5. Exporting to Word and PowerPoint

Editable tables and sharp figures for reports and slides

Contents

Choose the output for its destination . . . 1
Prepare the export examples . . . 1
A reusable download button . . . 2
Export Table 1 for Word . . . 2
Export a high-resolution figure . . . 3
Give each figure the size it needs . . . 3
    Sizes for journals . . . 4
Export vector versions . . . 4
Share the interactive plot as a web page . . . 5
Keep it sharp in Word and PowerPoint . . . 5
Optional: create Word and PowerPoint downloads . . . 6
    An editable Word table and figures . . . 7
    An editable table and figures in PowerPoint . . . 8

This page exports the Moorfields Table 1 from part 3 and the Ghana sepsis figures from part 4. The studies are separate; we combine their outputs only to practise exporting. Allow 20 minutes for the table and two figures.

Choose the output for its destination

Use RTF for an editable Word table and PNG for a picture in Word or PowerPoint. SVG and PDF keep lines and shapes sharp when resized. The optional sections save clickable HTML and create complete Office files.

Prepare the export examples

Run this cell first, even if you just finished part 4. Each page has its own R session. The scripts rebuild the objects without writing export files.

```
# Rebuild the table and plots in this page's fresh R session.
library(gt)
source("data/moorfields-table.R", local = TRUE)
source("data/sepsis-figures.R", local = TRUE)

"The table and figures are ready to export."
```

The setup code is available as moorfields-table.R and sepsis-figures.R. In local R you can reuse the objects from earlier parts instead.

`tbl` and `table_gt` hold Table 1. In this lesson, `figure` is the sepsis PC1 boxplot; the other plot objects come from part 4.

A reusable download button

Run this helper once before exporting. It creates the browser's download links; you do not need to study its internals.

Required browser setup: create download links

```
# Turn a file in this R session into a browser download link.
download_button <- function(path, label = NULL) {
  mime <- switch(
    tools::file_ext(path),
    rtf = "application/rtf",
    html = "text/html",
    png = "image/png",
    svg = "image/svg+xml",
    pdf = "application/pdf",
    docx =
      ↪  "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
    pptx =
      ↪  "application/vnd.openxmlformats-officedocument.presentationml.presentation",
    "application/octet-stream"
  )
  htmltools::tags$a(
    href = paste0("data:", mime, ";base64,", base64enc::base64encode(path)),
    download = basename(path),
    class = "btn btn-primary",
    if (is.null(label)) paste("Download", basename(path)) else label
  )
}

"Download helper ready."
```

Files written in browser R disappear when you reload the page. Download them before leaving. In local R, the files save to your working directory.

Export Table 1 for Word

```
# RTF keeps the table text and cells editable in Word.
writeLines(gt::as_rtf(table_gt), "moorfields_table1.rtf")
download_button("moorfields_table1.rtf", "Download Table 1 for Word")
```

Open the RTF in Word, then save as `.docx`. The cells and text remain editable. Check column widths, title, and footnotes after opening: fonts and spacing can change between formats.

Optional: save the styled web table

```
writeLines(gt::as_raw_html(table_gt, inline_css = FALSE), "moorfields_table1.html")
download_button("moorfields_table1.html", "Download styled HTML Table 1")
```

HTML keeps the `gt` styling for a browser report or a display reference.

Export a high-resolution figure

Choose the physical size, then the resolution. This saves the sepsis boxplot at seven by five inches and 300 dots per inch (dpi).

```
# Save the named plot at its intended size, then offer the file for download.
ggplot2::ggsave(
  "sepsis_pc1_boxplot.png", plot = figure, device = ragg::agg_png,
  width = 7, height = 5, units = "in", dpi = 300, bg = "white"
)

download_button("sepsis_pc1_boxplot.png", "Download 300-dpi boxplot")
```

The image is 2100 × 1500 pixels. At seven inches wide it has 300 pixels per inch; at fourteen inches it has 150. Changing a resolution label cannot add pixels. `plot = figure` names the plot explicitly; without it, `ggsave()` uses the last plot displayed.

Try `dpi = 150`, download again, and compare the two files at the same size. The lower-resolution image has 1050 × 750 pixels.

Insert the file with **Insert → Pictures** in Word or PowerPoint. Keep its aspect ratio and check the labels at the final page or slide size. A screenshot may contain far fewer pixels than the exported file.

Give each figure the size it needs

A plot's panels, labels, and legends determine the space it needs. Give heatmaps more height for their labelled rows and multi-panel figures more width. Export the heatmap and patient paths at different shapes:

```
# Add a source credit to a copy; keep pattern_heatmap unchanged.
heatmap_export <- pattern_heatmap + labs(caption = source_note)

ggplot2::ggsave(
  "sepsis_heatmap.png", plot = heatmap_export, device = ragg::agg_png,
  width = 7, height = 7, units = "in", dpi = 300, bg = "white"
)

ggplot2::ggsave(
  "sepsis_paths.png", plot = paths, device = ragg::agg_png,
  width = 8, height = 4, units = "in", dpi = 300, bg = "white"
)

htmltools::tagList(
  download_button("sepsis_heatmap.png", "Download heatmap"),
  download_button("sepsis_paths.png", "Download patient paths")
)
```

Keep the source credit when sharing these CC BY 4.0 data. The exported heatmap receives one without changing `pattern_heatmap`; the boxplot already has a credit.

Open each image at the size you intend to use. If labels overlap, increase width or height. If you halve a figure's width, its text keeps the same point size and may no longer fit.

Sizes for journals

> **i** Optional: use a journal's dimensions
>
> Nature's guide gives 89 mm for one column and 183 mm for two. Check the target journal's guide and export near the printed size. Set `units = "mm"` to use its measurements directly.

```
# Use smaller text for a figure drawn at journal column width.
journal_trajectory <- trajectory + theme_classic(base_size = 8)

ggplot2::ggsave(
  "sepsis_trajectory_183mm.png", plot = journal_trajectory,
  device = ragg::agg_png, width = 183, height = 120, units = "mm",
  dpi = 300, bg = "white"
)

download_button("sepsis_trajectory_183mm.png", "Download journal-width trajectory")
```

> This replaces the trajectory's theme with `base_size = 8`. It is safe here because `trajectory` has no additional `theme()` settings. Doing the same to `figure` would also reset its legend position and bold title; add those settings again if needed.
> Try `width = 89, height = 70`. The two legends crowd that single-column version, so it needs fewer legend entries or a legend below the plot.

Export vector versions

SVG stores the chart as shapes, so it remains sharp when resized. Current Microsoft 365 Word and PowerPoint can insert SVG; check the Microsoft guidance for your platform.

```
# SVG stores shapes rather than a fixed grid of pixels.
ggplot2::ggsave(
  "sepsis_pc1_boxplot.svg", plot = figure, device = svglite::svglite,
  width = 7, height = 5, units = "in", bg = "white"
)

download_button("sepsis_pc1_boxplot.svg", "Download vector SVG boxplot")
```

Vector output does not need a DPI setting. An inserted SVG is a graphic, not an Office chart with an embedded spreadsheet.

> **i** Optional: save the six-panel figure as PDF
>
> ```
> ggplot2::ggsave("sepsis_six_heatmaps.pdf", plot = six_panels,
>                 width = 12, height = 7, units = "in")
>
> download_button("sepsis_six_heatmaps.pdf", "Download six-panel PDF")
> ```
>
> These heatmaps contain about 1,500 tiles, which suit vector output. Tens of thousands of points can make a large, slow vector file; PNG is often easier for that case.

Export troubleshooting

If the wrong plot saves, supply `plot =`. If text is cut off, enlarge the figure or reduce `base_size`. Heatmaps need height for their rows and width for their panels. Export a blurry image again at its final size and 300 dpi. If a browser file disappears after a reload, rerun its export cell.
`ggsave()` accepts ggplots. The interactive widget needs the next method.

Share the interactive plot as a web page

Optional: save clickable patient paths in local R

An HTML widget keeps hover and click controls. Run this in local R after the two setup scripts, with an `output/` folder in your project. It needs Pandoc, which browser R does not have.

```
library(plotly)

patient_key <- highlight_key(patients, ~patient_id)

clickable <- ggplot(patient_key, aes(
    x = P23, y = P4, group = patient_id, colour = outcome,
    text = paste0("Patient ", patient_id,
                  "<br>Visit: ", visit,
                  "<br>Pattern 23: ", round(P23, 2),
                  "<br>Pattern 4: ", round(P4, 2))
  )) +
  geom_path(alpha = 0.3) +
  geom_point(size = 1) +
  facet_wrap(~ outcome) +
  scale_colour_manual(values = line_colours) +
  labs(x = "Pattern 23 value", y = "Pattern 4 value", colour = "28-day outcome") +
  theme_classic(base_size = 12)

interactive_paths <- ggplotly(clickable, tooltip = "text") |>
  plotly::highlight(on = "plotly_click", color = "black")

# Pack the widget and its supporting files into one HTML page.
htmlwidgets::saveWidget(
  interactive_paths, "output/sepsis_paths_interactive.html",
  selfcontained = TRUE, title = "Ghana sepsis cohort: patient paths"
)
```

`plotly::highlight()` becomes part of `interactive_paths`, so click highlighting stays in the saved file. The prefix distinguishes it from flextable's different `highlight()` function.
`selfcontained = TRUE` packs the widget into one file, usually several megabytes because it includes plotly. Install rmarkdown if R asks for it. If Pandoc is missing, install Pandoc or use `selfcontained = FALSE`. The latter creates the HTML plus `sepsis_paths_interactive_files`; share both.
Word and PowerPoint cannot play the widget. Insert a static `paths` picture and link to the HTML for readers who want to explore it without R.

Keep it sharp in Word and PowerPoint

Insert the saved file, keep its aspect ratio, and inspect it at its final size. Where available, choose **High fidelity** and avoid downsampling the high-resolution image. Check the saved document or its PDF too.

Microsoft's guides cover image resolution and compression. An editable table has no pixel-resolution requirement.

Optional: create Word and PowerPoint downloads

The following cells run in the browser. Load the packages, run the Word example, then run PowerPoint: the last step reuses the styled table and heatmap file.

Load the Office packages

```
webr::install(c("flextable", "officer", "rvg"))
library(flextable)
library(officer)
library(rvg)

"Office export packages are ready."
```

For local R, download seminar-exports.R into the lesson folder. Put the prepared CSVs and both setup scripts in `data/`, install the packages listed in the script, then run `source("seminar-exports.R")`. It writes every export on this page, including interactive HTML, to `output/`.

An editable Word table and figures

Build the Word report

`as_flex_table()` reuses the calculations in `tbl`. Style it separately: `gt` styling does not transfer. Explicit widths and `include = -autofit` avoid automatic sizing in browser R.

```
# Style the editable table separately from the gt display table.
ft <- as_flex_table(tbl, include = -autofit) |>
  theme_booktabs() |>
  bg(bg = "#003B49", part = "header") |>
  color(color = "white", part = "header") |>
  fontsize(size = 9, part = "all") |>
  width(j = 1, width = 2.7) |>
  width(j = 2:6, width = 0.7)

# Save the report's figures, even if you skipped the exports above.
ggplot2::ggsave(
  "sepsis_pc1_boxplot.png", plot = figure, device = ragg::agg_png,
  width = 7, height = 5, units = "in", dpi = 300, bg = "white"
)
ggplot2::ggsave(
  "sepsis_heatmap.png", plot = pattern_heatmap + labs(caption = source_note),
  device = ragg::agg_png, width = 7, height = 7, units = "in",
  dpi = 300, bg = "white"
)


# Add the table and figures in the order they should appear.
report <- read_docx() |>
  docx_set_paragraph_style(
    style_id = "Title", style_name = "Title",
    fp_t = fp_text(font.family = "Arial", font.size = 22,
                   bold = TRUE, color = "black")
  ) |>
  body_add_par("Seminar report: Table 1 and sepsis figures", style = "Title") |>
  body_add_flextable(ft) |>
  body_add_break() |>
  body_add_img("sepsis_pc1_boxplot.png", width = 6.2, height = 6.2 * 5 / 7) |>
  body_add_par("Figure 1. PC1 scores by visit and 28-day outcome.",
               style = "Normal") |>
  body_add_break() |>
  body_add_img("sepsis_heatmap.png", width = 6.2, height = 6.2) |>
  body_add_par("Figure 2. Pattern z-scores by group and visit.",
               style = "Normal")

print(report, target = "seminar_report.docx")
download_button("seminar_report.docx", "Download your Word report")
```

The table widths total 6.2 inches. The boxplot uses `6.2 * 5 / 7` for its height to keep its 7×5 shape; the heatmap is square. `body_add_break()` starts each figure on a new page.
Try changing a title or header colour, rerun, then download the report.

An editable table and figures in PowerPoint

Build the three-slide presentation

```
# Give the table a larger font for slides.
ft_slide <- ft |>
  fontsize(size = 14, part = "all") |>
  width(j = 1, width = 3.0) |>
  width(j = 2:6, width = 1.1)

slides <- read_pptx() |>
  add_slide(layout = "Title and Content", master = "Office Theme") |>
  ph_with("Table 1. Baseline characteristics", ph_location_type("title")) |>
  ph_with(ft_slide, ph_location(left = 0.5, top = 1.5)) |>
  add_slide(layout = "Title and Content", master = "Office Theme") |>
  ph_with("Survivors move towards healthy donors", ph_location_type("title")) |>
  ph_with(dml(ggobj = trajectory), ph_location_type("body")) |>
  add_slide(layout = "Title Only", master = "Office Theme") |>
  ph_with("Pattern z-scores by group and visit", ph_location_type("title")) |>
  ph_with(external_img("sepsis_heatmap.png", width = 5.5, height = 5.5),
          ph_location(left = 2.25, top = 1.6, width = 5.5, height = 5.5))

print(slides, target = "seminar_slides.pptx")
download_button("seminar_slides.pptx", "Download your PowerPoint slides")
```

The slides contain an editable table, an editable trajectory drawing, and a heatmap picture. Table widths total 8.5 inches in the default ten-inch-wide template. Its 14-point font is larger than the Word table's. If `ft` or `sepsis_heatmap.png` is missing, run the Word cell first.

The trajectory drawing is editable shape by shape, not an Office chart linked to data. The heatmap stays a picture. For a department template, use `read_pptx("template.pptx")` with its layouts and placeholders.

← 4. Basic figures with ggplot2 6. Now you try: a clinical data project →