

Polychaete Assemblages pipeline

16 Jun 2026

Polychaete assemblage pipeline: Assemblages based on occurrences with geographic coordinates

Libraries

```
library(bioregion)
library(tidyverse)
library(vegan)
library(ape)
library(sf)
library(cluster)
library(dendextend)
library(RColorBrewer)
library(DescTools)
```

Open File

```
DF_Assemblages<- read.csv("DF_Assemblages.txt", sep = "\t", header = TRUE)

df <- DF_Assemblages[, c("accepted_taxon", "Lat_inic", "Long_inic")]

df$Lat_inic <- as.numeric(gsub(",", ".", df$Lat_inic))
df$Long_inic <- as.numeric(gsub(",", ".", df$Long_inic))
```

Grid Cell Aggregation

Analysis with 1/4 degree grid cells as sampling stations, using a mathematical rounding technique to aggregate raw coordinates into discrete spatial bins. Grouping all species occurrences that fall within the same 0.25° x 0.25° area into a single “Site”.

```
# 1. Assign each point to its 0.25° cell using floor rounding
# This is the correct, proven method for cell assignment
# The Site ID is based on the SW corner of the cell (consistent label)
df_clean <- df %>%
  select(-any_of(c("Site", "Grid_Lat", "Grid_Long"))) %>%
  mutate(
    Grid_Lat = floor(Lat_inic * 4) / 4,
    Grid_Long = floor(Long_inic * 4) / 4,
    Site = paste(Grid_Lat, Grid_Long, sep = "_")
  )

cat("Number of occupied grid cells:", length(unique(df_clean$Site)), "\n")

## Number of occupied grid cells: 80
```

```

# 2. Create df_geo for downstream use
df_geo <- df_clean

# 3. Build proper 0.25° polygon for each occupied cell
#   The polygon is constructed from the SW corner (Grid_Lat, Grid_Long)
#   which is exactly what the floor method gives us
site_coords <- df_geo %>%
  distinct(Site, Grid_Lat, Grid_Long)

# Build one polygon per site by offsetting 0.25° from the SW corner
occupied_grid_sf <- site_coords %>%
  rowwise() %>%
  mutate(
    geometry = st_polygon(list(matrix(c(
      Grid_Long,      Grid_Lat,      # SW corner
      Grid_Long + 0.25, Grid_Lat,      # SE corner
      Grid_Long + 0.25, Grid_Lat + 0.25, # NE corner
      Grid_Long,      Grid_Lat + 0.25, # NW corner
      Grid_Long,      Grid_Lat        # close polygon
    ), ncol = 2, byrow = TRUE))) %>%
    list()
  ) %>%
  ungroup() %>%
  st_as_sf(crs = 4326) %>%
  mutate(
    Centroid_Lat = Grid_Lat + 0.125, # true centroid of the cell
    Centroid_Long = Grid_Long + 0.125
  )

cat("Number of grid polygons:", nrow(occupied_grid_sf), "\n")

## Number of grid polygons: 80

# 4. Export proper polygon grid to QGIS
st_write(occupied_grid_sf,
  "Balearic_Grid_Cells_025deg.gpkg",
  delete_dsn = TRUE)

## Deleting source `Balearic_Grid_Cells_025deg.gpkg' using driver `GPKG'
## Writing layer `Balearic_Grid_Cells_025deg' to data source
## `Balearic_Grid_Cells_025deg.gpkg' using driver `GPKG'
## Writing 80 features with 5 fields and geometry type Polygon.

```

Create shared lookup table to be used plots

```

## Cell Numbering (shared Lookup – top-to-bottom, Left-to-right within rows)
# Sort: descending latitude (north first), then ascending longitude (west
first)
cell_number_lookup <- site_coords %>%
  arrange(desc(Grid_Lat), Grid_Long) %>%

```

```
mutate(CellNumber = row_number())

# Quick check
cat("Cell numbering preview:\n")

## Cell numbering preview:

print(head(cell_number_lookup, 10))

##           Site Grid_Lat Grid_Long CellNumber
## 1      41_4.75    41.00     4.75           1
## 2      41_5      41.00     5.00           2
## 3     40.75_5    40.75     5.00           3
## 4     40.5_2.5   40.50     2.50           4
## 5     40.5_2.75  40.50     2.75           5
## 6     40.5_3     40.50     3.00           6
## 7     40.5_3.75  40.50     3.75           7
## 8     40.25_2.5  40.25     2.50           8
## 9     40.25_3.25 40.25     3.25           9
## 10    40.25_3.75 40.25     3.75          10
```

Site-Species Matrix

```
# Keep only unique Site-Species pairs
occurrence_data <- unique(df_clean[, c("Site", "accepted_taxon")])

# Build the site-species matrix
comat <- net_to_mat(occurrence_data)
```

Compute Dissimilarity

As recommended for bioregionalization, use the Simpson dissimilarity index to ensure results are not biased by differences in species richness between cells.

```
dist_matrix_raw <- dissimilarity(comat, metric = "Simpson")
```

Run Clustering with Evaluation Range

```
set.seed(42) # for reproducibility

tree_to_eval <- hclu_hierarclust(dist_matrix_raw,
                                method = "average",
                                randomize = TRUE,
                                n_clust = 2:50)
```

Evaluate Bioregions

```
# Create a clean network from the matrix to guarantee ID matching
clean_net <- mat_to_net(comat)

# Calculate evaluation metrics
```

```
eval_metrics <- bioregionalization_metrics(tree_to_eval,
                                           dissimilarity = dist_matrix_raw,
                                           net           = clean_net,
                                           site_col      = "Node1",
                                           species_col   = "Node2",
                                           eval_metric   = "all")
```

Print the full table

```
print(eval_metrics$evaluation_df)
```

##	K	n_clusters	pc_distance	anosim	avg_endemism	tot_endemism
## K_1_1	K_1_1	1	0.00000000	NA	1.0000000	1.0000000
## K_1_2	K_1_2	1	0.00000000	NA	1.0000000	1.0000000
## K_4	K_4	4	0.08557506	0.3034860	1.0000000	1.0000000
## K_5	K_5	5	0.11318581	0.3077317	0.8996510	0.9982669
## K_6	K_6	6	0.58341760	0.5169806	0.8048032	0.8492201
## K_7	K_7	7	0.60097146	0.5320198	0.6889459	0.8440208
## K_8	K_8	8	0.65204831	0.5829783	0.7006597	0.8422877
## K_9	K_9	9	0.66740794	0.5987645	0.6215091	0.8388215
## K_10_1	K_10_1	10	0.72864863	0.6454626	0.5804480	0.8232236
## K_10_2	K_10_2	10	0.72864863	0.6454626	0.5804480	0.8232236
## K_10_3	K_10_3	10	0.72864863	0.6454626	0.5804480	0.8232236
## K_13	K_13	13	0.85345470	0.7224505	0.4542896	0.6256499
## K_14	K_14	14	0.86704675	0.7575871	0.4392575	0.6239168
## K_15	K_15	15	0.86796101	0.7595731	0.4112558	0.6239168
## K_16	K_16	16	0.87671618	0.7803144	0.4005549	0.6204506
## K_17	K_17	17	0.88003876	0.7859283	0.3819676	0.6204506
## K_18	K_18	18	0.91905942	0.8344053	0.3644741	0.5857886
## K_19	K_19	19	0.93273070	0.8517336	0.3539890	0.5857886
## K_20	K_20	20	0.93373639	0.8531849	0.3453454	0.5857886
## K_22_1	K_22_1	22	0.93446780	0.8529388	0.3226084	0.5840555
## K_22_2	K_22_2	22	0.93446780	0.8529388	0.3226084	0.5840555
## K_23	K_23	23	0.96846425	0.9178943	0.3139487	0.5233969
## K_24	K_24	24	0.97722955	0.9229850	0.3058938	0.5025997
## K_25	K_25	25	0.98554849	0.9334075	0.2966700	0.5008666
## K_26	K_26	26	0.98673979	0.9360519	0.2923828	0.4956672
## K_27	K_27	27	0.98707817	0.9362106	0.2822162	0.4852686
## K_28	K_28	28	0.98753530	0.9374338	0.2809133	0.4852686
## K_29	K_29	29	0.98843355	0.9383526	0.2735492	0.4835355
## K_30	K_30	30	0.99136124	0.9429012	0.2693860	0.4818024
## K_31	K_31	31	0.99150753	0.9429222	0.2617714	0.4818024
## K_32	K_32	32	0.99208379	0.9434950	0.2547393	0.4818024
## K_33	K_33	33	0.99272377	0.9454924	0.2546691	0.4818024
## K_34	K_34	34	0.99310230	0.9460420	0.2485988	0.4800693
## K_35_1	K_35_1	35	0.99505939	0.9501616	0.2468648	0.4783362
## K_35_2	K_35_2	35	0.99505939	0.9501616	0.2468648	0.4783362
## K_37	K_37	37	0.99530319	0.9503220	0.2408602	0.4783362
## K_38	K_38	38	0.99610038	0.9519210	0.2358668	0.4783362
## K_39_1	K_39_1	39	0.99724894	0.9551301	0.2332701	0.4783362
## K_39_2	K_39_2	39	0.99724894	0.9551301	0.2332701	0.4783362

```
## K_39_3 K_39_3      39  0.99724894 0.9551301    0.2332701    0.4783362
## K_42      K_42      42  0.99779750 0.9566550    0.2168400    0.4783362
## K_43      K_43      43  0.99797038 0.9570041    0.2129091    0.4783362
## K_44      K_44      44  0.99805643 0.9571349    0.2093073    0.4783362
## K_45      K_45      45  0.99822522 0.9575167    0.2052874    0.4783362
## K_46      K_46      46  0.99877377 0.9599774    0.2051724    0.4766031
## K_47_1 K_47_1      47  0.99898172 0.9603790    0.2010214    0.4748700
## K_47_2 K_47_2      47  0.99898172 0.9603790    0.2010214    0.4748700
## K_49      K_49      49  0.99934743 0.9622080    0.1979185    0.4748700
## K_50      K_50      50  0.99940838 0.9625973    0.1939787    0.4748700
```

— Individual plots for each metric

Where duplicate n_clusters exist (randomization ties), keep the best score

```
eval_df <- eval_metrics$evaluation_df %>%
```

```
  group_by(n_clusters) %>%
```

```
  summarise(
```

```
    pc_distance = max(pc_distance),
```

```
    anosim      = max(anosim),
```

```
    avg_endemism = max(avg_endemism),
```

```
    tot_endemism = max(tot_endemism),
```

```
    .groups = "drop"
```

```
)
```

Helper function to plot one metric with a vertical line at its elbow/peak

```
plot_metric <- function(df, metric, colour, title, higher_is_better = TRUE) {
```

```
  best_k <- if (higher_is_better) {
```

```
    df$n_clusters[which.max(df[[metric])]
```

```
  } else {
```

```
    df$n_clusters[which.min(df[[metric])]
```

```
  }
```

```
  ggplot(df, aes(x = n_clusters, y = .data[[metric]])) +
```

```
    geom_line(colour = colour, linewidth = 0.9) +
```

```
    geom_point(colour = colour, size = 3) +
```

```
    geom_vline(xintercept = best_k,
```

```
               linetype = "dashed", colour = "grey40", linewidth = 0.7) +
```

```
    annotate("text",
```

```
            x = best_k + 0.5, y = min(df[[metric]]),
```

```
            label = paste("k =", best_k),
```

```
            hjust = 0, size = 3.5, colour = "grey40") +
```

```
    scale_x_continuous(breaks = seq(2, max(df$n_clusters), by = 2)) +
```

```
    labs(title = title,
```

```
         x = "Number of clusters (k)",
```

```
         y = metric) +
```

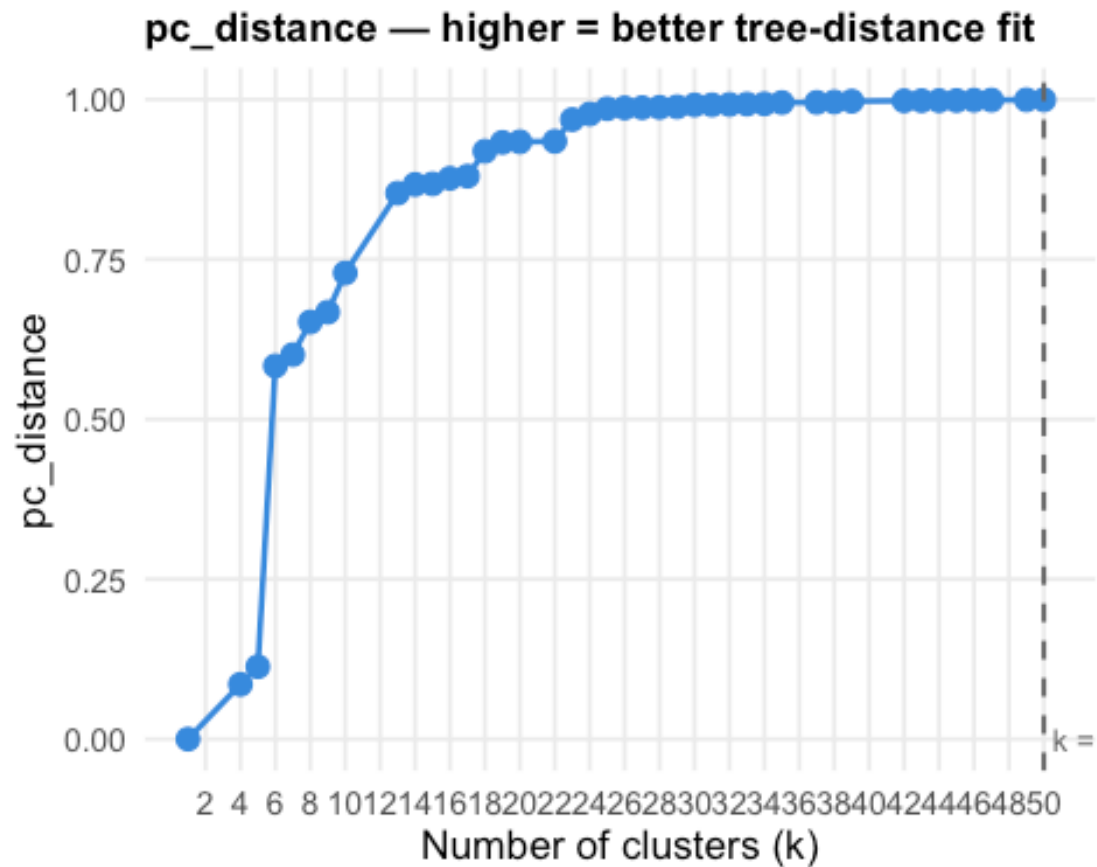
```
    theme_minimal(base_size = 13) +
```

```
    theme(panel.grid.minor = element_blank(),
```

```
          plot.title = element_text(size = 13, face = "bold"))
```

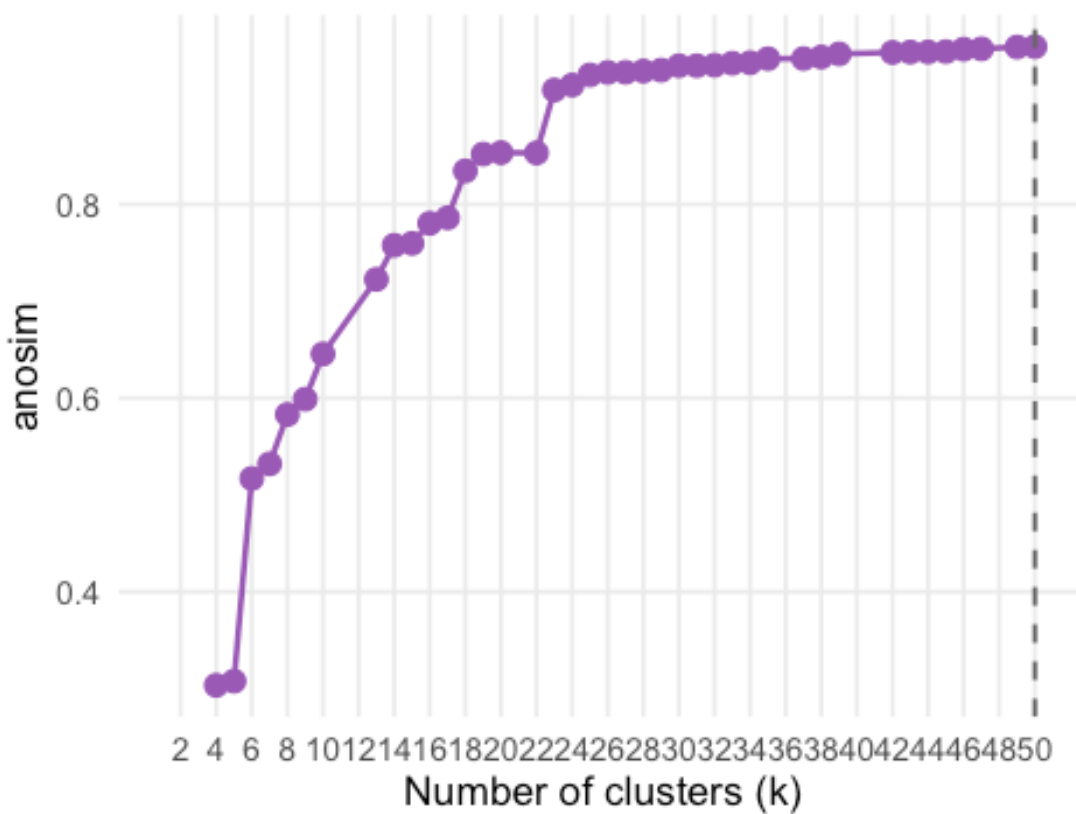
```
}
```

```
# Plot each metric individually
plot_metric(eval_df, "pc_distance",
            colour      = "#378ADD",
            title        = "pc_distance – higher = better tree-distance
fit",
            higher_is_better = TRUE)
```

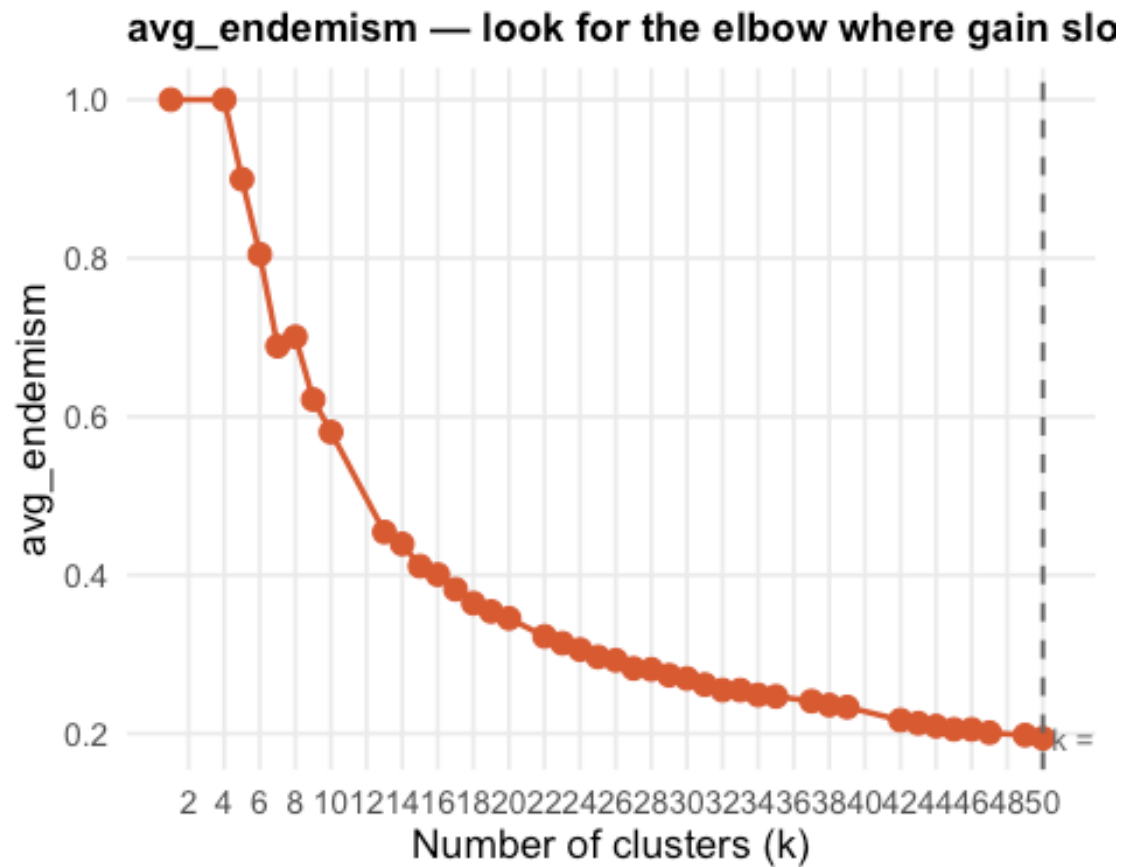


```
plot_metric(eval_df, "anosim",
            colour      = "#9B59B6",
            title        = "ANOSIM – higher = better cluster separation",
            higher_is_better = TRUE)
```

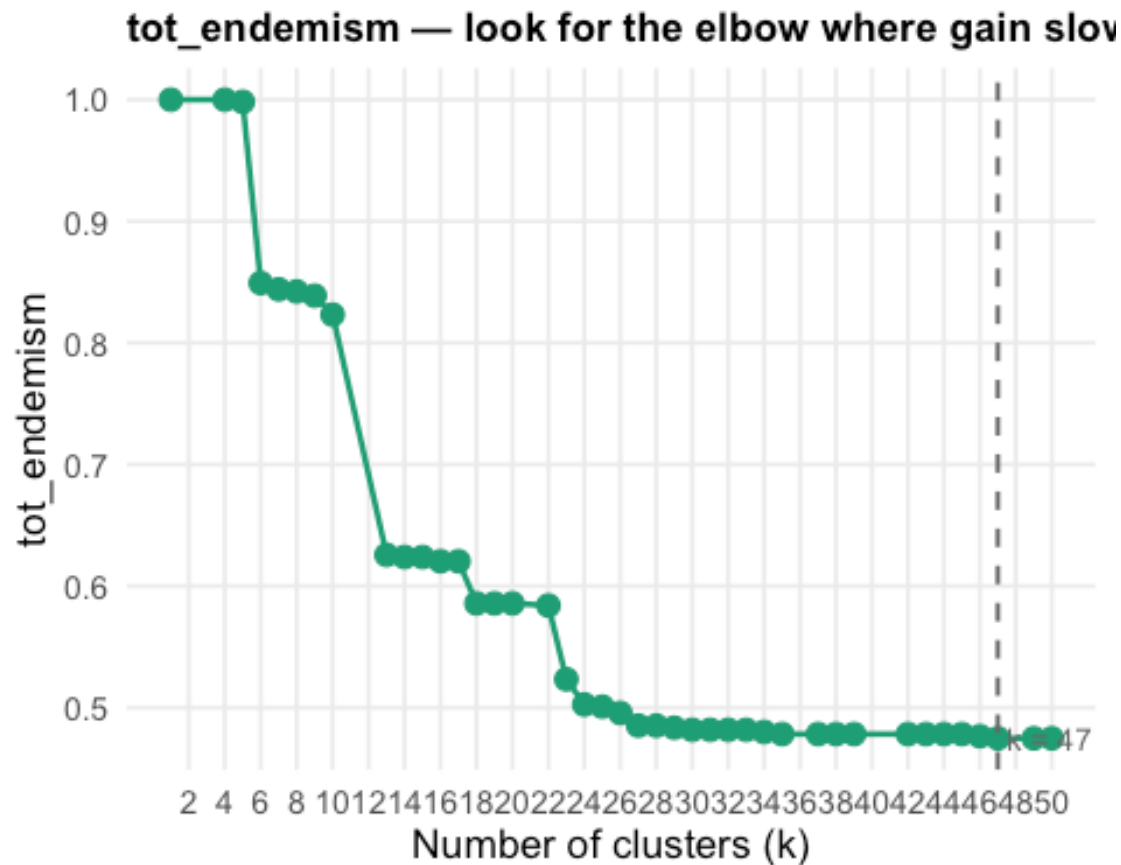
ANOSIM — higher = better cluster separation



```
plot_metric(eval_df, "avg_endemism",  
            colour = "#D85A30",  
            title = "avg_endemism – look for the elbow where gain  
slows",  
            higher_is_better = FALSE)
```



```
plot_metric(eval_df, "tot_endemism",
            colour = "#1D9E75",
            title = "tot_endemism — look for the elbow where gain
slows",
            higher_is_better = FALSE)
```



Multi-Criterion Plot to Choose k

```
# Build dist object needed for silhouette
dist_df      <- as.data.frame(dist_matrix_raw)
dist_df$Simpson <- as.numeric(dist_df$Simpson)
dist_sq      <- pivot_wider(dist_df, names_from = Site2, values_from =
Simpson)
dist_sq      <- column_to_rownames(dist_sq, "Site1")
dist_sq      <- as.matrix(dist_sq)
dist_sq[is.na(dist_sq)] <- t(dist_sq)[is.na(dist_sq)]
diag(dist_sq) <- 0
dist_for_pcoa <- as.dist(dist_sq)

# Extract the hclust tree
tree_hclust  <- tree_to_eval$algorithm$final.tree
dist_labels  <- attr(dist_for_pcoa, "Labels")

# Silhouette scores for k = 2:20
sil_scores <- sapply(2:20, function(k) {
  clusters <- cutree(tree_hclust, k = k)
  clusters <- clusters[dist_labels]
  mean(silhouette(clusters, dist_for_pcoa)[, 3])
})
```

```

sil_df <- data.frame(n_clusters = 2:20,
                    score       = sil_scores,
                    criterion    = "Silhouette")

# Extract pc_distance and avg_endemism – keep best score per k
eval_df <- eval_metrics$evaluation_df

pc_df <- eval_df %>%
  group_by(n_clusters) %>%
  summarise(score = max(pc_distance), .groups = "drop") %>%
  filter(n_clusters >= 2, n_clusters <= 20) %>%
  mutate(criterion = "pc_distance",
         score      = (score - min(score)) / (max(score) - min(score)))

avg_end_df <- eval_df %>%
  group_by(n_clusters) %>%
  summarise(score = max(avg_endemism), .groups = "drop") %>%
  filter(n_clusters >= 2, n_clusters <= 20) %>%
  mutate(criterion = "avg_endemism",
         score      = (score - min(score)) / (max(score) - min(score)),
         score      = 1 - score) # invert: Lower avg_endemism = better
separation

sil_df <- sil_df %>%
  mutate(score = (score - min(score)) / (max(score) - min(score)))

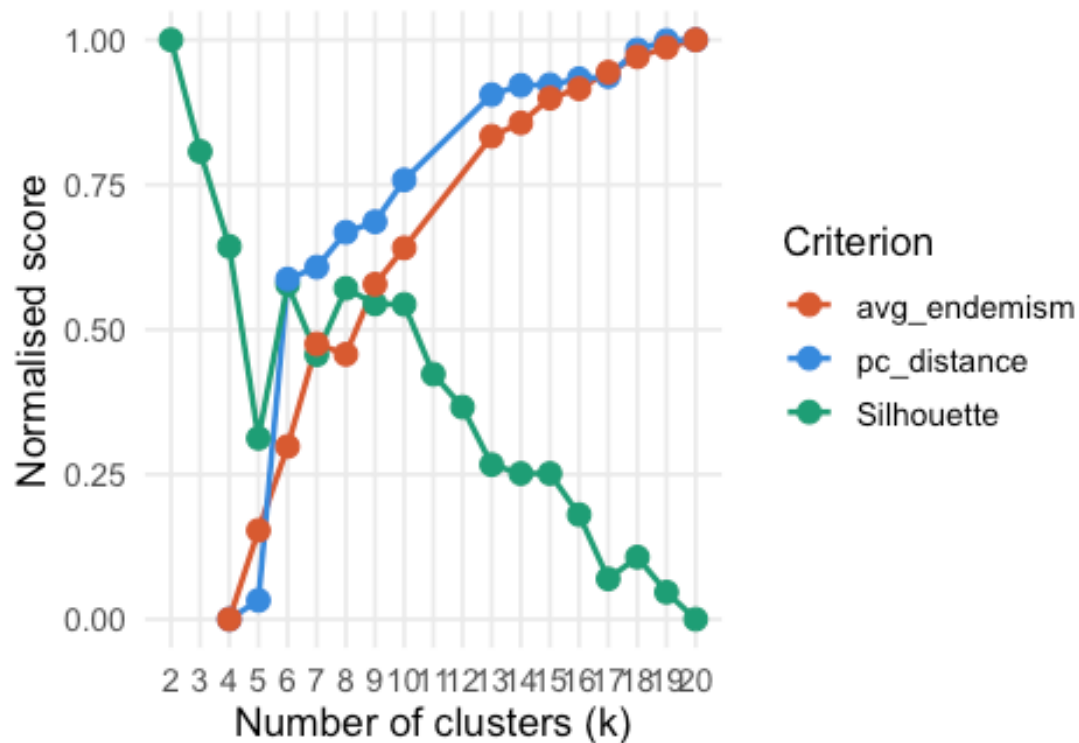
combined_df <- bind_rows(sil_df, pc_df, avg_end_df)

ggplot(combined_df, aes(x = n_clusters, y = score, color = criterion)) +
  geom_line(linewidth = 0.9) +
  geom_point(size = 3) +
  scale_x_continuous(breaks = 2:20) +
  scale_color_manual(
    values = c("Silhouette" = "#1D9E75",
              "pc_distance" = "#378ADD",
              "avg_endemism" = "#D85A30"),
    name = "Criterion"
  ) +
  labs(title = "Multi-criterion comparison for optimal number of
bioregions",
       subtitle = "All scores normalised 0-1 · avg_endemism inverted so
higher = better",
       x = "Number of clusters (k)",
       y = "Normalised score") +
  theme_minimal(base_size = 13) +
  theme(panel.grid.minor = element_blank(),
        plot.subtitle = element_text(color = "grey50"))

```

Multi-criterion comparison for optimal number

All scores normalised 0-1 · avg_endemism inverted so



Extract Final Bioregions

— SET YOUR CHOSEN k HERE

```
chosen_k    <- 13          # <-- change this number based on the plot above
cluster_col <- paste0("K_", chosen_k)
#
```

```
final_bioregions <- cut_tree(tree_to_eval, n_clust = chosen_k)
```

```
# Assign colors (requires bioregion >= v1.4.0)
colored_bioregions <- bioregion_colors(final_bioregions)
```

UPGMA Dendrogram with Cluster Cut Line

```
cut_height    <- tree_to_eval$algorithm$output_cut_height[paste0("k_",
chosen_k)]
```

```
# Map original Site Labels → cell numbers for dendrogram Leaf Labels
ordered_labels <- labels(as.dendrogram(tree_hclust))
ordered_clusters <- cutree(tree_hclust, k = chosen_k)[ordered_labels]
```

```

# Replace site IDs with cell numbers on the x-axis
label_map      <- setNames(cell_number_lookup$CellNumber,
                           cell_number_lookup$Site)
ordered_cell_nums <- label_map[ordered_labels]

# Build turbo palette directly from final_bioregions (no dependency on map
chunk)
cluster_levels <-
sort(unique(as.integer(final_bioregions$clusters[[cluster_col]])))
turbo_colours  <- setNames(
  viridis::turbo(length(cluster_levels)),
  as.character(cluster_levels)
)

dend_obj <- as.dendrogram(tree_hclust)
labels(dend_obj) <- ordered_cell_nums

par(mar = c(6, 4, 3, 2))
plot(dend_obj,
     main    = "UPGMA dendrogram – Balearic Polychaete Assemblages",
     sub     = paste0("Simpson dissimilarity · cut at h = ",
                      round(cut_height, 3), " → k = ", chosen_k, "
clusters"),
     ylab    = "Simpson dissimilarity",
     cex     = 0.7,
     leaflab = "perpendicular")

abline(h = cut_height, col = "red", lty = "dashed", lwd = 1.5)

# Cluster number labels above the cut line
cluster_positions <- tapply(seq_along(ordered_clusters), ordered_clusters,
mean)
for (cl in names(cluster_positions)) {
  text(x      = cluster_positions[cl],
       y      = cut_height + 0.015,
       labels = cl,
       col    = "red",
       cex    = 0.85,
       font   = 2)
}

# Shaded rectangles – colours matched to turbo_colours (same as map)
cluster_runs <- rle(ordered_clusters)
x_start      <- 0
for (i in seq_along(cluster_runs$lengths)) {
  cl_id <- as.character(cluster_runs$values[i])
  x_end <- x_start + cluster_runs$lengths[i]
  rect(xleft  = x_start + 0.5,
       xright = x_end    + 0.5,

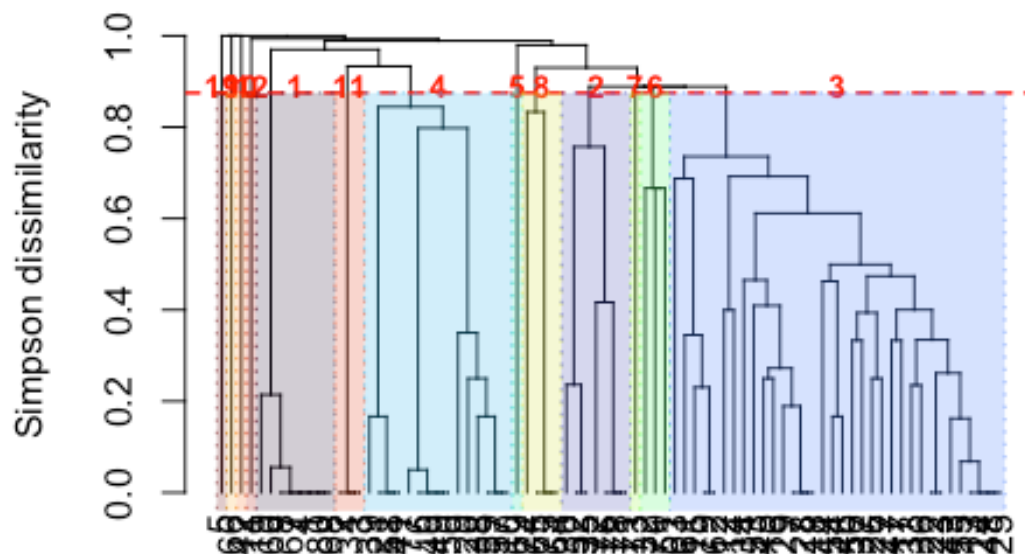
```

```

ybottom = -0.05,
ytop    = cut_height,
col      = adjustcolor(turbo_colours[cl_id], alpha.f = 0.25),
border   = adjustcolor(turbo_colours[cl_id], alpha.f = 0.6),
lty      = "dotted")
x_start <- x_end
}

```

JPGMA dendrogram — Balearic Polychaete Assembl



Simpson dissimilarity · cut at $h = 0.875 \rightarrow k = 13$ clusters

```
par(mar = c(5, 4, 4, 2))
```

Export dendrogram as PDF at 300 dpi

```

pdf("UPGMA_Dendrogram_Polychaetes.pdf",
    width = 20,      # inches – wide enough for 70 labels
    height = 10,     # inches
    useDingbats = FALSE)

par(mar = c(6, 4, 3, 2))
plot(dend_obj,
     main = "UPGMA dendrogram – Balearic Polychaete Assemblages",
     sub = paste0("Simpson dissimilarity · cut at h = ",
                  round(cut_height, 3), " → k = ", chosen_k, "
clusters"),

```

```

    ylab      = "Simpson dissimilarity",
    cex       = 0.7,
    leaflab   = "perpendicular")

abline(h = cut_height, col = "red", lty = "dashed", lwd = 1.5)

for (cl in names(cluster_positions)) {
  text(x      = cluster_positions[cl],
       y      = cut_height + 0.015,
       labels = cl,
       col    = "red",
       cex    = 0.85,
       font   = 2)
}

cluster_runs <- rle(ordered_clusters)
x_start      <- 0
for (i in seq_along(cluster_runs$lengths)) {
  cl_id <- as.character(cluster_runs$values[i])
  x_end  <- x_start + cluster_runs$lengths[i]
  rect(xleft  = x_start + 0.5,
       xright  = x_end   + 0.5,
       ybottom = -0.05,
       ytop    = cut_height,
       col     = adjustcolor(turbo_colours[cl_id], alpha.f = 0.25),
       border  = adjustcolor(turbo_colours[cl_id], alpha.f = 0.6),
       lty     = "dotted")
  x_start <- x_end
}

par(mar = c(5, 4, 4, 2))
dev.off()

## quartz_off_screen
##                2

cat("Dendrogram exported to UPGMA_Dendrogram_Polychaetes.pdf\n")

## Dendrogram exported to UPGMA_Dendrogram_Polychaetes.pdf

```

Isolates: single-site bioregions to flag (not hard-filter) in summaries

```

isolates <- c(5, 7, 9, 10, 12, 13)  # <-- update if your isolates change
with a different k

```

PCoA — Visual Inspection of Assemblage Scatter

```
# Run PCoA
pcoa_result <- pcoa(dist_for_pcoa)

# Variance explained
pcoa_result$values[1:5, ]

##   Eigenvalues Relative_eig Rel_corr_eig Broken_stick Cum_corr_eig
## 1    6.427636    0.19940681  0.03886627  0.06399351  0.03886627
## 2    3.084493    0.09569132  0.02438852  0.05100650  0.06325479
## 3    2.935996    0.09108442  0.02374544  0.04451299  0.08700023
## 4    2.443371    0.07580156  0.02161210  0.04018399  0.10861233
## 5    2.138376    0.06633959  0.02029129  0.03693724  0.12890362
##   Cumul_br_stick
## 1    0.06399351
## 2    0.11500001
## 3    0.15951301
## 4    0.19969700
## 5    0.23663423

# Site scores
pcoa_scores <- as.data.frame(pcoa_result$vectors[, 1:2])
colnames(pcoa_scores) <- c("PCoA1", "PCoA2")
pcoa_scores$Site <- rownames(pcoa_scores)

# Join cluster assignments
cluster_df <- as.data.frame(final_bioregions$clusters)
pcoa_scores <- left_join(pcoa_scores, cluster_df, by = c("Site" =
"ID"))
pcoa_scores$Bioregion <- factor(pcoa_scores[[cluster_col]])
pcoa_scores$Is_Isolate <- pcoa_scores$Bioregion %in% as.character(isolate)

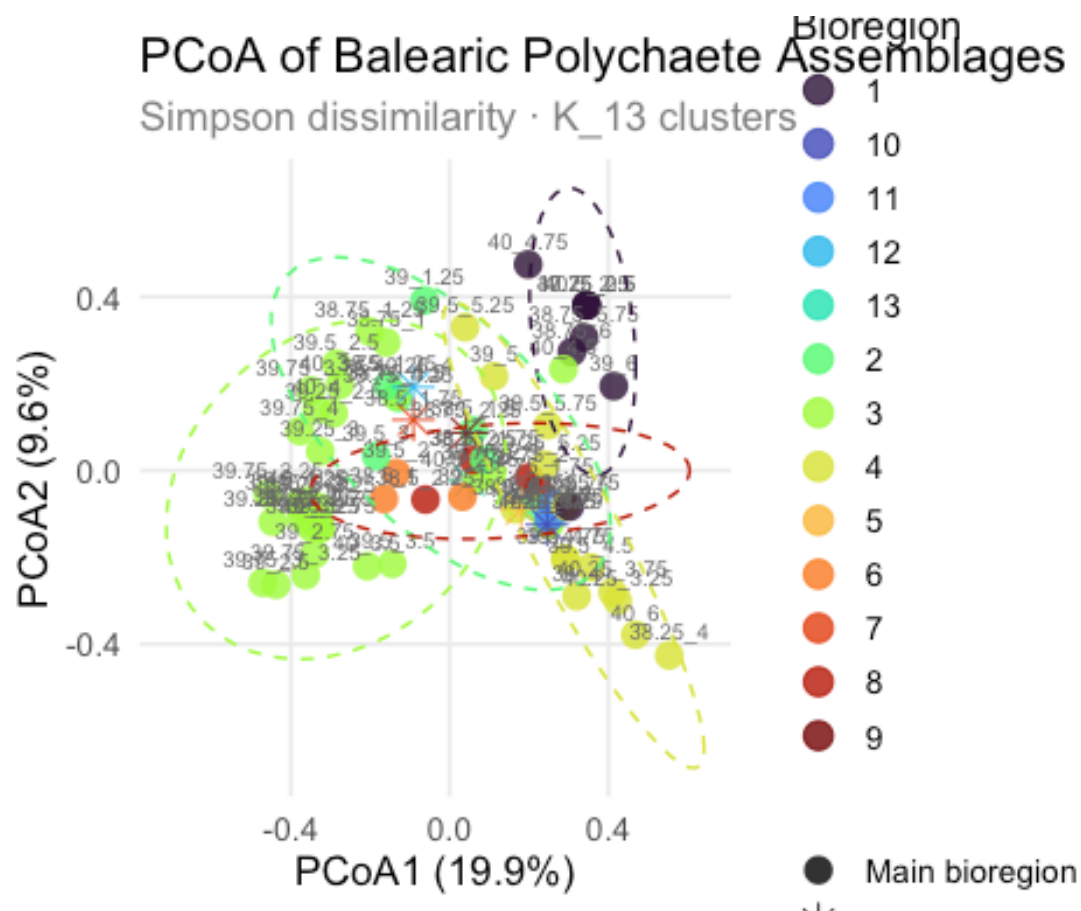
# Axis labels
var1 <- round(pcoa_result$values$Relative_eig[1] * 100, 1)
var2 <- round(pcoa_result$values$Relative_eig[2] * 100, 1)

ggplot(pcoa_scores, aes(x = PCoA1, y = PCoA2, color = Bioregion, shape =
Is_Isolate)) +
  geom_point(size = 4, alpha = 0.85) +
  geom_text(aes(label = Site), size = 2.5, vjust = -1,
    color = "grey40", show.legend = FALSE) +
  stat_ellipse(
    data = pcoa_scores[!pcoa_scores$Is_Isolate, ],
    aes(group = Bioregion, color = Bioregion),
    type = "t", level = 0.95, linetype = "dashed",
    linewidth = 0.5, show.legend = FALSE
  ) +
  scale_shape_manual(values = c("FALSE" = 16, "TRUE" = 8),
    labels = c("Main bioregion", "Isolate"),
```

```

      name = NULL) +
  scale_color_viridis_d(option = "turbo", name = "Bioregion") +
  labs(title = "PCoA of Balearic Polychaete Assemblages",
       subtitle = paste0("Simpson dissimilarity · ", cluster_col, "
clusters"),
       x = paste0("PCoA1 (", var1, "%)"),
       y = paste0("PCoA2 (", var2, "%)")) +
  theme_minimal(base_size = 13) +
  theme(legend.position = "right",
       plot.subtitle = element_text(color = "grey50"),
       panel.grid.minor = element_blank())

```



Create Spatial Objects

```

unique_grid_sites <- unique(df_clean[, c("Site", "Grid_Long", "Grid_Lat")])

balearic_sf <- st_as_sf(unique_grid_sites,
                        coords = c("Grid_Long", "Grid_Lat"),
                        crs = 4326)

```

Map the Assemblages

```

# Join bioregion clusters and cell numbers
bioregion_sf <- merge(occupied_grid_sf,

```

```

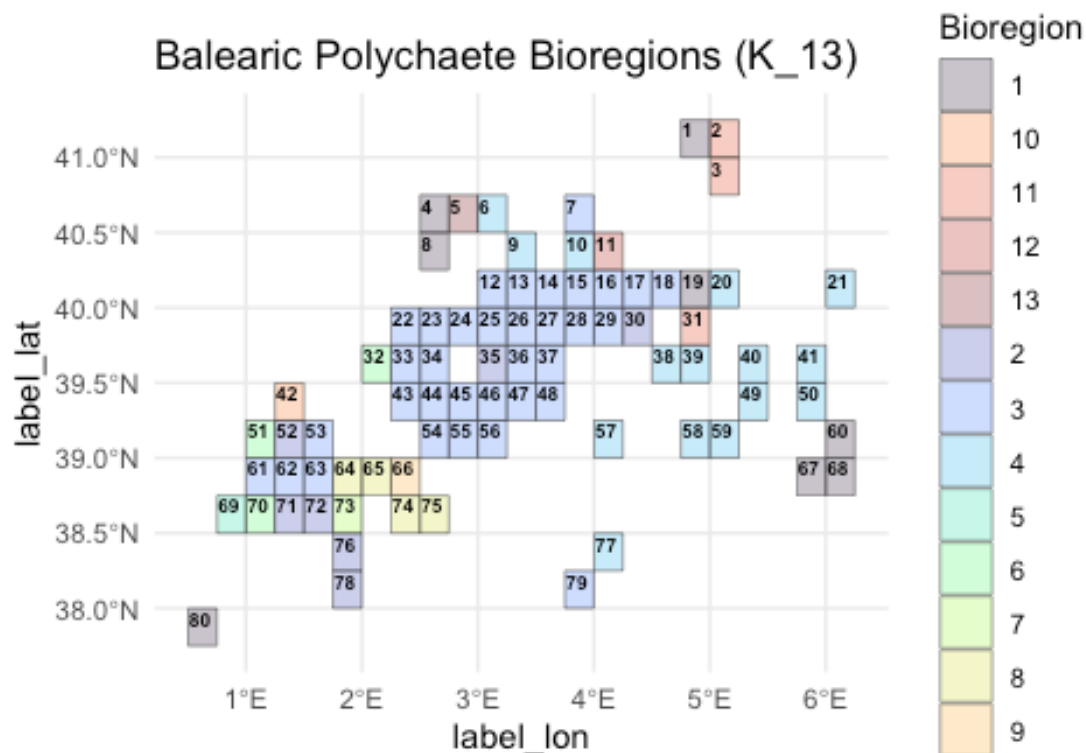
        final_bioregions$clusters,
        by.x = "Site",
        by.y = "ID") %>%
left_join(cell_number_lookup[, c("Site", "CellNumber")], by = "Site")

# Build turbo palette
cluster_levels <- sort(unique(as.integer(bioregion_sf[[cluster_col]])))
n_clusters_actual <- length(cluster_levels)
turbo_colours <- setNames(
  viridis::turbo(n_clusters_actual),
  as.character(cluster_levels)
)
turbo_colours_alpha <- setNames(
  adjustcolor(viridis::turbo(n_clusters_actual), alpha.f = 0.30),
  as.character(cluster_levels)
)

# Top-left corner label positions
bioregion_sf <- bioregion_sf %>%
  mutate(
    label_lon = Grid_Long + 0.01,
    label_lat = Grid_Lat + 0.25 - 0.02
  )

ggplot(bioregion_sf) +
  geom_sf(aes(fill = factor(.data[[cluster_col]])), colour = "grey30",
  linewidth = 0.2) +
  geom_text(
    aes(x = label_lon, y = label_lat, label = CellNumber),
    size = 2.2,
    hjust = 0,
    vjust = 1,
    colour = "black",
    fontface = "bold",
    inherit.aes = FALSE
  ) +
  scale_fill_manual(
    values = turbo_colours_alpha,
    name = "Bioregion"
  ) +
  theme_minimal() +
  labs(
    title = paste0("Balearic Polychaete Bioregions (", cluster_col, ")")
  )

```



Export to QGIS

```
bioregion_sf[[cluster_col]] <- as.character(bioregion_sf[[cluster_col]])
bioregion_sf$Bioregion_label <- sprintf("Bioregion_%02d",
as.integer(bioregion_sf[[cluster_col]]))
bioregion_sf$hex_colour <- turbo_colours[bioregion_sf[[cluster_col]]]

st_write(bioregion_sf,
  "Balearic_Polychaete_Bioregions.gpkg",
  delete_dsn = TRUE)

## Deleting source `Balearic_Polychaete_Bioregions.gpkg' using driver `GPKG'
## Writing layer `Balearic_Polychaete_Bioregions' to data source
## `Balearic_Polychaete_Bioregions.gpkg' using driver `GPKG'
## Writing 80 features with 11 fields and geometry type Polygon.

# Build a .qml sidecar file so QGIS loads the exact turbo colours
# automatically
# One <category> entry per bioregion
category_xml <- mapply(function(k, hex) {
  label <- sprintf("Bioregion_%02d", as.integer(k))
  # Convert hex to 0-255 RGB for QGIS
  r <- strttoi(substr(hex, 2, 3), 16L)
```

```

g <- strtoi(substr(hex, 4, 5), 16L)
b <- strtoi(substr(hex, 6, 7), 16L)
sprintf(
  '    <category value="%s" label="%s" render="true" symbol="%s"/>',
  k, label, k
)
}, names(turbo_colours), turbo_colours)

symbol_xml <- mapply(function(k, hex) {
  r <- strtoi(substr(hex, 2, 3), 16L)
  g <- strtoi(substr(hex, 4, 5), 16L)
  b <- strtoi(substr(hex, 6, 7), 16L)
  sprintf(
    '    <symbol name="%s" type="fill" alpha="1" clip_to_extent="1"
force_rhr="0">
      <layer class="SimpleFill" enabled="1" pass="0" locked="0">
        <Option type="Map">
          <Option name="color" value="%d,%d,%d,255" type="QString"/>
          <Option name="style" value="solid" type="QString"/>
          <Option name="outline_color" value="77,77,77,255" type="QString"/>
          <Option name="outline_width" value="0.1" type="QString"/>
        </Option>
      </layer>
    </symbol>',
    k, r, g, b
  )
}, names(turbo_colours), turbo_colours)

qml <- paste0(
  '<!DOCTYPE qgis PUBLIC "http://mrcc.com/qgis.dtd" "SYSTEM">
<qgis version="4.0.0" styleCategories="Symbolology">
  <renderer-v2 type="categorizedSymbol" attr="", cluster_col, ""
enableorderby="0" forceraster="0" symbollevels="0" usingSymbolLevels="0">
    <categories>
', paste(category_xml, collapse = "\n"), '
    </categories>
    <symbols>
', paste(symbol_xml, collapse = "\n"), '
    </symbols>
  </renderer-v2>
</qgis>'
)

qml <- paste0(
  '<!DOCTYPE qgis PUBLIC "http://mrcc.com/qgis.dtd" "SYSTEM">
<qgis version="4.0.0" styleCategories="Symbolology|Labeling">
  <renderer-v2 type="categorizedSymbol" attr="", cluster_col, ""
enableorderby="0" forceraster="0" symbollevels="0" usingSymbolLevels="0">
    <categories>

```

```

', paste(category_xml, collapse = "\n"), '
  </categories>
  <symbols>
', paste(symbol_xml, collapse = "\n"), '
  </symbols>
</renderer-v2>
<labeling type="simple">
  <settings calloutType="simple">
    <text-style fontFamily="Arial" fontSize="7" textColor="0,0,0,255"
      fontWeight="75" fontItalic="0" textOpacity="1"
      fieldName="CellNumber" isExpression="0"
      namedStyle="Bold" blendMode="0"
previewBkgrdColor="255,255,255,255"/>
    <text-buffer bufferDraw="0" bufferSize="0.5" bufferColor="0,0,0,255"
      bufferOpacity="0.5" bufferBlendMode="0"
bufferJoinStyle="128"
      bufferNoFill="1" bufferSizeUnits="MM"/>
    <placement placement="1" quadOffset="2" offsetX="0" offsetY="0"
      xOffset="0" yOffset="0" units="MM"
      placementFlags="10" overrunDistance="0"
      overrunDistanceUnit="MM"
overrunDistanceMapUnitScale="3x:0,0,0,0,0,0"/>
    <rendering drawLabels="1" scaleVisibility="0" minScale="0" maxScale="0"
      obstacle="0" obstacleFactor="1" obstacleType="0"
      zIndex="0" limitNumLabels="0" maxNumLabels="2000"
      mergeLines="0" minFeatureSize="0" labelPerPart="0"/>
  </settings>
</labeling>
</qgis>'
)

writeLines(qml, "Balearic_Polychaete_Bioregions.qml")
cat("Wrote Balearic_Polychaete_Bioregions.qml – place it in the same folder
as the .gpkg\n")

## Wrote Balearic_Polychaete_Bioregions.qml – place it in the same folder as
the .gpkg

```

Sampling Intensity as polygon grid

```

intensity_counts      <- as.data.frame(table(df_geo$Site))
colnames(intensity_counts) <- c("Site", "DataPoints")

intensity_grid_sf <- merge(occupied_grid_sf,
                           intensity_counts[-35, ], #Excluding this outlier,
                           by = "Site")

pdf(file = "Records per Cell.pdf", width=6, height=4)

plot(intensity_grid_sf["DataPoints"],

```

```

    main    = "Sampling Intensity (Total Data Points per 0.25° Cell)",
    pch     = 15,
    cex     = 1.5,
    pal     = hcl.colors(10, "viridis", rev = TRUE),
    breaks  = "equal")

dev.off()

## quartz_off_screen
##                2

st_write(intensity_grid_sf, "Polychaete_Sampling_Intensity.gpkg", delete_dsn
= TRUE)

## Deleting source `Polychaete_Sampling_Intensity.gpkg' using driver `GPKG'
## Writing layer `Polychaete_Sampling_Intensity' to data source
## `Polychaete_Sampling_Intensity.gpkg' using driver `GPKG'
## Writing 79 features with 6 fields and geometry type Polygon.

```

Species Richness per Grid Cell

Calculate Species Richness per Grid Cell

```

# Remove duplicate Site-Species records so each species counts once per cell
species_per_site <- distinct(df_clean, Site, accepted_taxon)

```

```

# Count the number of species per cell

```

```

richness_counts <- as.data.frame(table(species_per_site$Site))
colnames(richness_counts) <- c("Site", "SpeciesRichness")

```

```

richness_grid_sf <- merge(occupied_grid_sf,
                          richness_counts[-35, ], #Excluding this outlier
                          by = "Site")

```

```

# Map

```

```

pdf(file = "Species richness per Cell.pdf", width=6, height=4)

```

```

#par(bg = "gray95") #Grey background (Optional)

```

```

plot(richness_grid_sf["SpeciesRichness"],
     main    = "Species Richness",
     pch     = 15,
     cex     = 1.5,
     pal     = hcl.colors(10, "viridis", rev = TRUE),
     breaks  = "equal")

```

```

dev.off()

```

```

## quartz_off_screen
##                2

```

```

st_write(richness_grid_sf, "Polychaete_Species_Richness.gpkg", delete_dsn =
TRUE)

```

```
## Deleting source `Polychaete_Species_Richness.gpkg' using driver `GPKG'
## Writing layer `Polychaete_Species_Richness' to data source
## `Polychaete_Species_Richness.gpkg' using driver `GPKG'
## Writing 79 features with 6 fields and geometry type Polygon.
```

Bioregion Metrics

```
# Presence-absence input
occurrence_data <- df_clean %>% distinct(Site, accepted_taxon)

# Sampling effort per site
site_effort <- df_clean %>% count(Site, name = "NbDataPoints")

# Cluster assignments
cluster_map <- final_bioregions$clusters %>%
  select(ID, all_of(cluster_col))

# — Complete master table — ALL bioregions including isolates


---


nb_sites_all <- cluster_map %>%
  group_by(.data[[cluster_col]]) %>%
  summarise(NbSites = n(), .groups = "drop") %>%
  rename(Bioregion = all_of(cluster_col))

richness_all <- occurrence_data %>%
  left_join(cluster_map, by = c("Site" = "ID")) %>%
  group_by(.data[[cluster_col]]) %>%
  summarise(Richness = n_distinct(accepted_taxon), .groups = "drop") %>%
  rename(Bioregion = all_of(cluster_col))

effort_all <- cluster_map %>%
  left_join(site_effort, by = c("ID" = "Site")) %>%
  group_by(.data[[cluster_col]]) %>%
  summarise(NbDataPoints = sum(NbDataPoints), .groups = "drop") %>%
  rename(Bioregion = all_of(cluster_col))

# Endemism
data_with_clusters <- occurrence_data %>%
  left_join(cluster_map, by = c("Site" = "ID"))

species_bioregion_span <- data_with_clusters %>%
  group_by(accepted_taxon) %>%
  summarise(n_bioregions = n_distinct(.data[[cluster_col]]), .groups =
"drop")

endemic_species <- species_bioregion_span %>%
  filter(n_bioregions == 1) %>%
  pull(accepted_taxon)
```

```

final_endemism_table <- data_with_clusters %>%
  filter(accepted_taxon %in% endemic_species) %>%
  group_by(.data[[cluster_col]]) %>%
  summarise(Rich_Endemics = n_distinct(accepted_taxon), .groups = "drop") %>%
  rename(Bioregion = 1)

# Join all metrics
master_table_complete <- nb_sites_all %>%
  left_join(richness_all, by = "Bioregion") %>%
  left_join(effort_all, by = "Bioregion") %>%
  left_join(final_endemism_table, by = "Bioregion") %>%
  mutate(
    Rich_Endemics = coalesce(Rich_Endemics, 0L),
    Isolate = Bioregion %in% as.character(isolate),
    Bioregion_num = as.integer(Bioregion)
  ) %>%
  arrange(Bioregion_num) %>%
  select(-Bioregion_num)

print(master_table_complete)

## # A tibble: 13 × 6
##   Bioregion NbSites Richness NbDataPoints Rich_Endemics Isolate
##   <chr>      <int>   <int>      <int>      <int> <lgl>
## 1 1         8       5        65         4 FALSE
## 2 2         7     189       794        28 FALSE
## 3 3        34     439     6546       234 FALSE
## 4 4        15     171     684        82 FALSE
## 5 5         1       3         6         0 TRUE
## 6 6         3      28        51         6 FALSE
## 7 7         1       3         3         0 TRUE
## 8 8         4      13        26         3 FALSE
## 9 9         1       1         1         1 TRUE
## 10 10        1       1         1         1 TRUE
## 11 11        3       2        12         0 FALSE
## 12 12        1       2         2         1 TRUE
## 13 13        1       1         1         1 TRUE

```

Endemic Species List per Bioregion

```

# Get family information from original dataframe
family_lookup <- DF_Assemblages %>%
  distinct(accepted_taxon, family)

# Build full endemic species list with bioregion and family
endemic_list <- data_with_clusters %>%
  filter(accepted_taxon %in% endemic_species) %>%
  distinct(accepted_taxon, .data[[cluster_col]]) %>%
  left_join(family_lookup, by = "accepted_taxon") %>%
  rename(Bioregion = all_of(cluster_col)) %>%

```

```

mutate(Bioregion_num = as.integer(Bioregion)) %>%
arrange(Bioregion_num, family, accepted_taxon) %>%
select(Bioregion, family, accepted_taxon)

# Print summary count to console
cat("Total endemic species across all bioregions:",
n_distinct(endemic_list$accepted_taxon), "\n")

## Total endemic species across all bioregions: 361

print(endemic_list)

##      Bioregion      family
accepted_taxon
## 1          1      Oenonidae
Oenone
## 2          1      Phyllodocidae
Rhynchonereella
## 3          1      Tomopteridae
Tomopteris
## 4          1      Typhloscolecidae
Sagitella
## 5          2      Aphroditidae      Aphrodita
alta
## 6          2      Cirratulidae      Caulleriella
alata
## 7          2      Dorvilleidae      Ophryotrocha
labronica
## 8          2      Fabriciidae      Pseudofabriciola
filamentosa
## 9          2      Maldanidae      Euclymene
collaris
## 10         2      Maldanidae      Euclymene
lombricoides
## 11         2      Maldanidae      Johnstonia
clymenoides
## 12         2      Phyllodocidae      Eulalia
expusilla
## 13         2      Phyllodocidae      Eumida
parva
## 14         2      Phyllodocidae      Lugia
pterophora
## 15         2      Phyllodocidae      Mystides
borealis
## 16         2      Polynoidae      Harmothoe
aspera
## 17         2      Polynoidae      Harmothoe
gilchristi
## 18         2      Sabellidae      Bispira
volutacornis

```

## 19	2	Sabellidae	Branchiomma
luctuosum			
## 20	2	Serpulidae	Pileolaria
berkeleyana			
## 21	2	Serpulidae	Pileolaria
heteropoma			
## 22	2	Serpulidae	Semivermilia
cribrata			
## 23	2	Serpulidae	Simplaria
pseudomilitaris			
## 24	2	Serpulidae	Spirorbis (Spirorbis)
marioni			
## 25	2	Serpulidae	Vinearina
koehleri			
## 26	2	Sphaerodoridae	Sphaerodoridium
claparedii			
## 27	2	Spionidae	Dispio
uncinata			
## 28	2	Spionidae	Malacoceros
girardi			
## 29	2	Spionidae	Spio
martinensis			
## 30	2	Terebellidae	Polycirrus
denticulatus			
## 31	2	Terebellidae	Polycirrus
haematodes			
## 32	2	Trichobranchidae	Octobranchus
lingulatus			
## 33	3	Aberrantidae	
Aberranta			
## 34	3	Acrocirridae	Acrocirrus
frontifilis			
## 35	3	Ampharetidae	Auchenoplax
crinita			
## 36	3	Ampharetidae	
Sosane			
## 37	3	Amphinomidae	Cryptonome
conclava			
## 38	3	Aphroditidae	
Hermione			
## 39	3	Arenicolidae	Branchiomaldane
vincentii			
## 40	3	Capitellidae	Dasybranchus
caducus			
## 41	3	Capitellidae	Leiochrides
australis			
## 42	3	Capitellidae	Mastobranchus
trinchessii			
## 43	3	Capitellidae	Mediomastus
fragilis			

## 44	3	Capitellidae	
Neopseudocapitella			
## 45	3	Capitellidae	Notomastus
aberans			
## 46	3	Capitellidae	Notomastus
latericeus			
## 47	3	Capitellidae	Notomastus
lineatus			
## 48	3	Capitellidae	Peresiella
clymenoides			
## 49	3	Capitellidae	
Pseudocapitella			
## 50	3	Capitellidae	Pseudoleiocapitella
fauveli			
## 51	3	Chaetopteridae	Phyllochaetopterus
anglicus			
## 52	3	Chaetopteridae	Spiochaetopterus
typicus			
## 53	3	Chrysopetalidae	Paleanotus
chrysolepis			
## 54	3	Cirratulidae	Caulleriella
mediterranea			
## 55	3	Cirratulidae	Caulleriella
parva			
## 56	3	Cirratulidae	Chaetozone
carpenteri			
## 57	3	Cirratulidae	Chaetozone
gibber			
## 58	3	Cirratulidae	Chaetozone
vivipara			
## 59	3	Cirratulidae	Cirratulus
cirratus			
## 60	3	Cirratulidae	Cirriformia
afer			
## 61	3	Cirratulidae	Drilonereis
filum			
## 62	3	Cirratulidae	Kirkegaardia
heterochaeta			
## 63	3	Cirratulidae	Kirkegaardia
marypetersenae			
## 64	3	Cossuridae	
Cossura			
## 65	3	Dorvilleidae	Dorvillea
similis			
## 66	3	Dorvilleidae	Parougia
albomaculata			
## 67	3	Dorvilleidae	Pettiboneia
urciensis			
## 68	3	Dorvilleidae	Schistomeringos
neglecta			

## 69	3	Eunicidae	Eunice
pennata			
## 70	3	Eunicidae	Eunice
schizobranchia			
## 71	3	Eunicidae	Eunice
vittata			
## 72	3	Eunicidae	Leodice
harassii			
## 73	3	Eunicidae	Lysidice
hebes			
## 74	3	Eunicidae	Paucibranchia
bellii			
## 75	3	Eunicidae	Paucibranchia
fallax			
## 76	3	Euphrosinidae	Euphrosine
cirrata			
## 77	3	Fabriciidae	Novafabricia
infratorquata			
## 78	3	Fabriciidae	Pseudofabricia
aberrans			
## 79	3	Fabriciidae	Pseudofabriciola
capensis			
## 80	3	Flabelligeridae	Flabelligera
diplochaitus			
## 81	3	Flabelligeridae	
Pherusa			
## 82	3	Flabelligeridae	Piromis
eruca			
## 83	3	Glyceridae	Glycera
alba			
## 84	3	Glyceridae	Glycera
fallax			
## 85	3	Glyceridae	Glycera
oxycephala			
## 86	3	Glyceridae	Glycera
tridactyla			
## 87	3	Glyceridae	Glycera
unicornis			
## 88	3	Goniadidae	Goniadella
galaica			
## 89	3	Hesionidae	Gyptis
propinqua			
## 90	3	Hesionidae	Hesione
pantherina			
## 91	3	Hesionidae	Hesiospina
aurantiaca			
## 92	3	Hesionidae	Nereimyra
punctata			
## 93	3	Hesionidae	Psamathe
fusca			

## 94	3	Lacydoniidae	Lacydonia
miranda			
## 95	3	Lumbrineridae	Abyssoninoe
hibernica			
## 96	3	Lumbrineridae	Gallardoneris
nonatoi			
## 97	3	Lumbrineridae	Hilbigneris
gracilis			
## 98	3	Lumbrineridae	Lumbrineriopsis
paradoxa			
## 99	3	Lumbrineridae	Scoletoma
fragilis			
## 100	3	Magelonidae	Magelona
equilamellae			
## 101	3	Maldanidae	Axiothella
constricta			
## 102	3	Maldanidae	Euclymene
oerstedii			
## 103	3	Maldanidae	Leiochone
leiopygos			
## 104	3	Maldanidae	Praxillella
praetermissa			
## 105	3	Maldanidae	
Praxillura			
## 106	3	Myzostomatidae	
Myzostoma			
## 107	3	Nephtyidae	Aglaophamus
agilis			
## 108	3	Nephtyidae	Micronephthys
longicornis			
## 109	3	Nephtyidae	Nephtys
cirrosa			
## 110	3	Nephtyidae	Nephtys
hombergii			
## 111	3	Nereididae	Composetia
hircinicola			
## 112	3	Nereididae	Hediste
diversicolor			
## 113	3	Nereididae	Micronereis
variegata			
## 114	3	Nereididae	Neanthes
fucata			
## 115	3	Nereididae	Neanthes
rubicunda			
## 116	3	Nereididae	Nereis
pulsatoria			
## 117	3	Nereididae	Perinereis
kaustiana			
## 118	3	Nereididae	Perinereis
nieri			

## 119	3	Nereididae	Platynereis
agilis			
## 120	3	Nereididae	
Sinonereis			
## 121	3	Oenonidae	Arabella
iricolor			
## 122	3	Oenonidae	Halla
parthenopeia			
## 123	3	Oenonidae	
Labrorostratus			
## 124	3	Onuphidae	Aponuphis
brementi			
## 125	3	Onuphidae	Aponuphis
ornata			
## 126	3	Onuphidae	Nothria
britannica			
## 127	3	Onuphidae	Nothria
conchylega			
## 128	3	Onuphidae	Onuphis
eremita			
## 129	3	Opheliidae	Armandia
cirrhusa			
## 130	3	Opheliidae	Armandia
polyophthalma			
## 131	3	Orbiniidae	Naineris
quadraticeps			
## 132	3	Orbiniidae	Phylo
foetida			
## 133	3	Orbiniidae	Scoloplos
typicus			
## 134	3	Orbiniidae	
Scoloplos\xcaarmiger			
## 135	3	Oweniidae	Myriochele
heeri			
## 136	3	Oweniidae	Owenia
fusiformis			
## 137	3	Paraonidae	Aricidea (Acmira)
assimilis			
## 138	3	Paraonidae	Aricidea (Acmira)
catherinae			
## 139	3	Paraonidae	Aricidea (Acmira)
simonae			
## 140	3	Paraonidae	Aricidea (Aricidea)
bansei			
## 141	3	Paraonidae	Aricidea (Strelzovia)
claudiae			
## 142	3	Paraonidae	Aricidea
similis			
## 143	3	Paraonidae	Cirrophorus
furcatus			

## 144	3	Paraonidae	Cirrophorus
nikebianchii			
## 145	3	Paraonidae	Paradoneis
armata			
## 146	3	Paraonidae	Paradoneis
ilvana			
## 147	3	Paraonidae	
Paraonides			
## 148	3	Paraonidae	
Paraonis			
## 149	3	Pectinariidae	Amphictene
auricoma			
## 150	3	Pectinariidae	Lagis
koreni			
## 151	3	Pectinariidae	
Pectinaria			
## 152	3	Phyllodocidae	
Anaitides			
## 153	3	Phyllodocidae	Eteone
flava			
## 154	3	Phyllodocidae	Eteone
longa			
## 155	3	Phyllodocidae	Eulalia
clavigera			
## 156	3	Phyllodocidae	Eulalia
fuscescens			
## 157	3	Phyllodocidae	Eulalia
mustela			
## 158	3	Phyllodocidae	Eulalia
pusilla			
## 159	3	Phyllodocidae	Hesionura
coineaui			
## 160	3	Phyllodocidae	Nereiphylla
castanea			
## 161	3	Phyllodocidae	Nereiphylla
paretti			
## 162	3	Phyllodocidae	Notophyllum
foliosum			
## 163	3	Phyllodocidae	Phyllodoce
longipes			
## 164	3	Phyllodocidae	Phyllodoce
mucosa			
## 165	3	Phyllodocidae	Pseudomystides
spinachia			
## 166	3	Phyllodocidae	Pterocirrus
limbatus			
## 167	3	Pilargidae	Sigambra
parva			
## 168	3	Poecilochaetidae	Poecilochaetus
serpens			

## 169	3	Polygordiidae	
Polygordius			
## 170	3	Polynoidae	Acanthicolepis
asperrima			
## 171	3	Polynoidae	Acholoe
squamosa			
## 172	3	Polynoidae	Adyte
hyalina			
## 173	3	Polynoidae	Alentia
gelatinosa			
## 174	3	Polynoidae	Harmothoe
aequespina			
## 175	3	Polynoidae	Harmothoe
antilopes			
## 176	3	Polynoidae	Harmothoe
imbricata			
## 177	3	Polynoidae	Harmothoe
impar			
## 178	3	Polynoidae	Harmothoe
longisetis			
## 179	3	Polynoidae	Lepidasthenia
elegans			
## 180	3	Polynoidae	Leucia
nivea			
## 181	3	Polynoidae	Malmgrenia
andreapolis			
## 182	3	Polynoidae	Malmgrenia
liliana			
## 183	3	Polynoidae	Malmgrenia
ljungmani			
## 184	3	Polynoidae	Malmgrenia
lunulata			
## 185	3	Polynoidae	Pollentia
perezi			
## 186	3	Polynoidae	Subadyte
pellucida			
## 187	3	Protodrilidae	
Protodrilus			
## 188	3	Sabellariidae	Phalacrostemma
cidariophilum			
## 189	3	Sabellariidae	Sabellaria
alveolata			
## 190	3	Sabellariidae	Sabellaria
spinulosa			
## 191	3	Sabellidae	Bispira
viola			
## 192	3	Sabellidae	Branchiomma
moebii			
## 193	3	Sabellidae	Chone
filicaudata			

## 194	3	Sabellidae	Chone
infundibuliformis			
## 195	3	Sabellidae	Dialychone
dunerificta			
## 196	3	Sabellidae	
Euchone			
## 197	3	Sabellidae	Jasmineira
caudata			
## 198	3	Sabellidae	Jasmineira
elegans			
## 199	3	Sabellidae	Myxicola
infundibulum			
## 200	3	Sabellidae	Notaulax
phaeotaenia			
## 201	3	Sabellidae	Paradialychone
gambiae			
## 202	3	Sabellidae	Parasabella
saxicola			
## 203	3	Sabellidae	Pseudopotamilla
reniformis			
## 204	3	Sabellidae	Sabella
discifera			
## 205	3	Sabellidae	Sabella
pavonina			
## 206	3	Scalibregmatidae	Sclerocheilus
minutus			
## 207	3	Serpulidae	Ditrupa
arietina			
## 208	3	Serpulidae	Hydroides
dirampha			
## 209	3	Serpulidae	Hydroides
helmata			
## 210	3	Serpulidae	Hydroides
nigra			
## 211	3	Serpulidae	Hydroides
norvegica			
## 212	3	Serpulidae	Protolaeospira (Protolaeospira)
striata			
## 213	3	Serpulidae	Protula
intestinum			
## 214	3	Serpulidae	Salmacina
dysteri			
## 215	3	Serpulidae	Semivermilia
crenata			
## 216	3	Serpulidae	Spiraserpula
massiliensis			
## 217	3	Serpulidae	Spirobranchus
tetraceros			
## 218	3	Serpulidae	Vermiliopsis
labiata			

## 219	3	Sigalionidae	Fimbriosthenelais
minor			
## 220	3	Sigalionidae	Pelogenia
arenosa			
## 221	3	Sigalionidae	Pholoe
inornata			
## 222	3	Sigalionidae	Pisione
guanche			
## 223	3	Sigalionidae	Pisione
puzae			
## 224	3	Sigalionidae	Sigalion
mathildae			
## 225	3	Sigalionidae	Sigalion
squamosus			
## 226	3	Sigalionidae	Sthenelais
boa			
## 227	3	Spionidae	Aonides
oxycephala			
## 228	3	Spionidae	Aonides
paucibranchiata			
## 229	3	Spionidae	Boccardia
polybranchia			
## 230	3	Spionidae	Dipolydora
coeca			
## 231	3	Spionidae	Dipolydora
giardi			
## 232	3	Spionidae	Dipolydora
langerhansi			
## 233	3	Spionidae	
Polydora			
## 234	3	Spionidae	Prionospio
cirrifera			
## 235	3	Spionidae	Prionospio
dubia			
## 236	3	Spionidae	Prionospio
fallax			
## 237	3	Spionidae	Prionospio
maciolekae			
## 238	3	Spionidae	Prionospio
multibranchiata			
## 239	3	Spionidae	Prionospio
steenstrupi			
## 240	3	Spionidae	Pseudopolydora
paucibranchiata			
## 241	3	Spionidae	Pygospio
elegans			
## 242	3	Spionidae	Scoelelepis
neglecta			
## 243	3	Spionidae	Scoelelepis
squamata			

## 244	3	Spionidae	Spio
decorata			
## 245	3	Spionidae	Spiophanes
bombyx			
## 246	3	Syllidae	Haplosyllis
granulosa			
## 247	3	Syllidae	Odontosyllis fulgurans
dolerens			
## 248	3	Syllidae	Syllis
bouvieri			
## 249	3	Syllidae	Syllis
lutea			
## 250	3	Syllidae	Trypanedenta
gemmipara			
## 251	3	Terebellidae	
Amphitrite			
## 252	3	Terebellidae	Euthelepus
setubalensis			
## 253	3	Terebellidae	Lanice
conchilega			
## 254	3	Terebellidae	Loimia
medusa			
## 255	3	Terebellidae	Lysilla
nivea			
## 256	3	Terebellidae	Pista
cretacea			
## 257	3	Terebellidae	Pista
labrunaeae			
## 258	3	Terebellidae	Polycirrus
burballes			
## 259	3	Terebellidae	Polycirrus
medusa			
## 260	3	Terebellidae	Polycirrus
pallidus			
## 261	3	Terebellidae	
Streblosoma			
## 262	3	Terebellidae	Thelepus
cincinnatus			
## 263	3	Terebellidae	Thelepus
setosus			
## 264	3	Trichobranchidae	
Terebellides			
## 265	3	Trichobranchidae	Trichobranchus
glacialis			
## 266	3	Trichobranchidae	
Trichobranchus\xcaroseus			
## 267	4	Lopadorrhynchidae	Lopadorrhynchus
uncinatus			
## 268	4	Phyllodocidae	Naiades
cantrainii			

## 269	4	Phyllodocidae	Paranaitis
kosteriensis			
## 270	4	Phyllodocidae	Vanadis
crystallina			
## 271	4	Phyllodocidae	Vanadis
formosa			
## 272	4	Serpulidae	
Hyalopomatus			
## 273	4	Serpulidae	
Protis			
## 274	4	Syllidae	Amblyosyllis
madeirensis			
## 275	4	Syllidae	Autolytus
neapolitanus			
## 276	4	Syllidae	Autolytus
rubrovittatus			
## 277	4	Syllidae	Brevicirrosyllis
weismanni			
## 278	4	Syllidae	Dioplosyllis
cirrosa			
## 279	4	Syllidae	Epigamia
alexandri			
## 280	4	Syllidae	Epigamia
macrophthalma			
## 281	4	Syllidae	Erinaceusyllis
belizensis			
## 282	4	Syllidae	Erinaceusyllis
cryptica			
## 283	4	Syllidae	Erinaceusyllis
erinaceus			
## 284	4	Syllidae	Exogone (Exogone)
torulosa			
## 285	4	Syllidae	Exogone (Parexogone)
gambiae			
## 286	4	Syllidae	Exogone
brevipes			
## 287	4	Syllidae	Irmula
spissipes			
## 288	4	Syllidae	Levidorum
pori			
## 289	4	Syllidae	Miscellania
dentata			
## 290	4	Syllidae	Myrianida
convoluta			
## 291	4	Syllidae	Myrianida
dentalia			
## 292	4	Syllidae	Myrianida
edwarsi			
## 293	4	Syllidae	Myrianida
inermis			

## 294	4	Syllidae	Myrianida
longoprimiticirrata			
## 295	4	Syllidae	Myrianida
quindecimdentata			
## 296	4	Syllidae	Myrianida
rubropunctata			
## 297	4	Syllidae	Neopetitia
amphophthalma			
## 298	4	Syllidae	Nudisyllis
divaricata			
## 299	4	Syllidae	Odontosyllis
dugesiana			
## 300	4	Syllidae	Opisthodonta
morena			
## 301	4	Syllidae	Opisthosyllis
brunnea			
## 302	4	Syllidae	Palposyllis
prosostoma			
## 303	4	Syllidae	Paraehlersia
dionisi			
## 304	4	Syllidae	Paralcyonosyllis
chamaeleon			
## 305	4	Syllidae	Parapionosyllis
gestans			
## 306	4	Syllidae	Parapionosyllis
papillosa			
## 307	4	Syllidae	Paraprocerastea
crocantinae			
## 308	4	Syllidae	Paratyposyllis
peresi			
## 309	4	Syllidae	Parexogone
caribensis			
## 310	4	Syllidae	Parexogone
cognettii			
## 311	4	Syllidae	Parexogone
convoluta			
## 312	4	Syllidae	Parexogone
hebes			
## 313	4	Syllidae	Parexogone
wolfi			
## 314	4	Syllidae	Perkinsyllis
anophthalma			
## 315	4	Syllidae	Procerastea
halleziana			
## 316	4	Syllidae	Procerastea
nematodes			
## 317	4	Syllidae	Prosphaerosyllis
brandhorsti			
## 318	4	Syllidae	Prosphaerosyllis
brevicirra			

## 319	4	Syllidae	Prosphaerosyllis
giandoi			
## 320	4	Syllidae	Prosphaerosyllis
tetralix			
## 321	4	Syllidae	Salvatoria
balani			
## 322	4	Syllidae	Salvatoria
celiae			
## 323	4	Syllidae	Salvatoria
euritmica			
## 324	4	Syllidae	Salvatoria
swedmarki			
## 325	4	Syllidae	Salvatoria
tenuicirrata			
## 326	4	Syllidae	Sphaerosyllis
bulbosa			
## 327	4	Syllidae	Sphaerosyllis
claparedei			
## 328	4	Syllidae	Sphaerosyllis
glandulata			
## 329	4	Syllidae	Sphaerosyllis
magnidentata			
## 330	4	Syllidae	Sphaerosyllis
ovigera			
## 331	4	Syllidae	Sphaerosyllis
piriferopsis			
## 332	4	Syllidae	Streptodonta
pterochaeta			
## 333	4	Syllidae	Streptosyllis
arenae			
## 334	4	Syllidae	Streptosyllis
bidentata			
## 335	4	Syllidae	Syllides
articulosus			
## 336	4	Syllidae	Syllides
bansei			
## 337	4	Syllidae	Syllides
convolutus			
## 338	4	Syllidae	Syllides
longocirratus			
## 339	4	Syllidae	Syllis
compacta			
## 340	4	Syllidae	Syllis
cruzi			
## 341	4	Syllidae	Syllis
nigricirris			
## 342	4	Syllidae	Syllis
parapari			
## 343	4	Syllidae	Syllis
pectinans			

## 344	4	Syllidae	Syllis
pontxioi			
## 345	4	Syllidae	Syllis
pulvinata			
## 346	4	Syllidae	Syllis
schulzi			
## 347	4	Syllidae	Syllis
tyrrhena			
## 348	4	Syllidae	Virchowia
clavata			
## 349	6	Ampharetidae	Ampharete
lindstroemi			
## 350	6	Ampharetidae	Lysippe
labiata			
## 351	6	Goniadidae	Goniada
maculata			
## 352	6	Maldanidae	Lumbriclymene
minor			
## 353	6	Onuphidae	Paradiopatra
hispanica			
## 354	6	Onuphidae	Rhamphobrachium (Spinigerium)
brevibrachiatum			
## 355	8	Goniadidae	Goniada
emerita			
## 356	8	Lumbrineridae	Augeneria
tentaculata			
## 357	8	Polynoidae	Harmothoe
fraserthomsoni			
## 358	9	Hesionidae	Dalhousia
atlantica			
## 359	10	Acoetidae	Panthalis
oerstedii			
## 360	12	Ampharetidae	Anobothrus
gracilis			
## 361	13	Alciopidae	
Alciopina			

```
# Export to CSV
write.csv(endemic_list,
          "Polychaete_Endemic_Species_per_Bioregion.csv",
          row.names = FALSE)

cat("Exported to Polychaete_Endemic_Species_per_Bioregion.csv\n")

## Exported to Polychaete_Endemic_Species_per_Bioregion.csv
```

Rarefaction per Bioregion

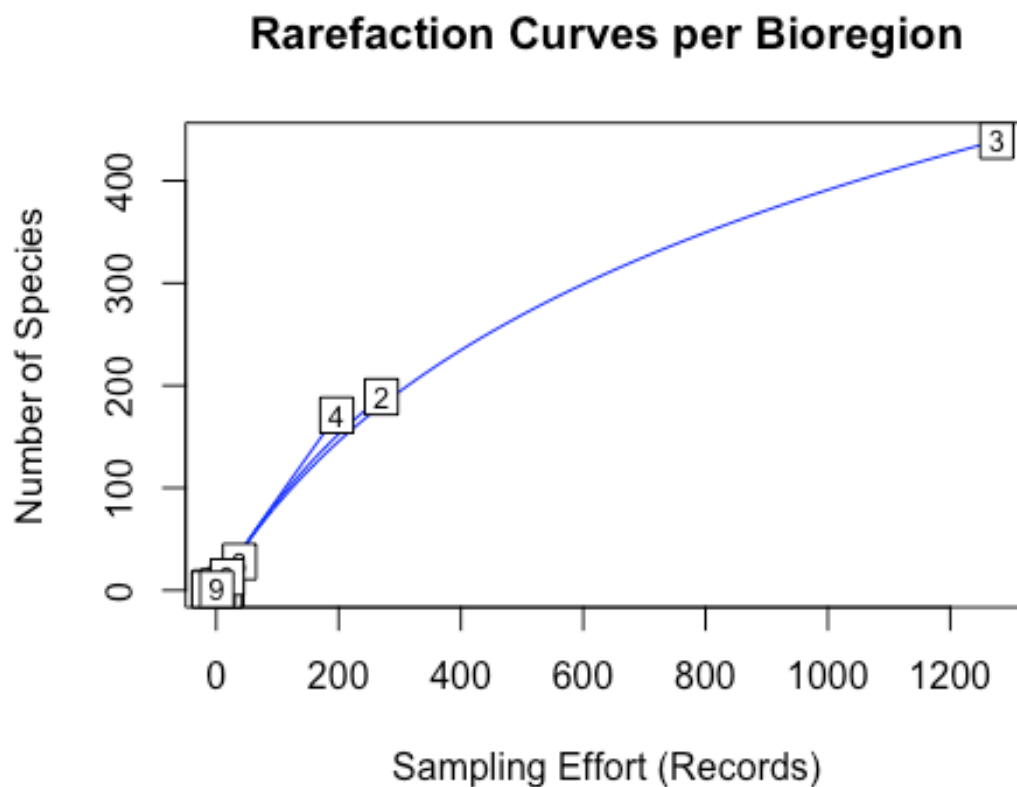
```
site_cluster_map <- final_bioregions$clusters
```

```

# Aggregate comat by bioregion
bioregion_comat <- as.data.frame(comat) %>%
  mutate(ID = rownames(comat)) %>%
  left_join(site_cluster_map, by = "ID") %>%
  group_by(.data[[cluster_col]]) %>%
  summarise(across(-ID, sum)) %>%
  column_to_rownames(cluster_col)

# Combined rarefaction curves
rarecurve(bioregion_comat,
  step = 10,
  col = "blue",
  label = TRUE,
  main = "Rarefaction Curves per Bioregion",
  xlab = "Sampling Effort (Records)",
  ylab = "Number of Species")

```



```

#
Individual panels #n <- nrow(bioregion_comat) #ncols <- ceiling(sqrt(n)) #nrows <- ceiling(n
/ ncols) #par(mfrow = c(nrows, ncols))

#for (i in 1:n) { # rarecurve(bioregion_comat[i, , drop = FALSE], # step = 1, # col = "blue", #
main = paste("Bioregion", rownames(bioregion_comat)[i]), # xlab = "Records", # ylab =
"Species") # grid() #} #par(mfrow = c(1, 1))

```

Species Accumulation Curve (All Sites)

```
sac_total <- specaccum(comat, method = "random", permutations = 1000)

pdf(file = "Collectors Curve.pdf", width=6, height=4)
plot(sac_total,
     ci.type = "poly",
     col     = "darkgreen",
     lwd     = 2,
     ci.lty  = 0,
     ci.col  = "lightgrey",
     main    = "Cumulative Richness of Balearic Polychaetes",
     xlab    = "Number of 1/4 Degree Grid Cells Sampled",
     ylab    = "Cumulative Species Count")

dev.off()

## quartz_off_screen
##                2
```

Effort vs Richness — Exploratory Analysis

— 1. Build the data frame

```
effort_richness <- richness_counts %>%
  left_join(site_effort, by = "Site") %>%
  left_join(cell_number_lookup[, c("Site", "CellNumber")], by = "Site") %>%
  rename(Richness = SpeciesRichness,
         Effort   = NbDataPoints) %>%
  mutate(Is_Leverage = Site == "39.25_3")

cat("Dimensions:", nrow(effort_richness), "cells\n")

## Dimensions: 80 cells

print(effort_richness)

##           Site Richness Effort CellNumber Is_Leverage
## 1  37.75_0.5         1      2         80        FALSE
## 2  38_1.75          2     10         78        FALSE
## 3  38_3.75          1      1         79        FALSE
## 4 38.25_1.75         1      4         76        FALSE
## 5  38.25_4           7     51         77        FALSE
## 6 38.5_0.75          3      6         69        FALSE
## 7  38.5_1           12     16         70        FALSE
## 8 38.5_1.25        121    538         71        FALSE
## 9  38.5_1.5          4      7         72        FALSE
## 10 38.5_1.75         3      3         73        FALSE
## 11 38.5_2.25         2      2         74        FALSE
## 12 38.5_2.5          1      1         75        FALSE
## 13 38.75_1          86    144         61        FALSE
```

## 14	38.75_1.25	65	87	62	FALSE
## 15	38.75_1.5	20	27	63	FALSE
## 16	38.75_1.75	12	17	64	FALSE
## 17	38.75_2	3	6	65	FALSE
## 18	38.75_2.25	1	1	66	FALSE
## 19	38.75_5.75	2	4	67	FALSE
## 20	38.75_6	3	31	68	FALSE
## 21	39_1	14	17	51	FALSE
## 22	39_1.25	86	150	52	FALSE
## 23	39_1.5	59	87	53	FALSE
## 24	39_2.5	6	16	54	FALSE
## 25	39_2.75	11	85	55	FALSE
## 26	39_3	12	31	56	FALSE
## 27	39_4	2	8	57	FALSE
## 28	39_4.75	1	1	58	FALSE
## 29	39_5	1	6	59	FALSE
## 30	39_6	3	22	60	FALSE
## 31	39.25_1.25	1	1	42	FALSE
## 32	39.25_2.25	14	86	43	FALSE
## 33	39.25_2.5	87	891	44	FALSE
## 34	39.25_2.75	4	99	45	FALSE
## 35	39.25_3	98	2172	46	TRUE
## 36	39.25_3.25	22	103	47	FALSE
## 37	39.25_3.5	1	6	48	FALSE
## 38	39.25_5.25	4	36	49	FALSE
## 39	39.25_5.75	3	17	50	FALSE
## 40	39.5_2	12	18	32	FALSE
## 41	39.5_2.25	86	330	33	FALSE
## 42	39.5_2.5	134	432	34	FALSE
## 43	39.5_3	1	1	35	FALSE
## 44	39.5_3.25	17	48	36	FALSE
## 45	39.5_3.5	22	283	37	FALSE
## 46	39.5_4.5	2	2	38	FALSE
## 47	39.5_4.75	2	6	39	FALSE
## 48	39.5_5.25	5	38	40	FALSE
## 49	39.5_5.75	1	1	41	FALSE
## 50	39.75_2.25	3	14	22	FALSE
## 51	39.75_2.5	14	89	23	FALSE
## 52	39.75_2.75	3	15	24	FALSE
## 53	39.75_3	60	153	25	FALSE
## 54	39.75_3.25	12	113	26	FALSE
## 55	39.75_3.5	29	339	27	FALSE
## 56	39.75_3.75	142	351	28	FALSE
## 57	39.75_4	68	167	29	FALSE
## 58	39.75_4.25	55	84	30	FALSE
## 59	39.75_4.75	1	3	31	FALSE
## 60	40_3	5	11	12	FALSE
## 61	40_3.25	1	6	13	FALSE
## 62	40_3.5	10	32	14	FALSE
## 63	40_3.75	87	164	15	FALSE

```
## 64      40_4      87    132      16    FALSE
## 65    40_4.25      6     28      17    FALSE
## 66    40_4.5       1      1      18    FALSE
## 67    40_4.75      2      2      19    FALSE
## 68      40_5     159    477      20    FALSE
## 69      40_6       3     14      21    FALSE
## 70  40.25_2.5      1      1       8    FALSE
## 71  40.25_3.25     4     15       9    FALSE
## 72  40.25_3.75     1      1      10    FALSE
## 73    40.25_4      2      2      11    FALSE
## 74    40.5_2.5     1      2       4    FALSE
## 75    40.5_2.75    1      1       5    FALSE
## 76    40.5_3       2     11       6    FALSE
## 77    40.5_3.75    3      3       7    FALSE
## 78    40.75_5      2      6       3    FALSE
## 79    41_4.75     1      1       1    FALSE
## 80     41_5       1      3       2    FALSE
```

— 2. Boxplots – raw distributions

```
par(mfrow = c(1, 2))
```

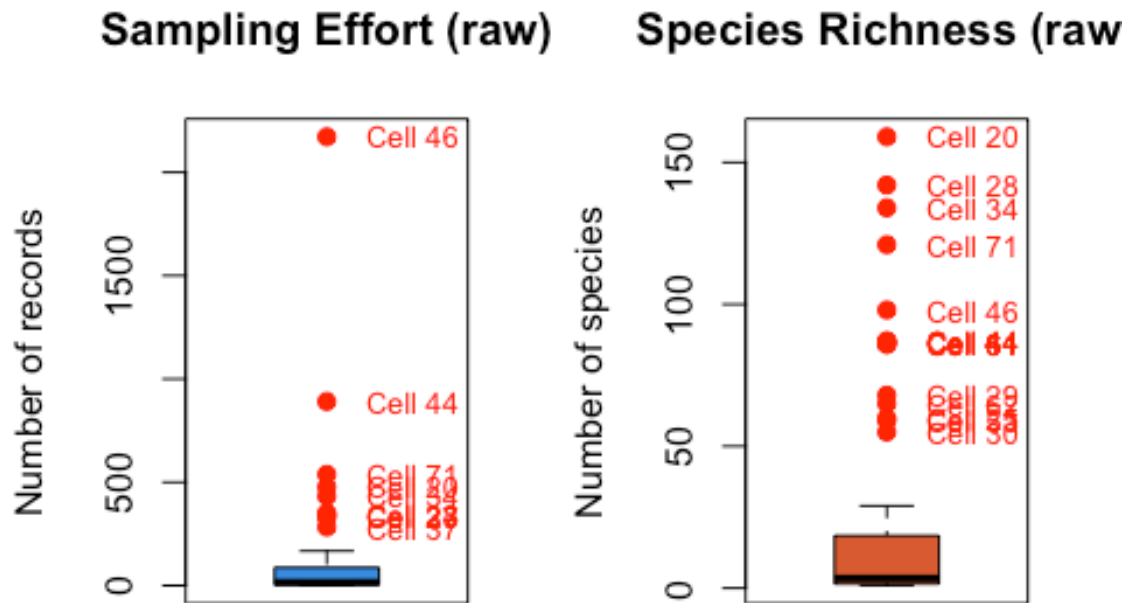
```
boxplot(effort_richness$Effort,
        main = "Sampling Effort (raw)",
        ylab = "Number of records",
        col = "#378ADD",
        outline = TRUE,
        outcol = "red",
        outpch = 19)
```

```
effort_out <- boxplot.stats(effort_richness$Effort)$out
for (val in effort_out) {
  cell_label <- effort_richness$CellNumber[effort_richness$Effort == val]
  text(x = 1.15, y = val, labels = paste("Cell", cell_label),
       cex = 0.8, col = "red", adj = 0)
}
```

```
boxplot(effort_richness$Richness,
        main = "Species Richness (raw)",
        ylab = "Number of species",
        col = "#D85A30",
        outline = TRUE,
        outcol = "red",
        outpch = 19)
```

```
richness_out <- boxplot.stats(effort_richness$Richness)$out
for (val in richness_out) {
  cell_label <- effort_richness$CellNumber[effort_richness$Richness == val]
  text(x = 1.15, y = val, labels = paste("Cell", cell_label),
```

```
cex = 0.8, col = "red", adj = 0)
}
```



```
par(mfrow = c(1, 1))

# — 3. Log-log transformation and model fitting
effort_richness <- effort_richness %>%
  mutate(log_Effort = log10(Effort),
         log_Richness = log10(Richness))

# Linear models
lm_full <- lm(log_Richness ~ log_Effort, data = effort_richness)
lm_noLev <- lm(log_Richness ~ log_Effort,
               data = filter(effort_richness, !Is_Leverage))

cat("\n— LM full dataset —————\n")
##
## — LM full dataset —————
print(summary(lm_full))
```



```

                                method = "pearson")
cor_pearson_noLev <- cor.test(filter(effort_richness,
!Is_Leverage)$log_Effort,
                                filter(effort_richness,
!Is_Leverage)$log_Richness,
                                method = "pearson")

# Spearman on raw values – with 95% CI via DescTools
cor_spearman_full <- SpearmanRho(effort_richness$Effort,
                                effort_richness$Richness,
                                conf.level = 0.95)

cor_spearman_noLev <- SpearmanRho(filter(effort_richness,
!Is_Leverage)$Effort,
                                filter(effort_richness,
!Is_Leverage)$Richness,
                                conf.level = 0.95)

# Shapiro-Wilk on residuals
sw_full <- shapiro.test(residuals(lm_full))
sw_noLev <- shapiro.test(residuals(lm_noLev))

# — 5. Log-log scatter plot


---


pred_grid <- data.frame(
  log_Effort = seq(min(effort_richness$log_Effort),
                    max(effort_richness$log_Effort),
                    length.out = 200)
)

pred_full <- predict(lm_full, newdata = pred_grid, interval = "confidence")
pred_noLev <- predict(lm_noLev, newdata = pred_grid, interval = "confidence")

pred_df <- bind_rows(
  data.frame(pred_grid, pred_full, model = "Full dataset"),
  data.frame(pred_grid, pred_noLev, model = "Without 39.25_3")
)

scatter_plot <- ggplot() +
  geom_ribbon(data = pred_df,
            aes(x = log_Effort, ymin = lwr, ymax = upr, fill = model),
            alpha = 0.15) +
  geom_line(data = pred_df,
            aes(x = log_Effort, y = fit, colour = model),
            linewidth = 0.9) +
  geom_point(data = filter(effort_richness, !Is_Leverage),
             aes(x = log_Effort, y = log_Richness),
             colour = "grey30", size = 2.5, alpha = 0.8) +
  geom_point(data = filter(effort_richness, Is_Leverage),

```

```

    aes(x = log_Effort, y = log_Richness),
    colour = "red", size = 3.5, shape = 17) +
geom_text(data = effort_richness,
    aes(x = log_Effort, y = log_Richness, label = CellNumber),
    size = 2.2, vjust = -0.8, colour = "grey40") +
annotate("text",
    x = min(effort_richness$log_Effort) + 0.05,
    y = max(effort_richness$log_Richness) - 0.05,
    label = paste0(
        "Pearson r (full) = ", round(cor_pearson_full$estimate,
3),
        "\nPearson r (no 39.25_3) = ",
round(cor_pearson_noLev$estimate, 3),
        "\nSpearman ρ (full) = ", round(cor_spearman_full["rho"],
3),
        "\nSpearman ρ (no 39.25_3) = ", round(cor_spearman_noLev["rho"],
3)),
    hjust = 0, vjust = 1, size = 3.2, colour = "grey20") +
scale_colour_manual(values = c("Full dataset" = "#378ADD",
    "Without 39.25_3" = "#D85A30"),
    name = "Model") +
scale_fill_manual(values = c("Full dataset" = "#378ADD",
    "Without 39.25_3" = "#D85A30"),
    name = "Model") +
labs(title = "Sampling effort vs species richness – log-log scale",
    subtitle = "Red triangle = 39.25_3 (high-leverage); shaded bands = 95%
CI",
    x = "log10 (Number of records)",
    y = "log10 (Species richness)") +
theme_minimal(base_size = 13) +
theme(panel.grid.minor = element_blank(),
    plot.subtitle = element_text(colour = "grey50"))

# Export as PDF at publication quality
ggsave("Effort_Richness_Loglog_Scatter.pdf",
    plot = scatter_plot,
    width = 10, # inches
    height = 7, # inches
    dpi = 300,
    device = "pdf")

cat("Exported to Effort_Richness_Loglog_Scatter.pdf\n")

## Exported to Effort_Richness_Loglog_Scatter.pdf

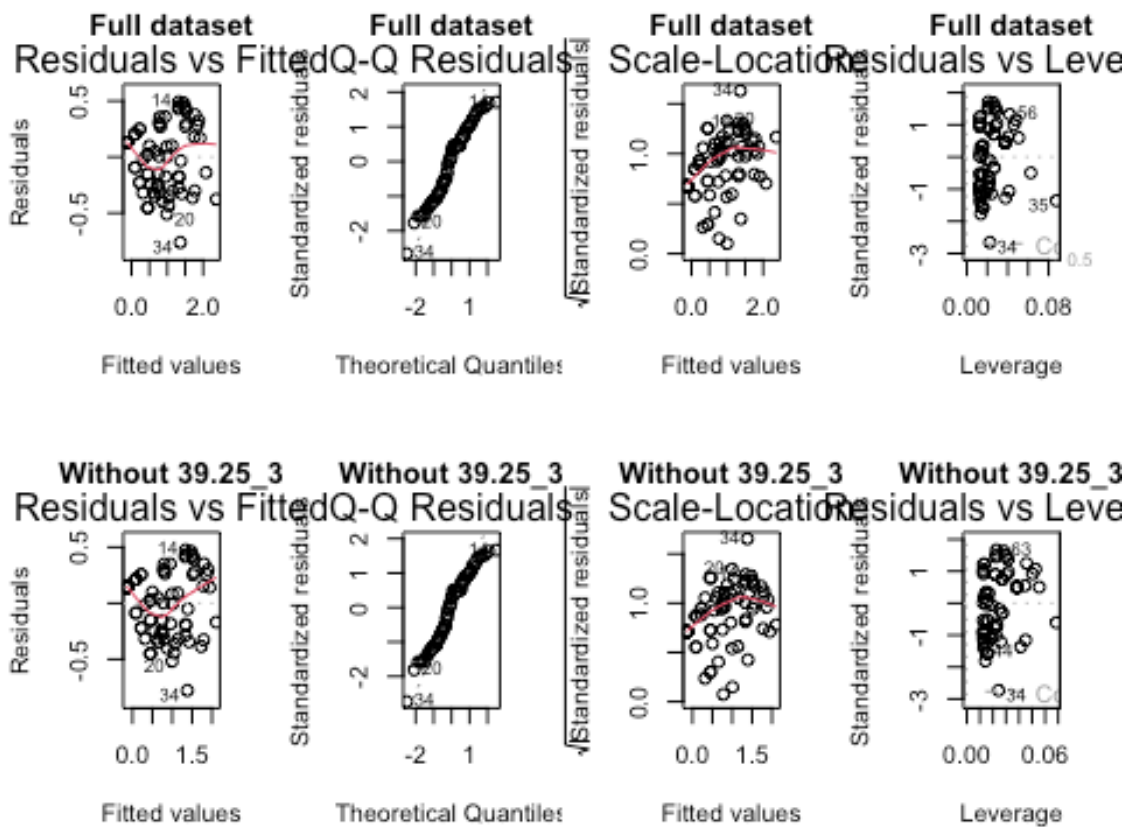
# — 6. Residual diagnostics

```

```

par(mfrow = c(2, 4))
plot(lm_full, main = "Full dataset")
plot(lm_noLev, main = "Without 39.25_3")

```



```
par(mfrow = c(1, 1))
```

```
# — 7. Summary comparison table
```

```
summary_table <- data.frame(
  Method = c("Pearson r (log-log)",
             "Pearson r (log-log)",
             "Spearman rho (raw)",
             "Spearman rho (raw)"),
  Dataset = c("Full (n = 70)",
             "Without 39.25_3 (n = 69)",
             "Full (n = 70)",
             "Without 39.25_3 (n = 69)"),
  Estimate = round(c(cor_pearson_full$estimate,
                    cor_pearson_noLev$estimate,
                    cor_spearman_full["rho"],
                    cor_spearman_noLev["rho"]), 3),
  CI_low = round(c(cor_pearson_full$conf.int[1],
                  cor_pearson_noLev$conf.int[1],
                  cor_spearman_full["lwr.ci"],
                  cor_spearman_noLev["lwr.ci"]), 3),
  CI_high = round(c(cor_pearson_full$conf.int[2],
                   cor_pearson_noLev$conf.int[2],
```

```

        cor_spearman_full["upr.ci"],
        cor_spearman_noLev["upr.ci"]), 3),
p_value = formatC(c(cor_pearson_full$p.value,
                    cor_pearson_noLev$p.value,
                    cor_spearman_full["p"],
                    cor_spearman_noLev["p"]),
                  format = "e", digits = 2),
SW_W     = round(c(sw_full$statistic, sw_noLev$statistic, NA, NA), 4),
SW_p     = round(c(sw_full$p.value,   sw_noLev$p.value,   NA, NA), 4)
)

cat("\n— Correlation summary table —————\n")

##
## — Correlation summary table —————

print(summary_table, row.names = FALSE)

##           Method                Dataset Estimate CI_low CI_high
p_value
## Pearson r (log-log)              Full (n = 70)    0.914  0.869  0.944
2.24e-32
## Pearson r (log-log) Without 39.25_3 (n = 69)    0.913  0.867  0.944
9.41e-32
## Spearman rho (raw)              Full (n = 70)    0.934  0.899  0.957
NA
## Spearman rho (raw) Without 39.25_3 (n = 69)    0.932  0.895  0.956
NA
##      SW_W      SW_p
## 0.9627 0.0196
## 0.9568 0.0093
##      NA      NA
##      NA      NA

write.csv(summary_table,
          "Effort_Richness_Correlation_Summary.csv",
          row.names = FALSE)
cat("Exported to Effort_Richness_Correlation_Summary.csv\n")

## Exported to Effort_Richness_Correlation_Summary.csv

```