```
#### Appendix - Full Script
rm(list = ls())

# ------------------------------
# Libraries
# ------------------------------
library(dplyr)
library(knitr)
library(kableExtra)
library(webshot2)
library(ggplot2)
library(patchwork)



# ------------------------------
# Functions
# ------------------------------

# 1) Weight scenario helper (unchanged; keeps names)
make_weights <- function(mode = c("equal", "extreme", "rationale"),
                         crit = NULL,
                         factor = 5,
                         overvalue = 0.3,
                         crit_names) {
  mode <- match.arg(mode)
  if (missing(crit_names) || length(crit_names) < 1) {
    stop("You must supply crit_names (vector of criterion names).")
  }
  n <- length(crit_names)
  w <- rep(1 / n, n)
  names(w) <- crit_names
  if (mode == "equal") return(w)
  if (mode %in% c("extreme", "rationale")) {
    if (is.null(crit)) stop("You must supply 'crit' for modes 'extreme' or 'rationale'.")
    if (is.numeric(crit)) crit <- crit_names[crit]
    if (!all(crit %in% crit_names)) stop("Unknown criteria in crit")
  }
  if (mode == "extreme") {
    base_vals <- rep(1, n); names(base_vals) <- crit_names
    base_vals[crit] <- factor
    return(base_vals / sum(base_vals))
  }
  if (mode == "rationale") {
    base_vals <- rep(1 / n, n); names(base_vals) <- crit_names
    base_vals[crit] <- base_vals[crit] * (1 + overvalue)
    return(base_vals / sum(base_vals))
  }
  stop("Unhandled mode.")
}

# 2) Signed weight vector (applies + / - depending on beneficial vector)
signed_weights <- function(weights, beneficial.vector, ncol) {
  signs <- ifelse(seq_len(ncol) %in% beneficial.vector, +1, -1)
  weights * signs
}

# 3) Compute per-criterion signed contributions (matrix: alternatives x criteria)
compute_contribs <- function(mat, weights, beneficial.vector) {
  sw <- signed_weights(weights, beneficial.vector, ncol(mat))
  contribs <- t(t(mat) * sw)
  rownames(contribs) <- rownames(mat)
  colnames(contribs) <- colnames(mat)
  contribs
}

# 4) Simple MAVT totals (for a matrix)
compute_mavt <- function(mat, weights, beneficial.vector) {
  sw <- signed_weights(weights, beneficial.vector, ncol(mat))
```

```
  t(t(mat) * sw) %>% rowSums()
}

# 5) Identify biggest / smallest contributors (for Lynetteholmen row)
get_lynette_contrib <- function(contrib_matrix) {
  row_lyn <- contrib_matrix[1, ]
  abs_row <- abs(row_lyn)
  total_abs <- sum(abs_row); if (total_abs == 0) total_abs <- 1
  max_idx <- which.max(abs_row); min_idx <- which.min(abs_row)
  biggest <- paste0(colnames(contrib_matrix)[max_idx], " (",
round(100*abs_row[max_idx]/total_abs,1), "%)")
  smallest <- paste0(colnames(contrib_matrix)[min_idx], " (",
round(100*abs_row[min_idx]/total_abs,1), "%)")
  c(biggest = biggest, smallest = smallest)
}

# 6) Color totals helper for kable (keeps your original styling)
color_total <- function(x) {
  max_val <- max(as.numeric(x)); min_val <- min(as.numeric(x))
  sapply(as.numeric(x), function(val) {
    if (val == max_val) {
      cell_spec(round(val,3), "html", background = "#d9f2d9")
    } else if (val == min_val) {
      cell_spec(round(val,3), "html", background = "#f2d9d9")
    } else {
      cell_spec(round(val,3), "html")
    }
  })
}

# 7) Tornado plot (accepts matrix OR named numeric vector). Negative = blue, Positive = red.
plot_tornado_percent <- function(obj, title_text = "Tornado diagram") {
  if (is.matrix(obj) || is.data.frame(obj)) {
    mat <- as.matrix(obj)
    if ("Lynetteholmen" %in% rownames(mat)) {
      rowvec <- as.numeric(mat["Lynetteholmen", ])
    } else {
      rowvec <- as.numeric(mat[1, ])
    }
    crit_names <- colnames(mat)
  } else if (is.numeric(obj) && !is.null(names(obj))) {
    rowvec <- as.numeric(obj)
    crit_names <- names(obj)
  } else stop("Please pass a per-criterion contribution matrix or a named numeric vector.")
  total_abs <- sum(abs(rowvec)); if (total_abs == 0) total_abs <- 1
  pct_contrib <- 100 * rowvec / total_abs
  df <- data.frame(Criterion = crit_names, ContributionPct = pct_contrib, stringsAsFactors =
FALSE) %>%
    mutate(AbsContributionPct = abs(ContributionPct)) %>%
    arrange(desc(AbsContributionPct))
  ggplot(df, aes(x = reorder(Criterion, AbsContributionPct), y = AbsContributionPct, fill =
ContributionPct > 0)) +
    geom_col() + coord_flip() +
    scale_fill_manual(values = c("FALSE" = "steelblue", "TRUE" = "firebrick"),
                      labels = c("Negative", "Positive"), name = "Direction") +
    labs(title = title_text, x = "Criterion", y = "Contribution (%)") +
    theme_minimal(base_size = 13)
}

# 8) Perturbation helpers:
# - perturb_scores: Economic Effects (col 1) varies only positively (1 -> 1+max_dev),
#   other criteria vary symmetrically (1-max_dev -> 1+max_dev).
# - perturb_weights: all weights vary symmetrically and renormalize.
perturb_scores <- function(base_scores, max_dev = 0.2, econ_index = 1) {
  n <- length(base_scores)
  multipliers <- runif(n, 1 - max_dev, 1 + max_dev)
  multipliers[econ_index] <- runif(1, 1, 1 + max_dev)  # economic effects only positive
  base_scores * multipliers
}
```

```
perturb_weights <- function(base_weights, max_dev = 0.2) {
  multipliers <- runif(length(base_weights), 1 - max_dev, 1 + max_dev)
  w <- base_weights * multipliers
  w / sum(w)
}

# 9) Monte Carlo that perturbs scores AND weights, returns totals
monte_carlo_totals_scores_weights <- function(raw_scores, base_weights, beneficial.vector,
                                              alt_name = "Lynetteholmen",
                                              n_iter = N_ITER, max_dev = 0.2, econ_index = 1) {
  n_crit <- ncol(raw_scores)
  totals <- numeric(n_iter)
  for (i in seq_len(n_iter)) {
    pert_scores <- perturb_scores(raw_scores[alt_name, ], max_dev = max_dev, econ_index =
econ_index)
    pert_w <- perturb_weights(base_weights, max_dev = max_dev)
    sw <- signed_weights(pert_w, beneficial.vector, n_crit)
    totals[i] <- sum(pert_scores * sw)
  }
  totals
}
monte_carlo_prob <- function(totals) mean(totals > 0) * 100

# 10) Compute mean per-criterion signed contributions under MC (for tornado plots)
mean_mc_contribs <- function(raw_scores, base_weights, beneficial.vector, max_dev = 0.2, n_iter
= N_ITER, econ_index = 1) {
  n_crit <- ncol(raw_scores)
  contribs_matrix <- matrix(NA, nrow = n_iter, ncol = n_crit)
  colnames(contribs_matrix) <- colnames(raw_scores)
  for (i in seq_len(n_iter)) {
    pert_scores <- perturb_scores(raw_scores["Lynetteholmen", ], max_dev = max_dev, econ_index
= econ_index)
    pert_w <- perturb_weights(base_weights, max_dev = max_dev)
    sw <- signed_weights(pert_w, beneficial.vector, n_crit)
    contribs_matrix[i, ] <- pert_scores * sw
  }
  # return named numeric vector (mean contribution per criterion)
  res <- colMeans(contribs_matrix)
  names(res) <- colnames(raw_scores)
  res
}


# ------------------------------
# 6 Criteria Analysis
# ------------------------------

crit_names <- c("Economic Effects", "Traffic", "Conservation", "Environmental Effects", "Health
and Wellbeing", "Resource Use")
alt_names <- c("Lynetteholmen", "0-Scenario")
beneficial.vector <- c(1,2)  # indices of beneficial criteria
N_ITER <- 50000   # Monte Carlo iterations
SAVE_DPI <- 300

#weights <- make_weights("equal", crit_names = crit_names)
weights <- make_weights("rationale", crit = c("Economic Effects", "Traffic"), overvalue = 3,
crit_names = crit_names)
#weights <- make_weights("extreme", crit = "Economic Effects", factor = 5, crit_names =
crit_names)
title <- "6 - Economic Rationale 300 PCT."

# ------------------------------
# Input data
# ------------------------------
# Score matrices: two alternatives (Lynetteholmen, 0-Scenario)
simple_scores <- matrix(c(1,5,6,11,4,5, 0,0,0,0,0,0), nrow = 2, byrow = TRUE)
ln_simple_scores <- matrix(c(0.69,1.79,1.95,2.48,1.61,1.79,  0,0,0,0,0,0), nrow = 2, byrow =
TRUE)
proportional_scores <- matrix(c(0.14,1,1.2,1.38,0.67,1, 0,0,0,0,0,0), nrow = 2, byrow = TRUE)
```

```
ln_proportional_scores <- matrix(c(0.13,0.69,0.79,0.87,0.51,0.69, 0,0,0,0,0,0), nrow = 2, byrow
= TRUE)

# set names
rownames(simple_scores) <- alt_names; colnames(simple_scores) <- crit_names
rownames(ln_simple_scores) <- alt_names; colnames(ln_simple_scores) <- crit_names
rownames(proportional_scores) <- alt_names; colnames(proportional_scores) <- crit_names
rownames(ln_proportional_scores) <- alt_names; colnames(ln_proportional_scores) <- crit_names

# bundle score sets
score_sets <- list(
  "Assigned" = simple_scores,
  "ln(Assigned)" = ln_simple_scores,
  "Proportional" = proportional_scores,
  "ln(Proportional)" = ln_proportional_scores
)

# ------------------------------
# Unperturbed computations
# ------------------------------
simple_contribs <- compute_contribs(simple_scores, weights, beneficial.vector)
ln_simple_contribs <- compute_contribs(ln_simple_scores, weights, beneficial.vector)
proportional_contribs <- compute_contribs(proportional_scores, weights, beneficial.vector)
ln_proportional_contribs <- compute_contribs(ln_proportional_scores, weights,
beneficial.vector)

manual_simple_scores <- rowSums(simple_contribs)["Lynetteholmen"]
manual_ln_simple_scores <- rowSums(ln_simple_contribs)["Lynetteholmen"]
manual_proportional_scores <- rowSums(proportional_contribs)["Lynetteholmen"]
manual_ln_proportional_scores <- rowSums(ln_proportional_contribs)["Lynetteholmen"]

# top / bottom contributors (unperturbed)
simple_contrib_info <- get_lynette_contrib(simple_contribs)
ln_simple_contrib_info <- get_lynette_contrib(ln_simple_contribs)
prop_contrib_info <- get_lynette_contrib(proportional_contribs)
ln_prop_contrib_info <- get_lynette_contrib(ln_proportional_contribs)

# ------------------------------
# Monte Carlo
# ------------------------------
# 20%
mc_totals_20 <- lapply(score_sets, monte_carlo_totals_scores_weights,
                       base_weights = weights, beneficial.vector = beneficial.vector,
                       n_iter = N_ITER, max_dev = 0.2, alt_name = "Lynetteholmen", econ_index =
1)
# 50%
mc_totals_50 <- lapply(score_sets, monte_carlo_totals_scores_weights,
                       base_weights = weights, beneficial.vector = beneficial.vector,
                       n_iter = N_ITER, max_dev = 0.5, alt_name = "Lynetteholmen", econ_index =
1)

# probs
prob20 <- round(sapply(mc_totals_20, function(x) mean(x > 0) * 100), 1)
prob50 <- round(sapply(mc_totals_50, function(x) mean(x > 0) * 100), 1)

# ------------------------------
# Build combined table
# ------------------------------
cols <- names(score_sets)

# Row 1: Scores (unperturbed Lynetteholmen totals)
scores_row <- c(manual_simple_scores, manual_ln_simple_scores,
                manual_proportional_scores, manual_ln_proportional_scores)
names(scores_row) <- cols

# Row 2: Top contributor (unperturbed)
top_row <- c(simple_contrib_info["biggest"], ln_simple_contrib_info["biggest"],
             prop_contrib_info["biggest"], ln_prop_contrib_info["biggest"])
names(top_row) <- cols
```

```
# Row 3: Bottom contributor (unperturbed)
bottom_row <- c(simple_contrib_info["smallest"], ln_simple_contrib_info["smallest"],
                prop_contrib_info["smallest"], ln_prop_contrib_info["smallest"])
names(bottom_row) <- cols

# Row 4: Monte Carlo 20% probabilities
mc20_row <- paste0(prob20, " %")
names(mc20_row) <- cols

# Row 5: Monte Carlo 50%
mc50_row <- paste0(prob50, " %")
names(mc50_row) <- cols

# Row 6: Empty spacer
empty_row <- rep(" ", length(cols))
names(empty_row) <- cols

# ------------------------------------------------------------
# Row 7 and Row 8:
# ------------------------------------------------------------

# Positive criteria (beneficial)
pos_idx <- c(1,2)
pos_names <- crit_names[pos_idx]
pos_vals <- sprintf("%.3f", weights[pos_idx])
pos_w_row <- paste0(pos_names, " = ", pos_vals)

pos_w_row_full <- rep(" ", length(cols))       # start with empty cells
pos_w_row_full[1:length(pos_w_row)] <- pos_w_row   # fill first 2 columns

# Negative criteria (non-beneficial)
neg_idx <- c(3,4,5,6)
neg_names <- crit_names[neg_idx]
neg_vals <- sprintf("%.3f", weights[neg_idx])
neg_w_row <- paste0(neg_names, " = ", neg_vals)

neg_w_row_full <- rep(" ", length(cols))      # start with empty cells
neg_w_row_full[1:length(neg_w_row)] <- neg_w_row   # place in first 4 columns

# ------------------------------
# Combine into a table
# ------------------------------
combined <- rbind(
  Score = scores_row,
  Top_Contributor = top_row,
  Bottom_Contributor = bottom_row,
  Spacer1 = empty_row,
  MonteCarlo_20pct = mc20_row,
  MonteCarlo_50pct = mc50_row,
  Spacer2 = empty_row,
  Positive_Weights = pos_w_row_full,
  Negative_Weights = neg_w_row_full
)



# Convert to data.frame and transpose so that columns = analyses, rows = requested items
combined_df <- as.data.frame(combined, stringsAsFactors = FALSE)

# Build display table: transform Score row into formatted (colored) HTML cells
score_vals <- as.numeric(as.character(combined_df["Score", ]))
score_cells <- sapply(score_vals, function(v) {
  if (is.na(v)) return(cell_spec("", "html"))
  if (v > 0) {
    cell_spec(round(v,3), "html", background = "#d9f2d9")   # green
  } else if (v < 0) {
    cell_spec(round(v,3), "html", background = "#f2d9d9")   # red
  } else {
```

```
    cell_spec(round(v,3), "html")
  }
})

display_df <- combined_df
display_df["Score", ] <- score_cells

# Replace underscores in rownames for nicer display
rownames(display_df) <- c("Score", "Top Contributor", "Bottom Contributor", " ",
                          "Monte Carlo 20 %", "Monte Carlo 50 %", "",
                          "Positive Weights",
                          "Negative Weights")

# ------------------------------
# Styled output table
# ------------------------------

prob_table_kbl <- kbl(display_df, caption = title, align = "c", row.names = TRUE, escape =
FALSE) %>%
  kable_styling(full_width = FALSE, bootstrap_options = c("striped","hover","condensed")) %>%
  row_spec(0, background = "#DFF2F2") %>%                          # header
  row_spec(nrow(display_df) - 1, background = "#DFF2F2") %>%     # Positive Weights
  row_spec(nrow(display_df), background = "#DFF2F2") %>%         # Negative Weights
  column_spec(1, border_right = TRUE)

# Save HTML → PNG
tmp_html <- tempfile(fileext = ".html")
save_kable(prob_table_kbl, file = tmp_html)

if (requireNamespace("rstudioapi", quietly = TRUE) && rstudioapi::isAvailable()) {
  script_dir <- dirname(rstudioapi::getActiveDocumentContext()$path)
} else {
  script_dir <- getwd()
}

combined_table_file <- file.path(script_dir, paste0(title, " - Combined Table.png"))
webshot(tmp_html, file = combined_table_file, vwidth = 1200, vheight = 900, zoom = 2)

cat("Saved combined table to:", combined_table_file, "\n")

# ------------------------------
# Tornado Plot
# ------------------------------
p1 <- plot_tornado_percent(simple_contribs, "Assigned")
p2 <- plot_tornado_percent(ln_simple_contribs, "ln(Assigned)")
p3 <- plot_tornado_percent(proportional_contribs, "Proportional")
p4 <- plot_tornado_percent(ln_proportional_contribs, "ln(Proportional)")

combined_percent_plot <- (p1 | p2) / (p3 | p4) + plot_annotation(title = title)

if (requireNamespace("rstudioapi", quietly = TRUE) && rstudioapi::isAvailable()) {
  script_dir <- dirname(rstudioapi::getActiveDocumentContext()$path)
} else {
  script_dir <- getwd()
}

# Save unperturbed outputs
unperturbed_tornado_file <- file.path(script_dir, paste0(title, " - Constant Tornado.png"))

# save tornado
ggsave(unperturbed_tornado_file, combined_percent_plot, width = 12, height = 8, dpi = SAVE_DPI)
cat("Saved unperturbed tornado to:", unperturbed_tornado_file, "\n")


# ------------------------------
# MC Histograms
# ------------------------------
hist_plots_20 <- mapply(function(x,nm) {
  ggplot(data.frame(score = x), aes(x = score)) +
```

```
    geom_histogram(bins = 50, fill = "#3B9C9C", color = "black") +
    labs(title = paste0(nm, " – 20% MC Histogram"), x = "Total Score", y = "Frequency") +
    theme_minimal()
}, mc_totals_20, names(mc_totals_20), SIMPLIFY = FALSE)

hist_plots_50 <- mapply(function(x,nm) {
  ggplot(data.frame(score = x), aes(x = score)) +
    geom_histogram(bins = 50, fill = "#3B9C9C", color = "black") +
    labs(title = paste0(nm, " – 50% MC Histogram"), x = "Total Score", y = "Frequency") +
    theme_minimal()
}, mc_totals_50, names(mc_totals_50), SIMPLIFY = FALSE)

#Combine Histograms
combined_hist_20 <- (hist_plots_20[[1]] | hist_plots_20[[2]]) / (hist_plots_20[[3]] |
hist_plots_20[[4]]) +
  plot_annotation(title = paste0(title, " – 20% Monte Carlo Histograms"))

combined_hist_50 <- (hist_plots_50[[1]] | hist_plots_50[[2]]) / (hist_plots_50[[3]] |
hist_plots_50[[4]]) +
  plot_annotation(title = paste0(title, " – 50% Monte Carlo Histograms"))

combined_hist_20_file <- file.path(script_dir, paste0(title, " - MC Histograms 20.png"))
combined_hist_50_file <- file.path(script_dir, paste0(title, " - MC Histograms 50.png"))

ggsave(combined_hist_20_file, combined_hist_20, width = 12, height = 8, dpi = SAVE_DPI)
ggsave(combined_hist_50_file, combined_hist_50, width = 12, height = 8, dpi = SAVE_DPI)

cat("Saved combined MC plots:\n", combined_hist_20_file, "\n", combined_hist_50_file, "\n")

cat("All done — outputs saved in:", script_dir, "\n")



# ------------------------------
# 8 Criteria Analysis
# ------------------------------

crit_names <- c("Economic Effects", "Traffic", "Soil Deposit", "Storm Surge",
                "Conservation", "Environmental Effects", "Health and Wellbeing", "Resource
Use")
alt_names <- c("Lynetteholmen", "0-Scenario")
beneficial.vector <- c(1,2,3,4)  # indices of beneficial criteria
N_ITER <- 50000   # Monte Carlo iterations
SAVE_DPI <- 300

#weights <- make_weights("equal", crit_names = crit_names)
weights <- make_weights("rationale", crit = c("Economic Effects", "Traffic"), overvalue = 3,
crit_names = crit_names)
#weights <- make_weights("extreme", crit = "Storm Surge", factor = 5, crit_names = crit_names)
title <- "8 - Economic Rationale 300 PCT."

# ------------------------------
# Input data
# ------------------------------
# Score matrices: two alternatives (Lynetteholmen, 0-Scenario)
simple_scores <- matrix(c(1,5,5,5,9,11,4,8,  0,0,0,0,0,0,0,0), nrow = 2, byrow = TRUE)
ln_simple_scores <- matrix(c(0.69,1.79,1.79,1.79,2.3,2.48,1.61,2.2,  0,0,0,0,0,0,0,0), nrow =
2, byrow = TRUE)
proportional_scores <- matrix(c(0.14,1,1,1,2.25,1.38,0.67,2, 0,0,0,0,0,0,0,0), nrow = 2, byrow
= TRUE)
ln_proportional_scores <- matrix(c(0.13,0.69,0.69,0.69,1.18,0.87,0.51,1.1, 0,0,0,0,0,0,0,0),
nrow = 2, byrow = TRUE)

# set names
rownames(simple_scores) <- alt_names; colnames(simple_scores) <- crit_names
rownames(ln_simple_scores) <- alt_names; colnames(ln_simple_scores) <- crit_names
rownames(proportional_scores) <- alt_names; colnames(proportional_scores) <- crit_names
rownames(ln_proportional_scores) <- alt_names; colnames(ln_proportional_scores) <- crit_names
```

```r
# bundle score sets
score_sets <- list(
  "Assigned" = simple_scores,
  "ln(Assigned)" = ln_simple_scores,
  "Proportional" = proportional_scores,
  "ln(Proportional)" = ln_proportional_scores
)

# ------------------------------
# Unperturbed computations
# ------------------------------
simple_contribs <- compute_contribs(simple_scores, weights, beneficial.vector)
ln_simple_contribs <- compute_contribs(ln_simple_scores, weights, beneficial.vector)
proportional_contribs <- compute_contribs(proportional_scores, weights, beneficial.vector)
ln_proportional_contribs <- compute_contribs(ln_proportional_scores, weights,
beneficial.vector)

manual_simple_scores <- rowSums(simple_contribs)["Lynetteholmen"]
manual_ln_simple_scores <- rowSums(ln_simple_contribs)["Lynetteholmen"]
manual_proportional_scores <- rowSums(proportional_contribs)["Lynetteholmen"]
manual_ln_proportional_scores <- rowSums(ln_proportional_contribs)["Lynetteholmen"]

# top / bottom contributors (unperturbed)
simple_contrib_info <- get_lynette_contrib(simple_contribs)
ln_simple_contrib_info <- get_lynette_contrib(ln_simple_contribs)
prop_contrib_info <- get_lynette_contrib(proportional_contribs)
ln_prop_contrib_info <- get_lynette_contrib(ln_proportional_contribs)

# ------------------------------
# Monte Carlo
# ------------------------------
# 20%
mc_totals_20 <- lapply(score_sets, monte_carlo_totals_scores_weights,
                       base_weights = weights, beneficial.vector = beneficial.vector,
                       n_iter = N_ITER, max_dev = 0.2, alt_name = "Lynetteholmen", econ_index =
1)
# 50%
mc_totals_50 <- lapply(score_sets, monte_carlo_totals_scores_weights,
                       base_weights = weights, beneficial.vector = beneficial.vector,
                       n_iter = N_ITER, max_dev = 0.5, alt_name = "Lynetteholmen", econ_index =
1)

# probs
prob20 <- round(sapply(mc_totals_20, function(x) mean(x > 0) * 100), 1)
prob50 <- round(sapply(mc_totals_50, function(x) mean(x > 0) * 100), 1)

# ------------------------------
# Build combined table
# ------------------------------
cols <- names(score_sets)

# Row 1: Scores (unperturbed Lynetteholmen totals)
scores_row <- c(manual_simple_scores, manual_ln_simple_scores,
                manual_proportional_scores, manual_ln_proportional_scores)
names(scores_row) <- cols

# Row 2: Top contributor (unperturbed)
top_row <- c(simple_contrib_info["biggest"], ln_simple_contrib_info["biggest"],
             prop_contrib_info["biggest"], ln_prop_contrib_info["biggest"])
names(top_row) <- cols

# Row 3: Bottom contributor (unperturbed)
bottom_row <- c(simple_contrib_info["smallest"], ln_simple_contrib_info["smallest"],
                prop_contrib_info["smallest"], ln_prop_contrib_info["smallest"])
names(bottom_row) <- cols

# Row 4: Monte Carlo 20% probabilities
mc20_row <- paste0(prob20, " %")
names(mc20_row) <- cols
```

```
# Row 5: Monte Carlo 50%
mc50_row <- paste0(prob50, " %")
names(mc50_row) <- cols

# Row 6: Empty spacer
empty_row <- rep(" ", length(cols))
names(empty_row) <- cols

# ------------------------------------------------------
# Row 7 and Row 8
# ------------------------------------------------------

# Positive criteria (beneficial)
pos_idx <- c(1,2,3,4)
pos_names <- crit_names[pos_idx]
pos_vals <- sprintf("%.3f", weights[pos_idx])
pos_w_row <- paste0(pos_names, " = ", pos_vals)
names(pos_w_row) <- cols

# Negative criteria (non-beneficial)
neg_idx <- c(5,6,7,8)
neg_names <- crit_names[neg_idx]
neg_vals <- sprintf("%.3f", weights[neg_idx])
neg_w_row <- paste0(neg_names, " = ", neg_vals)
names(neg_w_row) <- cols

# ------------------------------
# Combine into a table
# ------------------------------
combined <- rbind(
  Score = scores_row,
  Top_Contributor = top_row,
  Bottom_Contributor = bottom_row,
  Empty2 = empty_row,
  MonteCarlo_20pct = mc20_row,
  MonteCarlo_50pct = mc50_row,
  Empty = empty_row,
  Positive_Weights = pos_w_row,
  Negative_Weights = neg_w_row
)


# Convert to data.frame and transpose so that columns = analyses, rows = requested items
combined_df <- as.data.frame(combined, stringsAsFactors = FALSE)

# Build display table: transform Score row into formatted (colored) HTML cells
score_vals <- as.numeric(as.character(combined_df["Score", ]))
score_cells <- sapply(score_vals, function(v) {
  if (is.na(v)) return(cell_spec("", "html"))
  if (v > 0) {
    cell_spec(round(v,3), "html", background = "#d9f2d9")   # green
  } else if (v < 0) {
    cell_spec(round(v,3), "html", background = "#f2d9d9")   # red
  } else {
    cell_spec(round(v,3), "html")
  }
})

# For remaining rows, keep values as-is (already characters)
display_df <- combined_df
display_df["Score", ] <- score_cells

# Replace underscores in rownames for nicer display
rownames(display_df) <- c("Score", "Top Contributor", "Bottom Contributor", " ",
                          "Monte Carlo 20 %", "Monte Carlo 50 %", "",
                          "Positive Weights",
                          "Negative Weights")
```

```
# ------------------------------
# Styled output table
# ------------------------------

prob_table_kbl <- kbl(display_df, caption = title, align = "c", row.names = TRUE, escape =
FALSE) %>%
  kable_styling(full_width = FALSE, bootstrap_options = c("striped","hover","condensed")) %>%
  row_spec(0, background = "#DFF2F2") %>%                         # header
  row_spec(nrow(display_df) - 1, background = "#DFF2F2") %>%     # Positive Weights
  row_spec(nrow(display_df), background = "#DFF2F2") %>%         # Negative Weights
  column_spec(1, border_right = TRUE)

# Save HTML → PNG
tmp_html <- tempfile(fileext = ".html")
save_kable(prob_table_kbl, file = tmp_html)

if (requireNamespace("rstudioapi", quietly = TRUE) && rstudioapi::isAvailable()) {
  script_dir <- dirname(rstudioapi::getActiveDocumentContext()$path)
} else {
  script_dir <- getwd()
}

combined_table_file <- file.path(script_dir, paste0(title, " - Combined Table.png"))
webshot(tmp_html, file = combined_table_file, vwidth = 1200, vheight = 900, zoom = 2)

cat("Saved combined table to:", combined_table_file, "\n")


# ------------------------------
# Tornado Plot
# ------------------------------
p1 <- plot_tornado_percent(simple_contribs, "Assigned")
p2 <- plot_tornado_percent(ln_simple_contribs, "ln(Assigned)")
p3 <- plot_tornado_percent(proportional_contribs, "Proportional")
p4 <- plot_tornado_percent(ln_proportional_contribs, "ln(Proportional)")

combined_percent_plot <- (p1 | p2) / (p3 | p4) + plot_annotation(title = title)

if (requireNamespace("rstudioapi", quietly = TRUE) && rstudioapi::isAvailable()) {
  script_dir <- dirname(rstudioapi::getActiveDocumentContext()$path)
} else {
  script_dir <- getwd()
}

# Save unperturbed outputs
unperturbed_tornado_file <- file.path(script_dir, paste0(title, " - Constant Tornado.png"))

# save tornado
ggsave(unperturbed_tornado_file, combined_percent_plot, width = 12, height = 8, dpi = SAVE_DPI)
cat("Saved unperturbed tornado to:", unperturbed_tornado_file, "\n")


# ------------------------------
# MC Histograms
# ------------------------------
hist_plots_20 <- mapply(function(x,nm) {
  ggplot(data.frame(score = x), aes(x = score)) +
    geom_histogram(bins = 50, fill = "#3B9C9C", color = "black") +
    labs(title = paste0(nm, " – 20% MC Histogram"), x = "Total Score", y = "Frequency") +
    theme_minimal()
}, mc_totals_20, names(mc_totals_20), SIMPLIFY = FALSE)

hist_plots_50 <- mapply(function(x,nm) {
  ggplot(data.frame(score = x), aes(x = score)) +
    geom_histogram(bins = 50, fill = "#3B9C9C", color = "black") +
    labs(title = paste0(nm, " – 50% MC Histogram"), x = "Total Score", y = "Frequency") +
    theme_minimal()
}, mc_totals_50, names(mc_totals_50), SIMPLIFY = FALSE)
```

```r
#Combine Histograms
combined_hist_20 <- (hist_plots_20[[1]] | hist_plots_20[[2]]) / (hist_plots_20[[3]] |
hist_plots_20[[4]]) +
  plot_annotation(title = paste0(title, " – 20% Monte Carlo Histograms"))

combined_hist_50 <- (hist_plots_50[[1]] | hist_plots_50[[2]]) / (hist_plots_50[[3]] |
hist_plots_50[[4]]) +
  plot_annotation(title = paste0(title, " – 50% Monte Carlo Histograms"))

combined_hist_20_file <- file.path(script_dir, paste0(title, " - MC Histograms 20.png"))
combined_hist_50_file <- file.path(script_dir, paste0(title, " - MC Histograms 50.png"))

ggsave(combined_hist_20_file, combined_hist_20, width = 12, height = 8, dpi = SAVE_DPI)
ggsave(combined_hist_50_file, combined_hist_50, width = 12, height = 8, dpi = SAVE_DPI)

cat("Saved combined MC plots:\n", combined_hist_20_file, "\n", combined_hist_50_file, "\n")

cat("All done — outputs saved in:", script_dir, "\n")


# ------------------------------
# 6 Criteria Analysis - Inflated Economic Effects
# ------------------------------

crit_names <- c("Economic Effects", "Traffic", "Conservation", "Environmental Effects", "Health
and Wellbeing", "Resource Use")
alt_names <- c("Lynetteholmen", "0-Scenario")
beneficial.vector <- c(1,2)  # indices of beneficial criteria
N_ITER <- 50000   # Monte Carlo iterations
SAVE_DPI <- 300

#weights <- make_weights("equal", crit_names = crit_names)
weights <- make_weights("rationale", crit = c("Economic Effects", "Traffic"), overvalue = 5,
crit_names = crit_names)
#weights <- make_weights("extreme", crit = "Economic Effects", factor = 5, crit_names =
crit_names)
title <- "6 Inflated - Economic Rationale 500 PCT."

# ------------------------------
# Input data
# ------------------------------
# Score matrices: two alternatives (Lynetteholmen, 0-Scenario)
simple_scores <- matrix(c(4,5,6,11,4,5, 0,0,0,0,0,0), nrow = 2, byrow = TRUE)
ln_simple_scores <- matrix(c(1.61,1.79,1.95,2.48,1.61,1.79,  0,0,0,0,0,0), nrow = 2, byrow =
TRUE)
proportional_scores <- matrix(c(0.57,1,1.2,1.38,0.67,1, 0,0,0,0,0,0), nrow = 2, byrow = TRUE)
ln_proportional_scores <- matrix(c(0.45,0.69,0.79,0.87,0.51,0.69, 0,0,0,0,0,0), nrow = 2, byrow
= TRUE)

# set names
rownames(simple_scores) <- alt_names; colnames(simple_scores) <- crit_names
rownames(ln_simple_scores) <- alt_names; colnames(ln_simple_scores) <- crit_names
rownames(proportional_scores) <- alt_names; colnames(proportional_scores) <- crit_names
rownames(ln_proportional_scores) <- alt_names; colnames(ln_proportional_scores) <- crit_names

# bundle score sets
score_sets <- list(
  "Assigned" = simple_scores,
  "ln(Assigned)" = ln_simple_scores,
  "Proportional" = proportional_scores,
  "ln(Proportional)" = ln_proportional_scores
)

# ------------------------------
# Unperturbed computations
# ------------------------------
simple_contribs <- compute_contribs(simple_scores, weights, beneficial.vector)
ln_simple_contribs <- compute_contribs(ln_simple_scores, weights, beneficial.vector)
proportional_contribs <- compute_contribs(proportional_scores, weights, beneficial.vector)
```

```
ln_proportional_contribs <- compute_contribs(ln_proportional_scores, weights,
beneficial.vector)

manual_simple_scores <- rowSums(simple_contribs)["Lynetteholmen"]
manual_ln_simple_scores <- rowSums(ln_simple_contribs)["Lynetteholmen"]
manual_proportional_scores <- rowSums(proportional_contribs)["Lynetteholmen"]
manual_ln_proportional_scores <- rowSums(ln_proportional_contribs)["Lynetteholmen"]

# top / bottom contributors (unperturbed)
simple_contrib_info <- get_lynette_contrib(simple_contribs)
ln_simple_contrib_info <- get_lynette_contrib(ln_simple_contribs)
prop_contrib_info <- get_lynette_contrib(proportional_contribs)
ln_prop_contrib_info <- get_lynette_contrib(ln_proportional_contribs)

# ------------------------------
# Monte Carlo
# ------------------------------
# 20%
mc_totals_20 <- lapply(score_sets, monte_carlo_totals_scores_weights,
                       base_weights = weights, beneficial.vector = beneficial.vector,
                       n_iter = N_ITER, max_dev = 0.2, alt_name = "Lynetteholmen", econ_index =
1)
# 50%
mc_totals_50 <- lapply(score_sets, monte_carlo_totals_scores_weights,
                       base_weights = weights, beneficial.vector = beneficial.vector,
                       n_iter = N_ITER, max_dev = 0.5, alt_name = "Lynetteholmen", econ_index =
1)

# probs
prob20 <- round(sapply(mc_totals_20, function(x) mean(x > 0) * 100), 1)
prob50 <- round(sapply(mc_totals_50, function(x) mean(x > 0) * 100), 1)

# ------------------------------
# Build combined table
# ------------------------------
cols <- names(score_sets)

# Row 1: Scores (unperturbed Lynetteholmen totals)
scores_row <- c(manual_simple_scores, manual_ln_simple_scores,
                manual_proportional_scores, manual_ln_proportional_scores)
names(scores_row) <- cols

# Row 2: Top contributor (unperturbed)
top_row <- c(simple_contrib_info["biggest"], ln_simple_contrib_info["biggest"],
             prop_contrib_info["biggest"], ln_prop_contrib_info["biggest"])
names(top_row) <- cols

# Row 3: Bottom contributor (unperturbed)
bottom_row <- c(simple_contrib_info["smallest"], ln_simple_contrib_info["smallest"],
                prop_contrib_info["smallest"], ln_prop_contrib_info["smallest"])
names(bottom_row) <- cols

# Row 4: Monte Carlo 20% probabilities
mc20_row <- paste0(prob20, " %")
names(mc20_row) <- cols

# Row 5: Monte Carlo 50%
mc50_row <- paste0(prob50, " %")
names(mc50_row) <- cols

# Row 6: Empty spacer
empty_row <- rep(" ", length(cols))
names(empty_row) <- cols

# ------------------------------------------------------------
# Row 7 and Row 8
# ------------------------------------------------------------

# Positive criteria (beneficial)
```

```
pos_idx <- c(1,2)
pos_names <- crit_names[pos_idx]
pos_vals <- sprintf("%.3f", weights[pos_idx])
pos_w_row <- paste0(pos_names, " = ", pos_vals)

pos_w_row_full <- rep(" ", length(cols))       # start with empty cells
pos_w_row_full[1:length(pos_w_row)] <- pos_w_row   # fill first 2 columns

# Negative criteria (non-beneficial)
neg_idx <- c(3,4,5,6)
neg_names <- crit_names[neg_idx]
neg_vals <- sprintf("%.3f", weights[neg_idx])
neg_w_row <- paste0(neg_names, " = ", neg_vals)

neg_w_row_full <- rep(" ", length(cols))      # start with empty cells
neg_w_row_full[1:length(neg_w_row)] <- neg_w_row   # place in first 4 columns

# ------------------------------
# Combine into a table
# ------------------------------
combined <- rbind(
  Score = scores_row,
  Top_Contributor = top_row,
  Bottom_Contributor = bottom_row,
  Spacer1 = empty_row,
  MonteCarlo_20pct = mc20_row,
  MonteCarlo_50pct = mc50_row,
  Spacer2 = empty_row,
  Positive_Weights = pos_w_row_full,
  Negative_Weights = neg_w_row_full
)


# Convert to data.frame and transpose so that columns = analyses, rows = requested items
combined_df <- as.data.frame(combined, stringsAsFactors = FALSE)

# Build display table: transform Score row into formatted (colored) HTML cells
score_vals <- as.numeric(as.character(combined_df["Score", ]))
score_cells <- sapply(score_vals, function(v) {
  if (is.na(v)) return(cell_spec("", "html"))
  if (v > 0) {
    cell_spec(round(v,3), "html", background = "#d9f2d9")   # green
  } else if (v < 0) {
    cell_spec(round(v,3), "html", background = "#f2d9d9")   # red
  } else {
    cell_spec(round(v,3), "html")
  }
})

display_df <- combined_df
display_df["Score", ] <- score_cells


rownames(display_df) <- c("Score", "Top Contributor", "Bottom Contributor", " ",
                          "Monte Carlo 20 %", "Monte Carlo 50 %", "",
                          "Positive Weights",
                          "Negative Weights")

# ------------------------------
# Styled output table
# ------------------------------

prob_table_kbl <- kbl(display_df, caption = title, align = "c", row.names = TRUE, escape =
FALSE) %>%
  kable_styling(full_width = FALSE, bootstrap_options = c("striped","hover","condensed")) %>%
  row_spec(0, background = "#DFF2F2") %>%                        # header
  row_spec(nrow(display_df) - 1, background = "#DFF2F2") %>%     # Positive Weights
  row_spec(nrow(display_df), background = "#DFF2F2") %>%         # Negative Weights
  column_spec(1, border_right = TRUE)
```

```
# Save HTML → PNG
tmp_html <- tempfile(fileext = ".html")
save_kable(prob_table_kbl, file = tmp_html)

if (requireNamespace("rstudioapi", quietly = TRUE) && rstudioapi::isAvailable()) {
  script_dir <- dirname(rstudioapi::getActiveDocumentContext()$path)
} else {
  script_dir <- getwd()
}

combined_table_file <- file.path(script_dir, paste0(title, " - Combined Table.png"))
webshot(tmp_html, file = combined_table_file, vwidth = 1200, vheight = 900, zoom = 2)

cat("Saved combined table to:", combined_table_file, "\n")

# ------------------------------
# Tornado Plot
# ------------------------------
p1 <- plot_tornado_percent(simple_contribs, "Assigned")
p2 <- plot_tornado_percent(ln_simple_contribs, "ln(Assigned)")
p3 <- plot_tornado_percent(proportional_contribs, "Proportional")
p4 <- plot_tornado_percent(ln_proportional_contribs, "ln(Proportional)")

combined_percent_plot <- (p1 | p2) / (p3 | p4) + plot_annotation(title = title)

if (requireNamespace("rstudioapi", quietly = TRUE) && rstudioapi::isAvailable()) {
  script_dir <- dirname(rstudioapi::getActiveDocumentContext()$path)
} else {
  script_dir <- getwd()
}

# Save unperturbed outputs
unperturbed_tornado_file <- file.path(script_dir, paste0(title, " - Constant Tornado.png"))

# save tornado
ggsave(unperturbed_tornado_file, combined_percent_plot, width = 12, height = 8, dpi = SAVE_DPI)
cat("Saved unperturbed tornado to:", unperturbed_tornado_file, "\n")


# ------------------------------
# MC Histograms
# ------------------------------
hist_plots_20 <- mapply(function(x,nm) {
  ggplot(data.frame(score = x), aes(x = score)) +
    geom_histogram(bins = 50, fill = "#3B9C9C", color = "black") +
    labs(title = paste0(nm, " – 20% MC Histogram"), x = "Total Score", y = "Frequency") +
    theme_minimal()
}, mc_totals_20, names(mc_totals_20), SIMPLIFY = FALSE)

hist_plots_50 <- mapply(function(x,nm) {
  ggplot(data.frame(score = x), aes(x = score)) +
    geom_histogram(bins = 50, fill = "#3B9C9C", color = "black") +
    labs(title = paste0(nm, " – 50% MC Histogram"), x = "Total Score", y = "Frequency") +
    theme_minimal()
}, mc_totals_50, names(mc_totals_50), SIMPLIFY = FALSE)

#Combine Histograms
combined_hist_20 <- (hist_plots_20[[1]] | hist_plots_20[[2]]) / (hist_plots_20[[3]] |
hist_plots_20[[4]]) +
  plot_annotation(title = paste0(title, " – 20% Monte Carlo Histograms"))

combined_hist_50 <- (hist_plots_50[[1]] | hist_plots_50[[2]]) / (hist_plots_50[[3]] |
hist_plots_50[[4]]) +
  plot_annotation(title = paste0(title, " – 50% Monte Carlo Histograms"))

combined_hist_20_file <- file.path(script_dir, paste0(title, " - MC Histograms 20.png"))
combined_hist_50_file <- file.path(script_dir, paste0(title, " - MC Histograms 50.png"))
```

```
ggsave(combined_hist_20_file, combined_hist_20, width = 12, height = 8, dpi = SAVE_DPI)
ggsave(combined_hist_50_file, combined_hist_50, width = 12, height = 8, dpi = SAVE_DPI)

cat("Saved combined MC plots:\n", combined_hist_20_file, "\n", combined_hist_50_file, "\n")

cat("All done — outputs saved in:", script_dir, "\n")



# ------------------------------
# 8 Criteria Analysis - Inflated Economic Effects
# ------------------------------

crit_names <- c("Economic Effects", "Traffic", "Soil Deposit", "Storm Surge",
                "Conservation", "Environmental Effects", "Health and Wellbeing", "Resource
Use")
alt_names <- c("Lynetteholmen", "0-Scenario")
beneficial.vector <- c(1,2,3,4)  # indices of beneficial criteria
N_ITER <- 50000   # Monte Carlo iterations
SAVE_DPI <- 300

#weights <- make_weights("equal", crit_names = crit_names)
weights <- make_weights("rationale", crit = c("Economic Effects", "Traffic"), overvalue = 5,
crit_names = crit_names)
#weights <- make_weights("extreme", crit = "Storm Surge", factor = 5, crit_names = crit_names)
title <- "8 Inflated - Economic Rationale 500 PCT."

# ------------------------------
# Input data
# ------------------------------
# Score matrices: two alternatives (Lynetteholmen, 0-Scenario)
simple_scores <- matrix(c(4,5,5,5,9,11,4,8,  0,0,0,0,0,0,0,0), nrow = 2, byrow = TRUE)
ln_simple_scores <- matrix(c(1.61,1.79,1.79,1.79,2.3,2.48,1.61,2.2,  0,0,0,0,0,0,0,0), nrow =
2, byrow = TRUE)
proportional_scores <- matrix(c(0.57,1,1,1,2.25,1.38,0.67,2, 0,0,0,0,0,0,0,0), nrow = 2, byrow
= TRUE)
ln_proportional_scores <- matrix(c(0.45,0.69,0.69,0.69,1.18,0.87,0.51,1.1, 0,0,0,0,0,0,0,0),
nrow = 2, byrow = TRUE)


# set names
rownames(simple_scores) <- alt_names; colnames(simple_scores) <- crit_names
rownames(ln_simple_scores) <- alt_names; colnames(ln_simple_scores) <- crit_names
rownames(proportional_scores) <- alt_names; colnames(proportional_scores) <- crit_names
rownames(ln_proportional_scores) <- alt_names; colnames(ln_proportional_scores) <- crit_names

# bundle score sets
score_sets <- list(
  "Assigned" = simple_scores,
  "ln(Assigned)" = ln_simple_scores,
  "Proportional" = proportional_scores,
  "ln(Proportional)" = ln_proportional_scores
)

# ------------------------------
# Unperturbed computations
# ------------------------------
simple_contribs <- compute_contribs(simple_scores, weights, beneficial.vector)
ln_simple_contribs <- compute_contribs(ln_simple_scores, weights, beneficial.vector)
proportional_contribs <- compute_contribs(proportional_scores, weights, beneficial.vector)
ln_proportional_contribs <- compute_contribs(ln_proportional_scores, weights,
beneficial.vector)

manual_simple_scores <- rowSums(simple_contribs)["Lynetteholmen"]
manual_ln_simple_scores <- rowSums(ln_simple_contribs)["Lynetteholmen"]
manual_proportional_scores <- rowSums(proportional_contribs)["Lynetteholmen"]
manual_ln_proportional_scores <- rowSums(ln_proportional_contribs)["Lynetteholmen"]

# top / bottom contributors (unperturbed)
```

```
simple_contrib_info <- get_lynette_contrib(simple_contribs)
ln_simple_contrib_info <- get_lynette_contrib(ln_simple_contribs)
prop_contrib_info <- get_lynette_contrib(proportional_contribs)
ln_prop_contrib_info <- get_lynette_contrib(ln_proportional_contribs)

# ------------------------------
# Monte Carlo
# ------------------------------
# 20%
mc_totals_20 <- lapply(score_sets, monte_carlo_totals_scores_weights,
                       base_weights = weights, beneficial.vector = beneficial.vector,
                       n_iter = N_ITER, max_dev = 0.2, alt_name = "Lynetteholmen", econ_index =
1)
# 50%
mc_totals_50 <- lapply(score_sets, monte_carlo_totals_scores_weights,
                       base_weights = weights, beneficial.vector = beneficial.vector,
                       n_iter = N_ITER, max_dev = 0.5, alt_name = "Lynetteholmen", econ_index =
1)

# probs
prob20 <- round(sapply(mc_totals_20, function(x) mean(x > 0) * 100), 1)
prob50 <- round(sapply(mc_totals_50, function(x) mean(x > 0) * 100), 1)

# ------------------------------
# Build combined table (rows x columns)
# ------------------------------
cols <- names(score_sets)

# Row 1: Scores (unperturbed Lynetteholmen totals)
scores_row <- c(manual_simple_scores, manual_ln_simple_scores,
                manual_proportional_scores, manual_ln_proportional_scores)
names(scores_row) <- cols

# Row 2: Top contributor (unperturbed)
top_row <- c(simple_contrib_info["biggest"], ln_simple_contrib_info["biggest"],
             prop_contrib_info["biggest"], ln_prop_contrib_info["biggest"])
names(top_row) <- cols

# Row 3: Bottom contributor (unperturbed)
bottom_row <- c(simple_contrib_info["smallest"], ln_simple_contrib_info["smallest"],
                prop_contrib_info["smallest"], ln_prop_contrib_info["smallest"])
names(bottom_row) <- cols

# Row 4: Monte Carlo 20% probabilities
mc20_row <- paste0(prob20, " %")
names(mc20_row) <- cols

# Row 5: Monte Carlo 50%
mc50_row <- paste0(prob50, " %")
names(mc50_row) <- cols

# Row 6: Empty spacer
empty_row <- rep(" ", length(cols))
names(empty_row) <- cols

# ------------------------------------------------------------
# Row 7 and Row 8
# ------------------------------------------------------------

# Positive criteria (beneficial)
pos_idx <- c(1,2,3,4)
pos_names <- crit_names[pos_idx]
pos_vals <- sprintf("%.3f", weights[pos_idx])
pos_w_row <- paste0(pos_names, " = ", pos_vals)
names(pos_w_row) <- cols

# Negative criteria (non-beneficial)
neg_idx <- c(5,6,7,8)
neg_names <- crit_names[neg_idx]
```

```
neg_vals <- sprintf("%.3f", weights[neg_idx])
neg_w_row <- paste0(neg_names, " = ", neg_vals)
names(neg_w_row) <- cols

# ------------------------------
# Combine into a table
# ------------------------------
combined <- rbind(
  Score = scores_row,
  Top_Contributor = top_row,
  Bottom_Contributor = bottom_row,
  Empty2 = empty_row,
  MonteCarlo_20pct = mc20_row,
  MonteCarlo_50pct = mc50_row,
  Empty = empty_row,
  Positive_Weights = pos_w_row,
  Negative_Weights = neg_w_row
)


# Convert to data.frame and transpose so that columns = analyses, rows = requested items
combined_df <- as.data.frame(combined, stringsAsFactors = FALSE)

# Build display table: transform Score row into formatted (colored) HTML cells
score_vals <- as.numeric(as.character(combined_df["Score", ]))
score_cells <- sapply(score_vals, function(v) {
  if (is.na(v)) return(cell_spec("", "html"))
  if (v > 0) {
    cell_spec(round(v,3), "html", background = "#d9f2d9")   # green
  } else if (v < 0) {
    cell_spec(round(v,3), "html", background = "#f2d9d9")   # red
  } else {
    cell_spec(round(v,3), "html")
  }
})

# For remaining rows, keep values as-is (already characters)
display_df <- combined_df
display_df["Score", ] <- score_cells

# Replace underscores in rownames for nicer display
rownames(display_df) <- c("Score", "Top Contributor", "Bottom Contributor", " ",
                          "Monte Carlo 20 %", "Monte Carlo 50 %", "",
                          "Positive Weights",
                          "Negative Weights")

# ------------------------------
# Styled output table
# ------------------------------

prob_table_kbl <- kbl(display_df, caption = title, align = "c", row.names = TRUE, escape =
FALSE) %>%
  kable_styling(full_width = FALSE, bootstrap_options = c("striped","hover","condensed")) %>%
  row_spec(0, background = "#DFF2F2") %>%                       # header
  row_spec(nrow(display_df) - 1, background = "#DFF2F2") %>%     # Positive Weights
  row_spec(nrow(display_df), background = "#DFF2F2") %>%         # Negative Weights
  column_spec(1, border_right = TRUE)

# Save HTML → PNG
tmp_html <- tempfile(fileext = ".html")
save_kable(prob_table_kbl, file = tmp_html)

if (requireNamespace("rstudioapi", quietly = TRUE) && rstudioapi::isAvailable()) {
  script_dir <- dirname(rstudioapi::getActiveDocumentContext()$path)
} else {
  script_dir <- getwd()
}

combined_table_file <- file.path(script_dir, paste0(title, " - Combined Table.png"))
```

```
webshot(tmp_html, file = combined_table_file, vwidth = 1200, vheight = 900, zoom = 2)

cat("Saved combined table to:", combined_table_file, "\n")


# ------------------------------
# Tornado Plot
# ------------------------------
p1 <- plot_tornado_percent(simple_contribs, "Assigned")
p2 <- plot_tornado_percent(ln_simple_contribs, "ln(Assigned)")
p3 <- plot_tornado_percent(proportional_contribs, "Proportional")
p4 <- plot_tornado_percent(ln_proportional_contribs, "ln(Proportional)")

combined_percent_plot <- (p1 | p2) / (p3 | p4) + plot_annotation(title = title)

if (requireNamespace("rstudioapi", quietly = TRUE) && rstudioapi::isAvailable()) {
  script_dir <- dirname(rstudioapi::getActiveDocumentContext()$path)
} else {
  script_dir <- getwd()
}

# Save unperturbed outputs
unperturbed_tornado_file <- file.path(script_dir, paste0(title, " - Constant Tornado.png"))

# save tornado
ggsave(unperturbed_tornado_file, combined_percent_plot, width = 12, height = 8, dpi = SAVE_DPI)
cat("Saved unperturbed tornado to:", unperturbed_tornado_file, "\n")


# ------------------------------
# MC Histograms
# ------------------------------
hist_plots_20 <- mapply(function(x,nm) {
  ggplot(data.frame(score = x), aes(x = score)) +
    geom_histogram(bins = 50, fill = "#3B9C9C", color = "black") +
    labs(title = paste0(nm, " – 20% MC Histogram"), x = "Total Score", y = "Frequency") +
    theme_minimal()
}, mc_totals_20, names(mc_totals_20), SIMPLIFY = FALSE)

hist_plots_50 <- mapply(function(x,nm) {
  ggplot(data.frame(score = x), aes(x = score)) +
    geom_histogram(bins = 50, fill = "#3B9C9C", color = "black") +
    labs(title = paste0(nm, " – 50% MC Histogram"), x = "Total Score", y = "Frequency") +
    theme_minimal()
}, mc_totals_50, names(mc_totals_50), SIMPLIFY = FALSE)

#Combine Histograms
combined_hist_20 <- (hist_plots_20[[1]] | hist_plots_20[[2]]) / (hist_plots_20[[3]] |
hist_plots_20[[4]]) +
  plot_annotation(title = paste0(title, " – 20% Monte Carlo Histograms"))

combined_hist_50 <- (hist_plots_50[[1]] | hist_plots_50[[2]]) / (hist_plots_50[[3]] |
hist_plots_50[[4]]) +
  plot_annotation(title = paste0(title, " – 50% Monte Carlo Histograms"))

combined_hist_20_file <- file.path(script_dir, paste0(title, " - MC Histograms 20.png"))
combined_hist_50_file <- file.path(script_dir, paste0(title, " - MC Histograms 50.png"))

ggsave(combined_hist_20_file, combined_hist_20, width = 12, height = 8, dpi = SAVE_DPI)
ggsave(combined_hist_50_file, combined_hist_50, width = 12, height = 8, dpi = SAVE_DPI)

cat("Saved combined MC plots:\n", combined_hist_20_file, "\n", combined_hist_50_file, "\n")

cat("All done — outputs saved in:", script_dir, "\n")
```