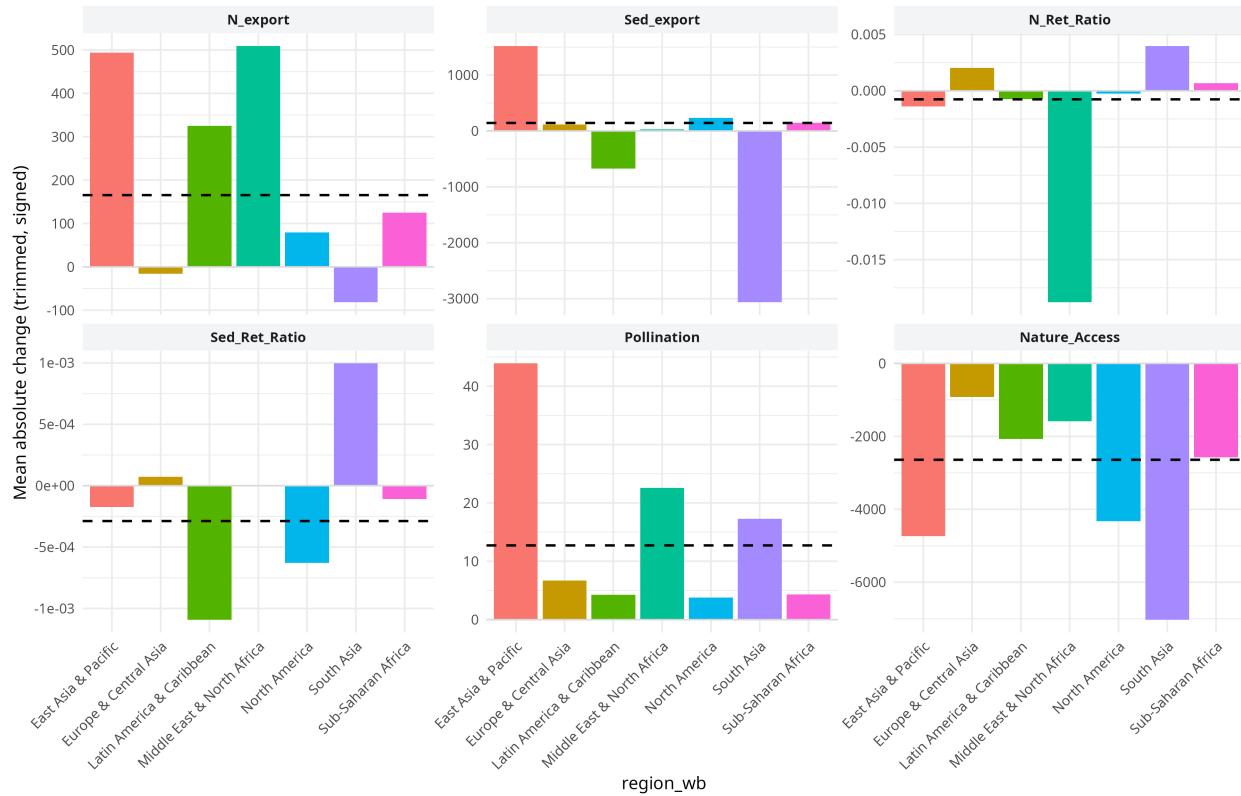
Signed mean change by region_un (global dashed ref)



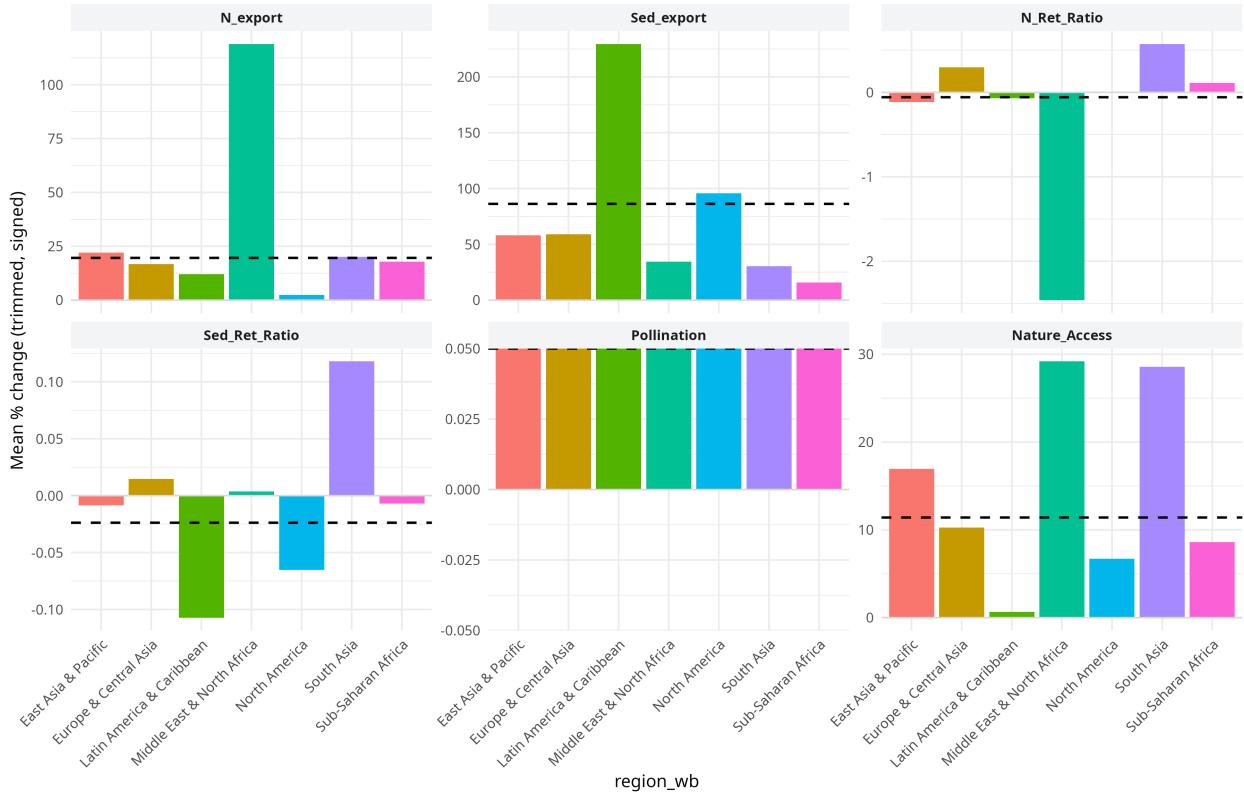
Signed mean change by region_un (global dashed ref)



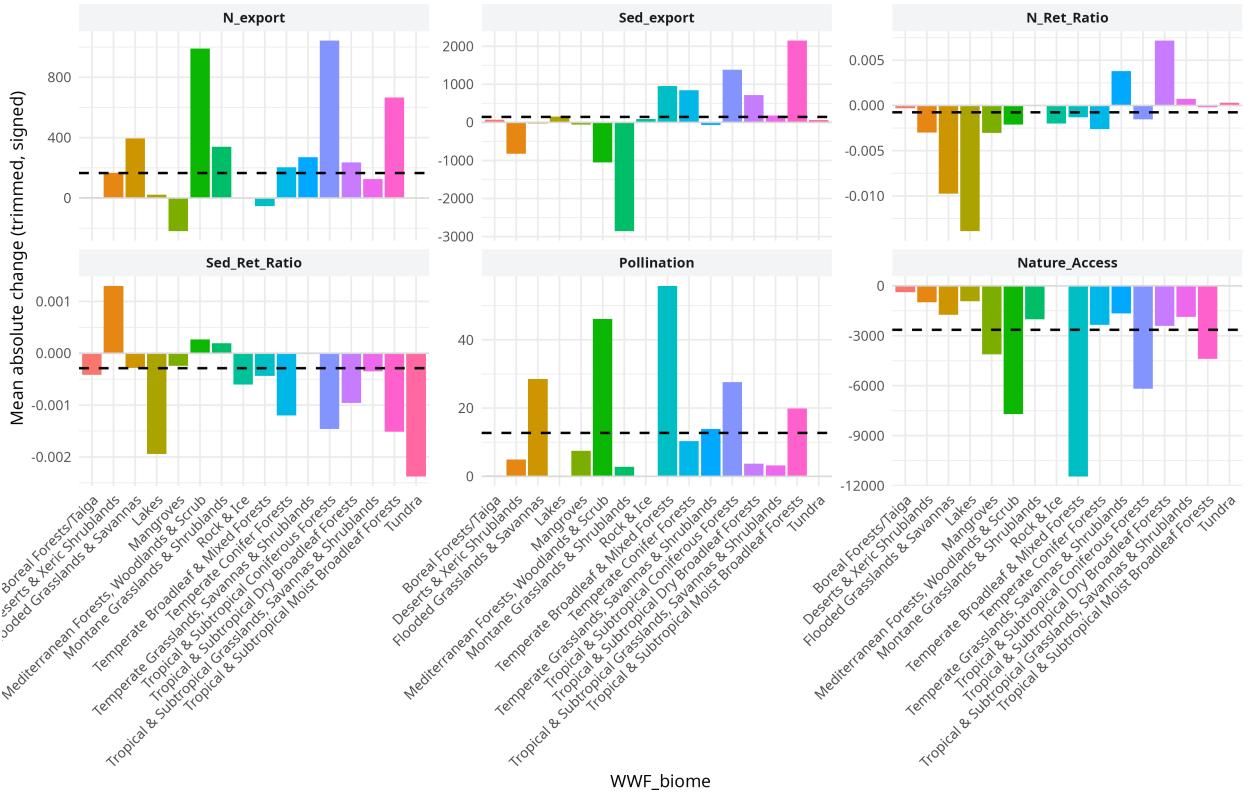
Signed mean change by region_wb (global dashed ref)

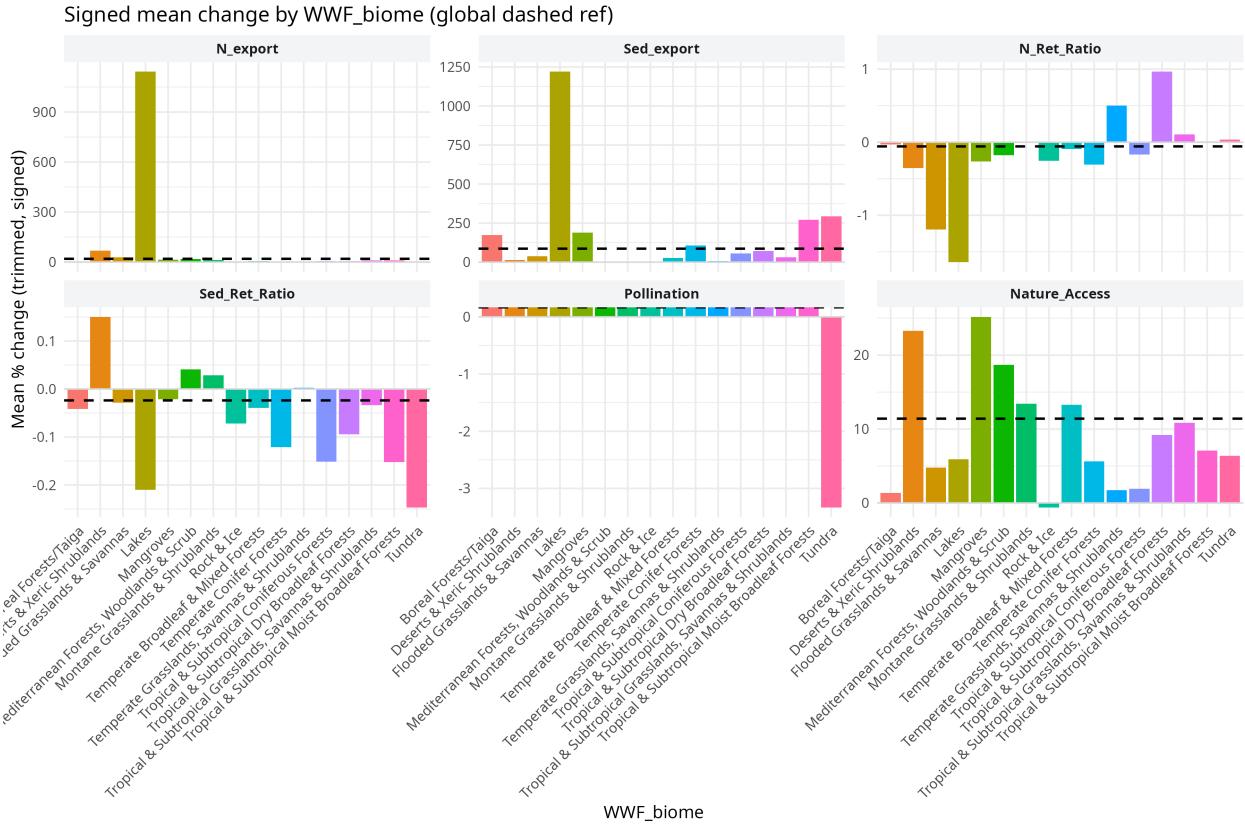


Signed mean change by region_wb (global dashed ref)



Signed mean change by WWF_biome (global dashed ref)



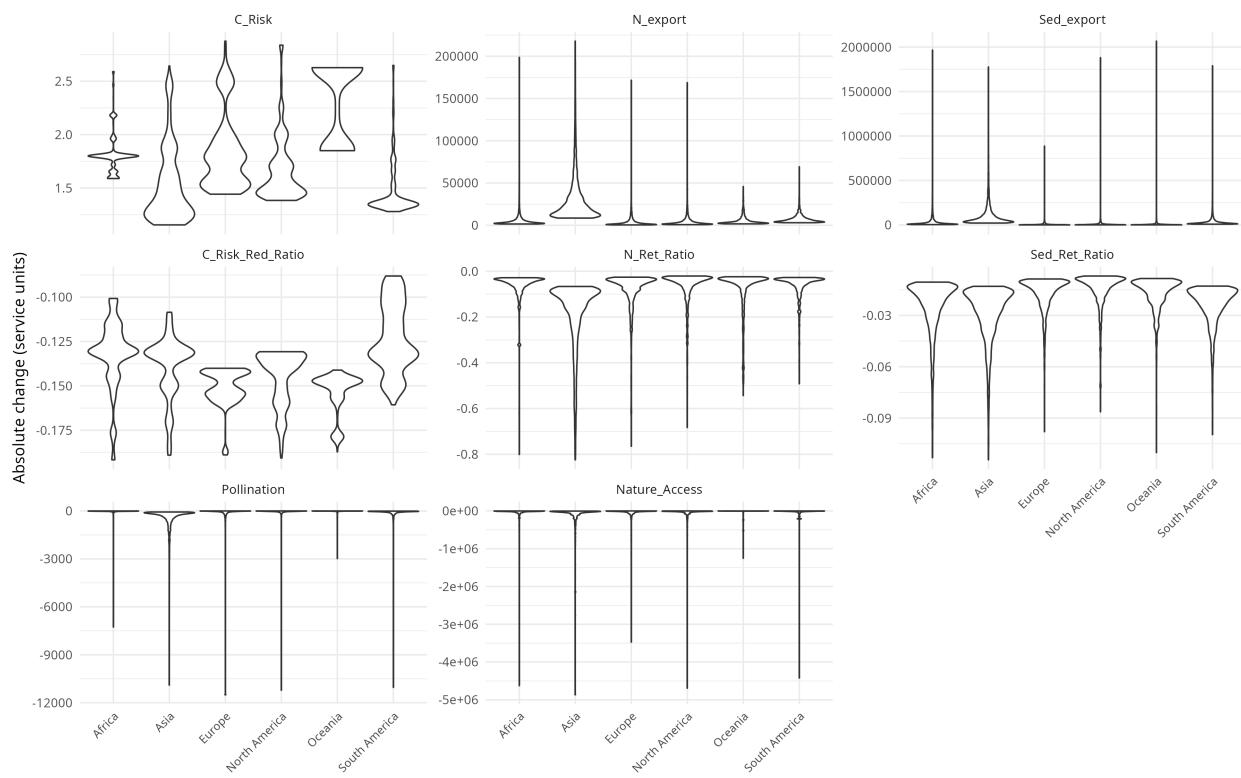


8 Hotspot violin plots

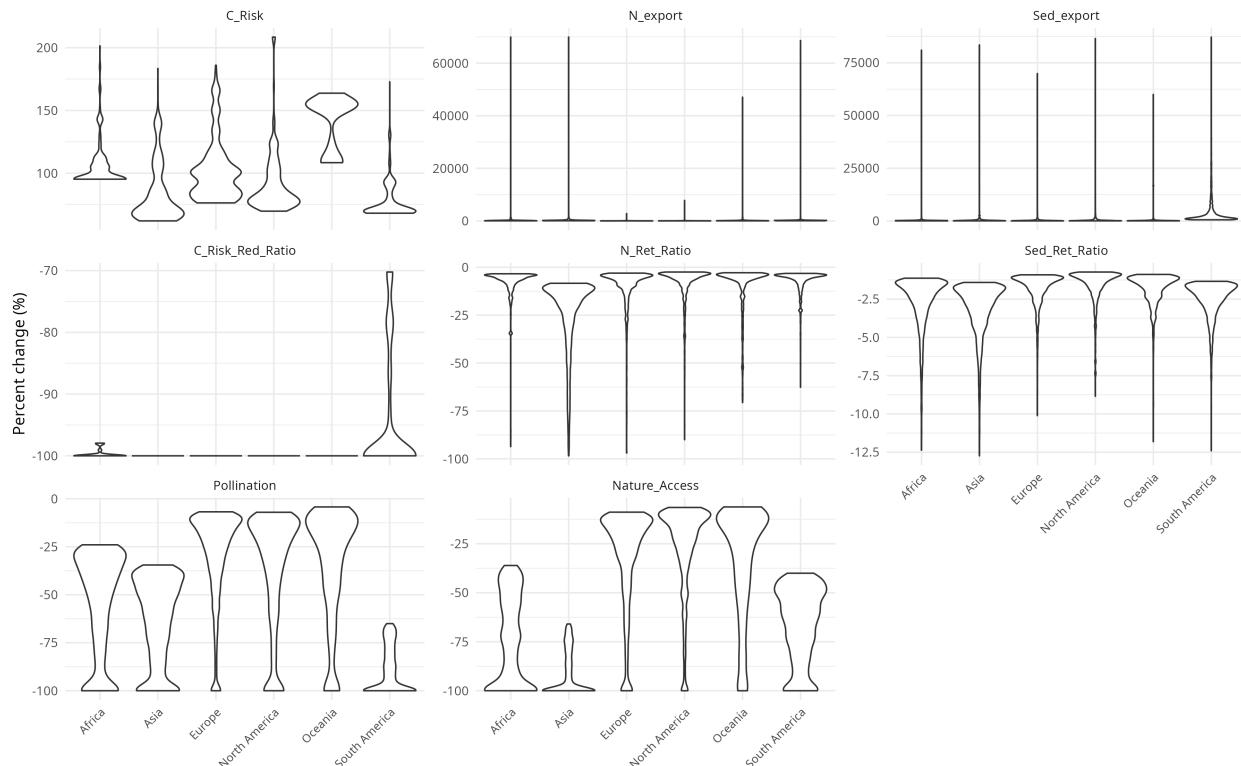
Summarize hotspot distributions by group using saved PNGs; skip computation if group columns are absent. Helper `run_hotspot_violins_by()` now lives in `R/hotspot_violins.R`, so we just call it from this report.

Outputs: PNGs are written to `outputs/plots/{abs|pct}/<group_col>/violins_*.png` and mirrored into `outputs/plots/latest/violins/` for embedding.

Absolute change in hotspots by continent
Violins only · NA/0 removed · per-service 99.9% trim

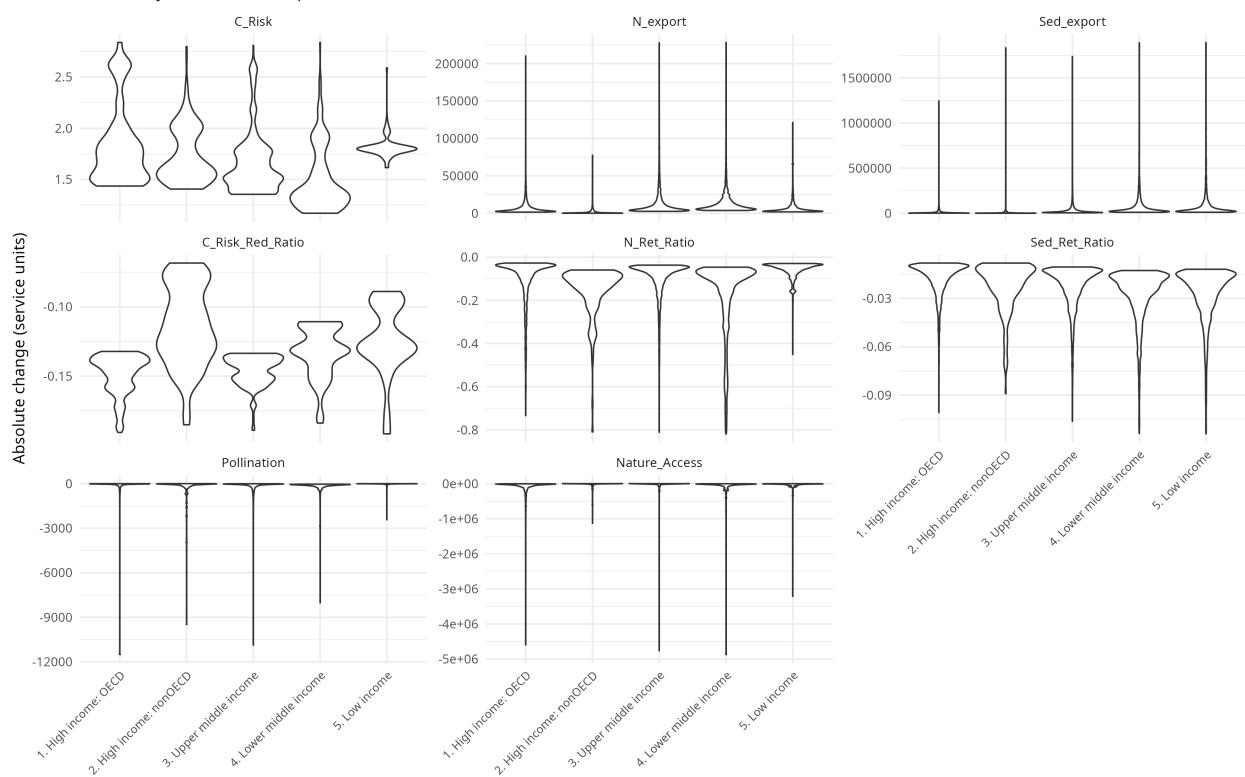


Percent change in hotspots by continent
Violins only · NA/0 removed · per-service 99.9% trim



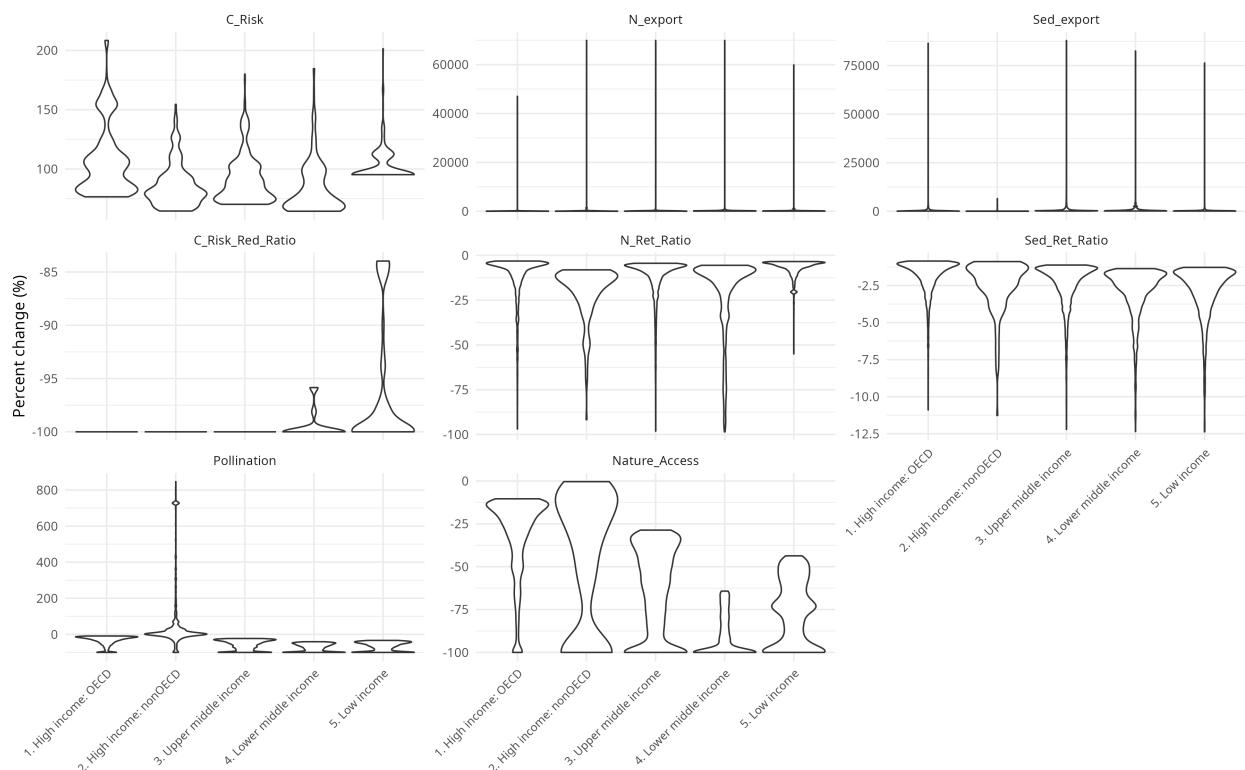
Absolute change in hotspots by income_grp

Violins only · NA/0 removed · per-service 99.9% trim

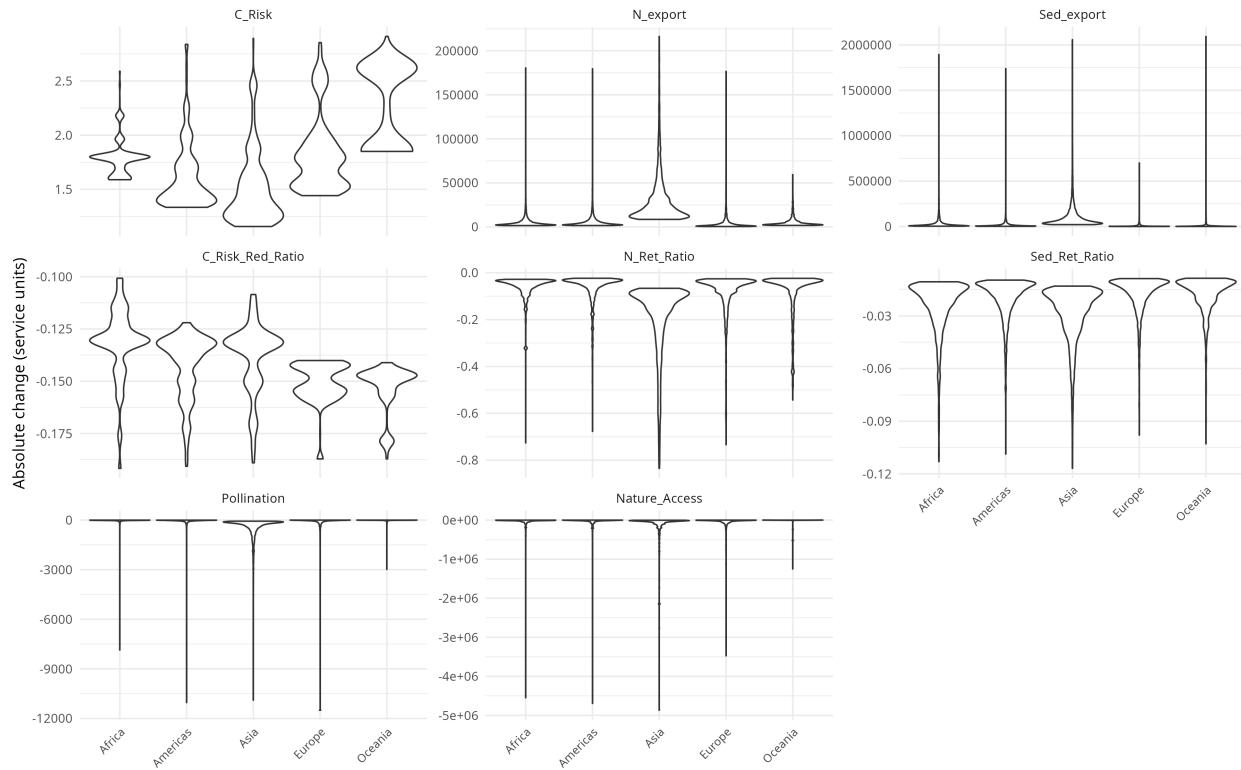


Percent change in hotspots by income_grp

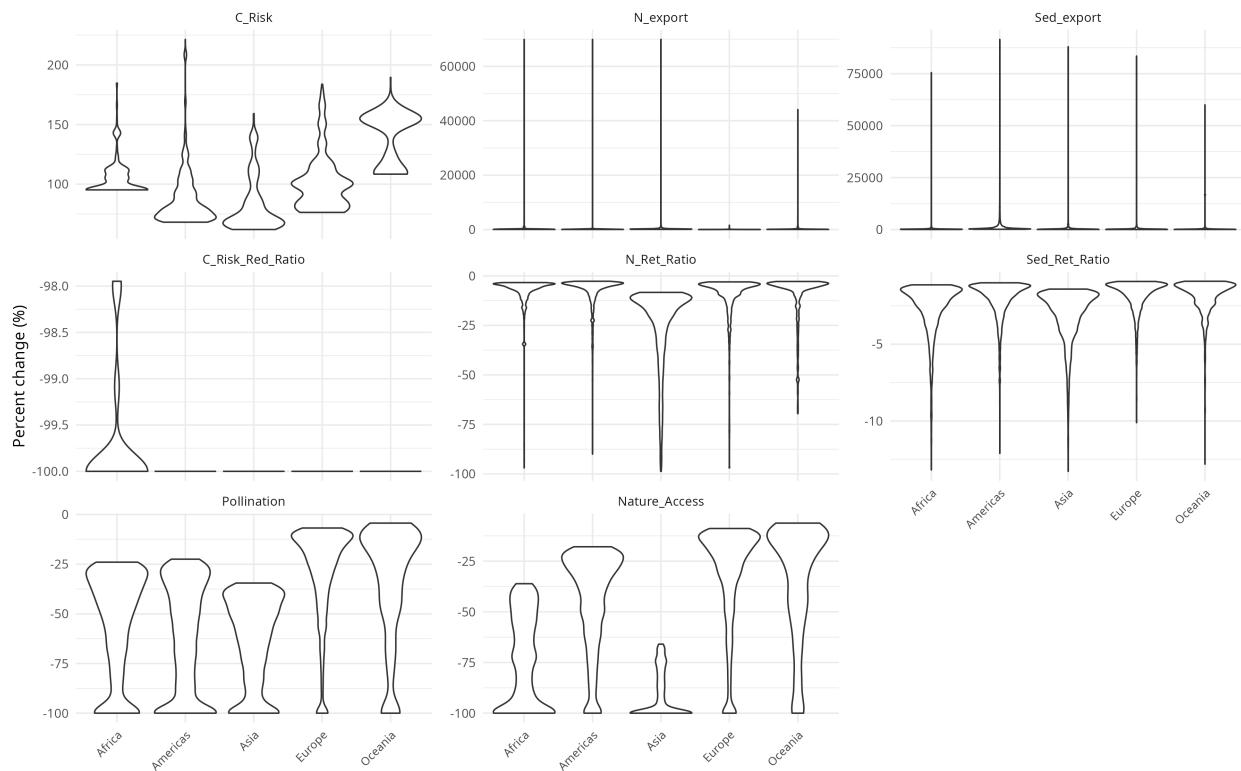
Violins only · NA/0 removed · per-service 99.9% trim



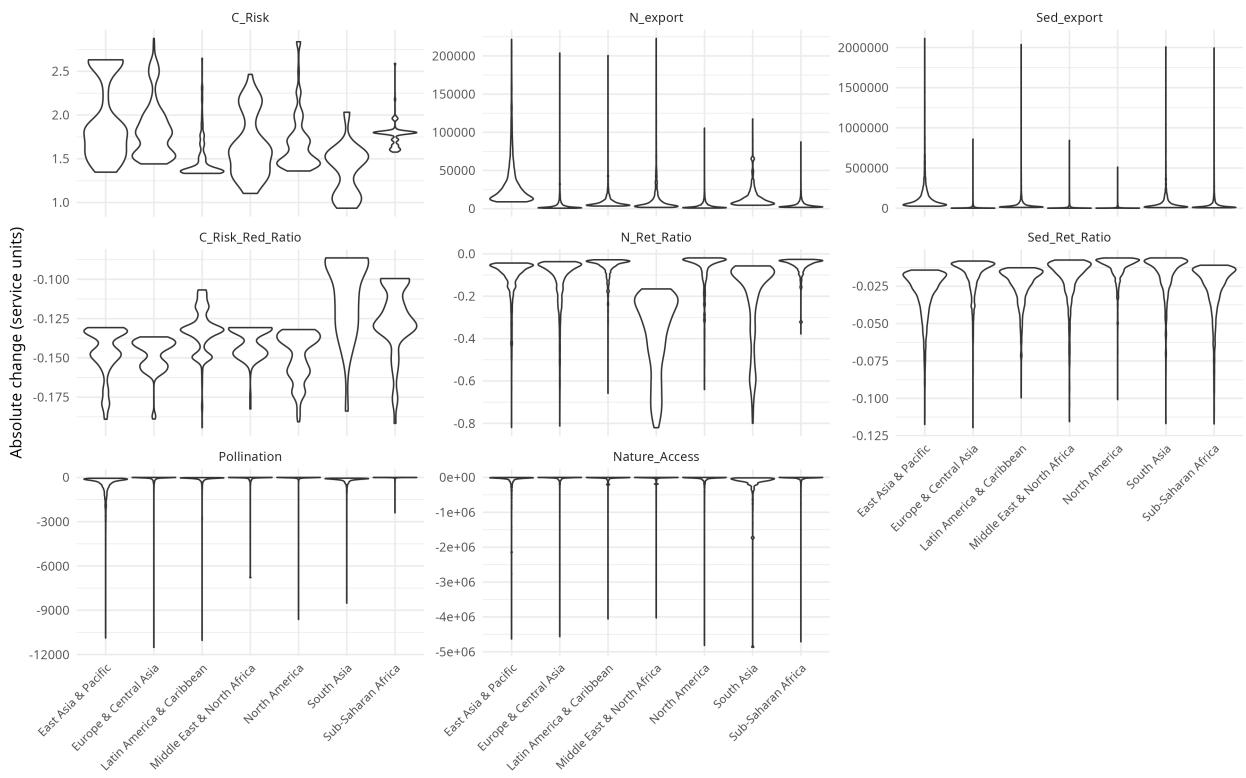
Absolute change in hotspots by region_un
Violins only · NA/0 removed · per-service 99.9% trim



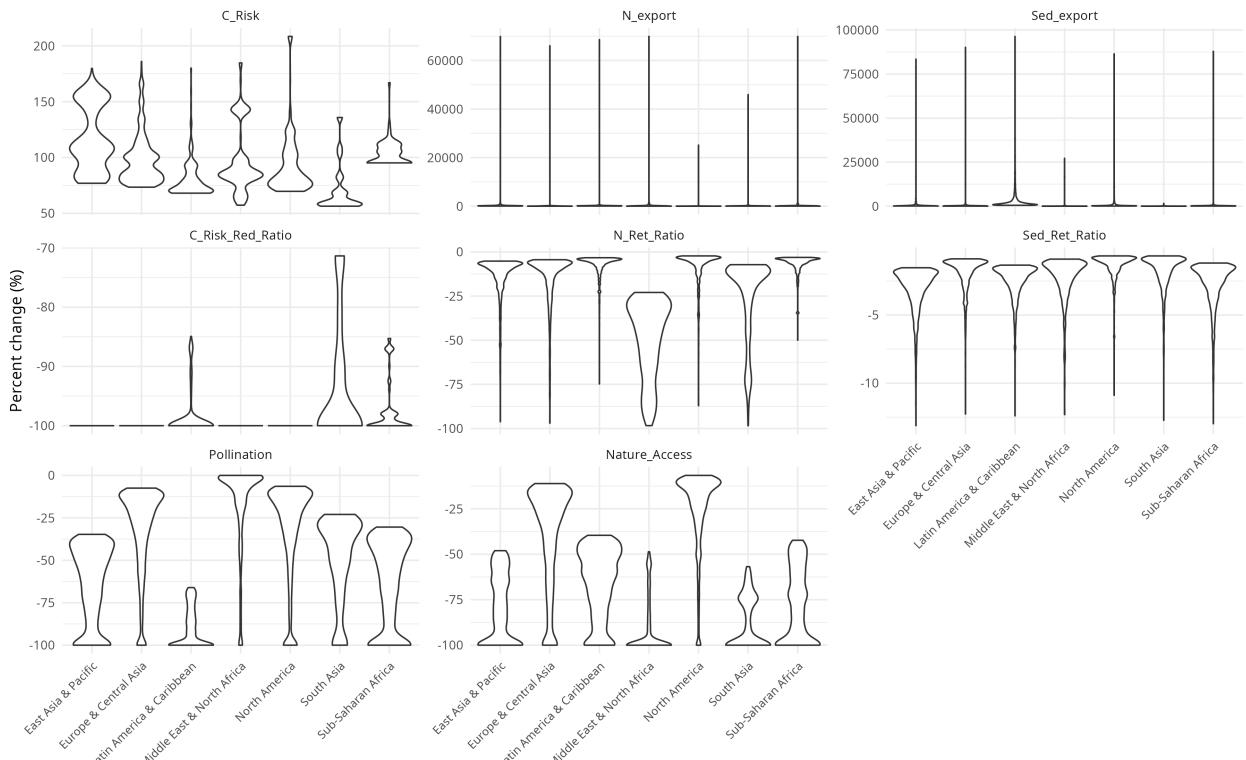
Percent change in hotspots by region_un
Violins only · NA/0 removed · per-service 99.9% trim



Absolute change in hotspots by region_wb
Violins only · NA/0 removed · per-service 99.9% trim

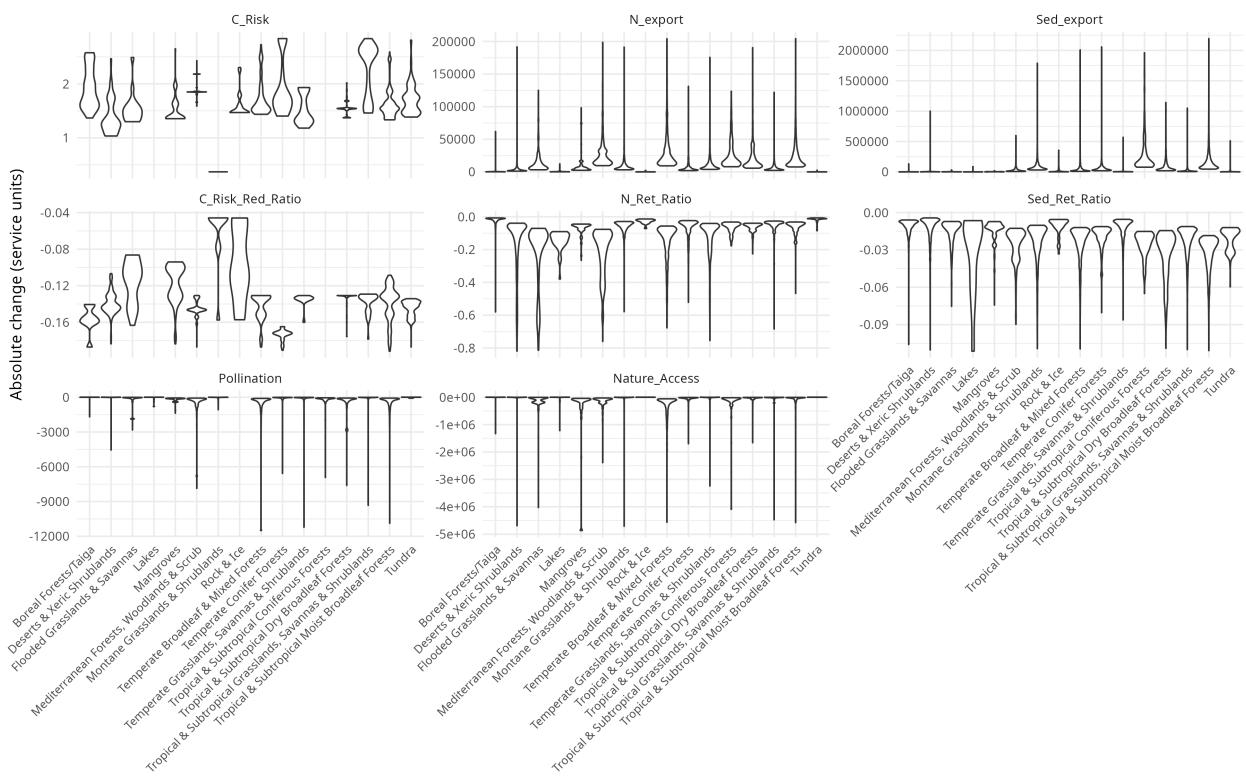


Percent change in hotspots by region_wb
Violins only · NA/0 removed · per-service 99.9% trim



Absolute change in hotspots by WWF biome

Violins only · NA/0 removed · per-service 99.9% trim



Percent change in hotspots by WWF biome

Violins only · NA/0 removed · per-service 99.9% trim

