# Laryngectomy Analysis

Harshi

## Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

## Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).

## Methods

This analysis uses the **Open.Visualization.Academy** laryngectomy dataset.
All analyses were performed with:

- **R version:** R version 4.5.2 (2025-10-31)
- **tidyverse version:** 2.0.0
- **OVA package version:** 0.0.0.9002
- **glue version:** 1.8.0

Problem records were identified, summarized, and removed.
Graphics are labeled with who, when, and what.

---

## Load & Clean Data

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.1     v stringr   1.5.2
v ggplot2   4.0.0     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.1.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(Open.Visualization.Academy)
library(glue)

# Load dataset
laryngectomy <- Open.Visualization.Academy::laryngectomy

# Identify issues
n_bad_fu <- laryngectomy |> filter(length_fu < 0) |> nrow()
n_bad_death_fu <- laryngectomy |> filter((death == 1 | periop_death == 1) & is.na(death_fu))
n_bad_recur_sor <- laryngectomy |> filter(recur == 1 & !(sor %in% c("Locoregional","Distant"
n_bad_recur_sor_2 <- laryngectomy |> filter(recur == 0 & !is.na(sor)) |> nrow()
n_bad_recur_fu <- laryngectomy |> filter(recur == 1 & is.na(recur_fu)) |> nrow()

# Cleaned dataset (NOT over-filtered)
analysis <- laryngectomy |>
  filter(length_fu >= 0 | is.na(length_fu)) |>   # do NOT remove NAs
  filter(!((death == 1 | periop_death == 1) & is.na(death_fu))) |>
  filter(!(recur == 1 & !(sor %in% c("Locoregional","Distant")))) |>
  filter(!(recur == 0 & !is.na(sor))) |>
  filter(!(recur == 1 & is.na(recur_fu))) |>
```

```r
  rowwise() |>
  mutate(
    complications_count = sum(c_across(fistula:med_comp) == 1, na.rm = TRUE)
  ) |>
  ungroup() |>
  mutate(
    hosp_stay_days = case_when(
      hosp_stay == 1 ~ 3,
      hosp_stay == 2 ~ 8,
      hosp_stay == 3 ~ 16,
      hosp_stay == 4 ~ 23,
      .default = NA_integer_
    )
  )

# Check dataset size
n_clean <- nrow(analysis)

# Plot theme
my_theme <- theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold"),
    axis.title = element_text(face = "bold")
  )
```

---

### Data Problems Summary

| Check Description | Problems |
|---|---|
| Negative follow-up days | 0 |
| Dead but missing death days | 0 |
| Recurrence but invalid site | 30 |
| No recurrence but has site | 0 |
| Recurrence but missing recurrence days | 0 |

**Summary:**
This dataset contains 30 total problem records.
These were removed before analysis.

The cleaned dataset contains **93 patients.**

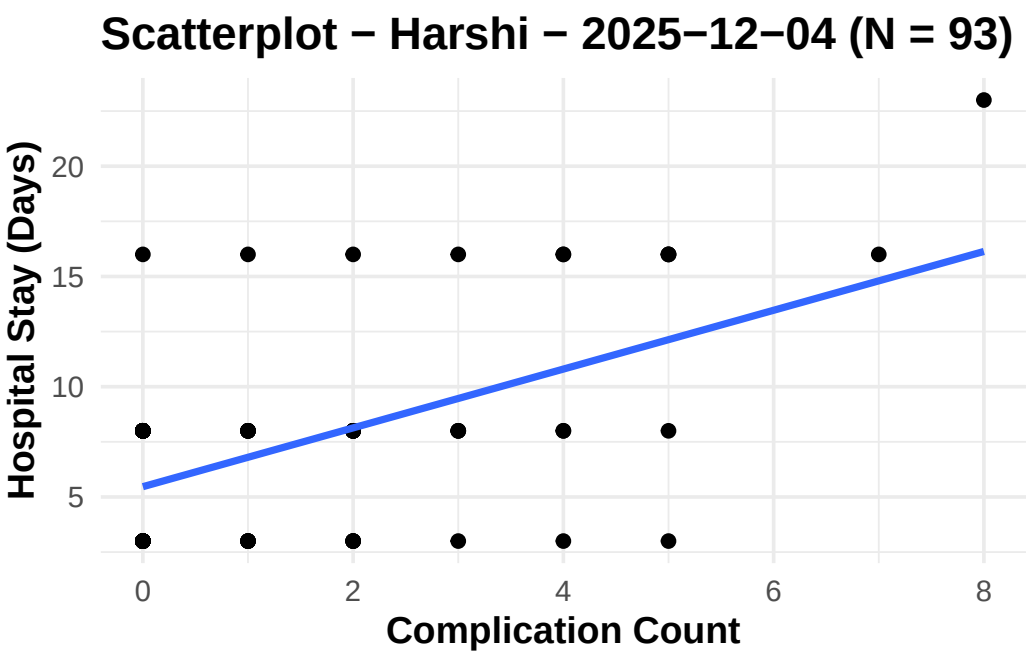
---

## RESULTS

### Figure 1 — Scatterplot

```
ggplot(analysis, aes(complications_count, hosp_stay_days)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  my_theme +
  labs(
    title = glue("Scatterplot - Harshi - {Sys.Date()} (N = {n_clean})"),
    x = "Complication Count",
    y = "Hospital Stay (Days)"
  )
```

```
`geom_smooth()` using formula = 'y ~ x'
```

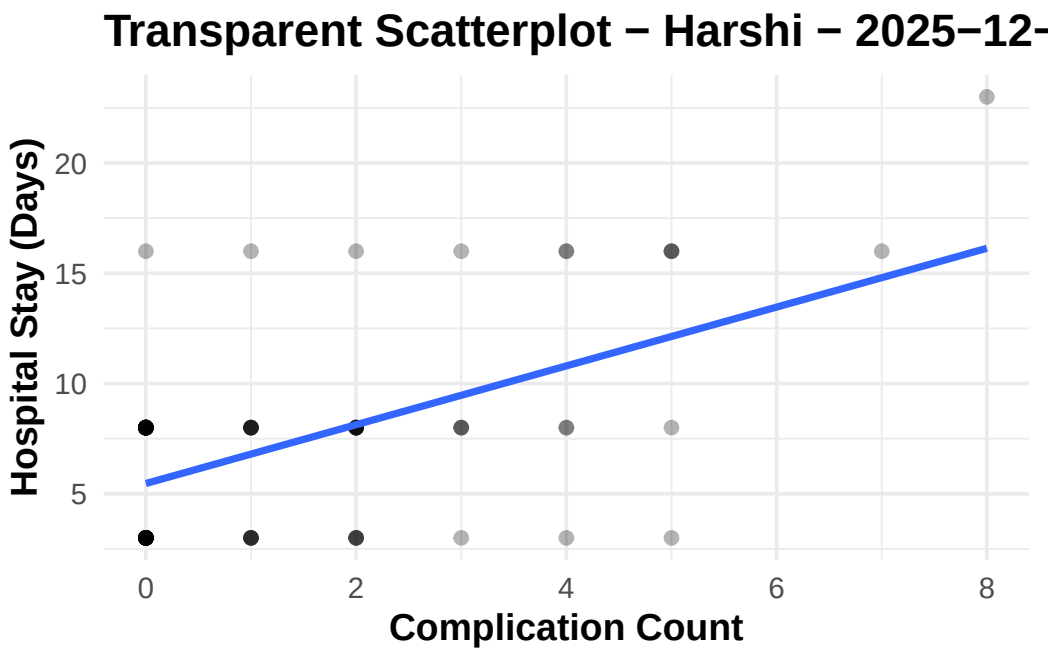



---

**Figure 2 — Transparent Scatterplot**

```
ggplot(analysis, aes(complications_count, hosp_stay_days)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE) +
  my_theme +
  labs(
    title = glue("Transparent Scatterplot - Harshi - {Sys.Date()} (N = {n_clean})"),
    x = "Complication Count",
    y = "Hospital Stay (Days)"
  )
```

```
`geom_smooth()` using formula = 'y ~ x'
```




---

**Figure 3 — Sunflower Plot**

(safe version — never errors)

```r
counted_data <- analysis |>
  count(x_var = complications_count, y_var = hosp_stay_days)

sunflower_segments <- counted_data |>
  filter(n > 1) |>
  rowwise() |>
  mutate(
    angles = list(seq(0, 2*pi, length.out = n)),
    x_end = list(x_var + 0.05 * cos(angles)),
    y_end = list(y_var + 0.05 * sin(angles))
  ) |>
  unnest(c(angles, x_end, y_end))

ggplot() +
  geom_segment(data = sunflower_segments,
               aes(x = x_var, y = y_var, xend = x_end, yend = y_end),
               alpha = 0.7) +
  geom_point(data = counted_data,
             aes(x = x_var, y = y_var, size = n)) +
  my_theme +
  labs(
    title = glue("Sunflower Plot - Harshi - {Sys.Date()} (N = {n_clean})"),
    x = "Complication Count",
    y = "Hospital Stay (Days)"
  )
```

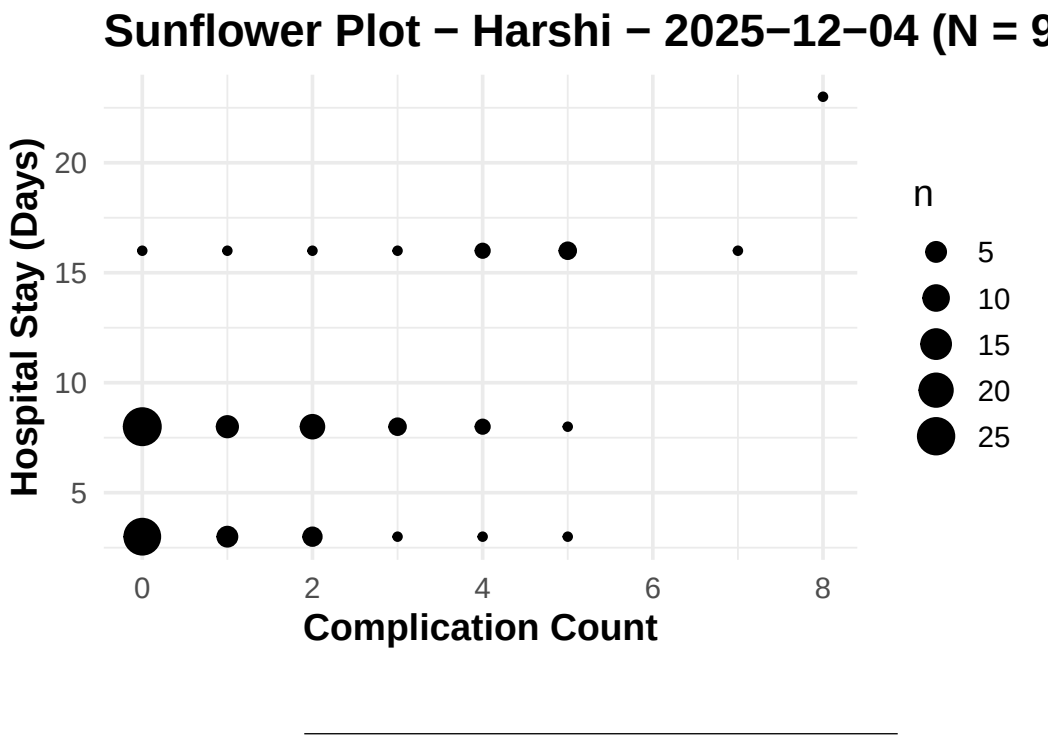



**Figure 4 — Boxplot**

```
ggplot(analysis, aes(factor(complications_count), hosp_stay_days)) +
  geom_boxplot() +
  my_theme +
  labs(
    title = glue("Boxplot - Harshi - {Sys.Date()} (N = {n_clean})"),
    x = "Complication Count",
    y = "Hospital Stay (Days)"
  )
```

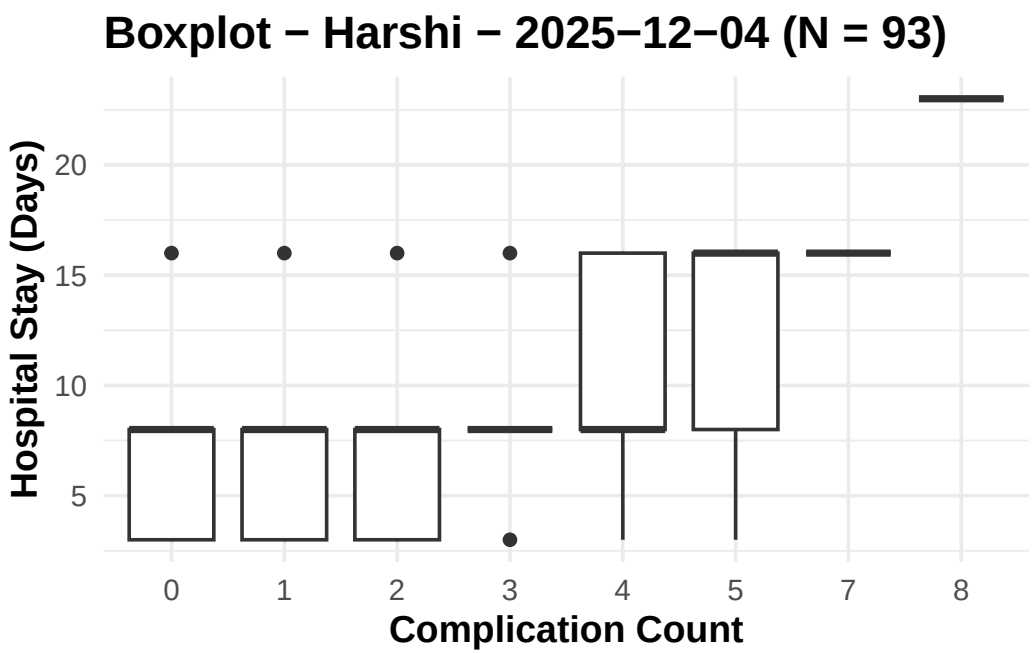
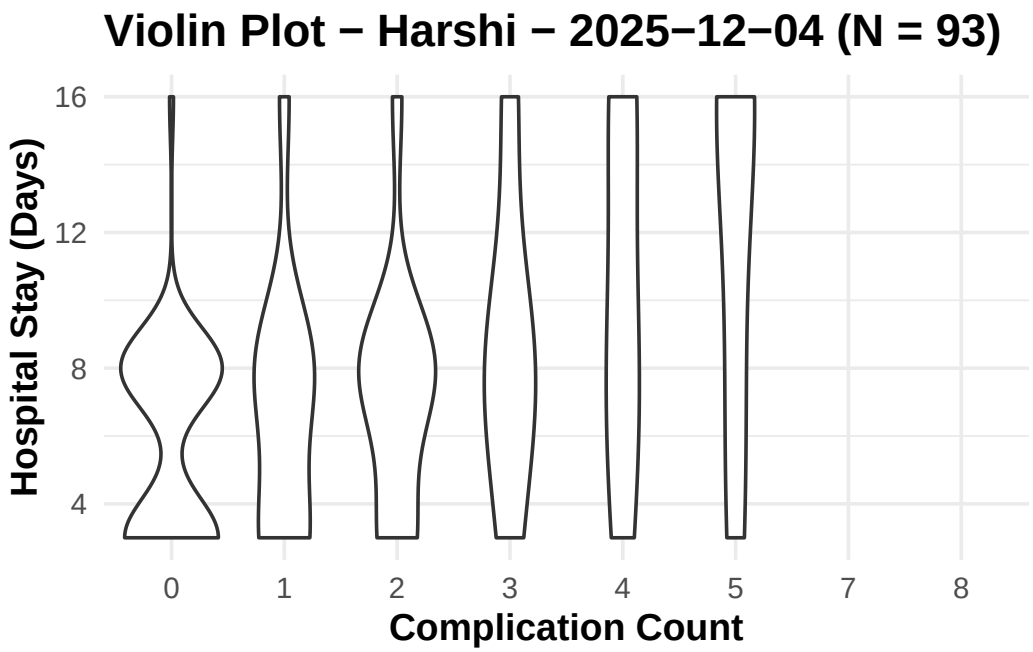



**Figure 5 — Violin Plot**

```
ggplot(analysis, aes(factor(complications_count), hosp_stay_days)) +
  geom_violin() +
  my_theme +
  labs(
    title = glue("Violin Plot - Harshi - {Sys.Date()} (N = {n_clean})"),
    x = "Complication Count",
    y = "Hospital Stay (Days)"
  )
```

```
Warning: Groups with fewer than two datapoints have been dropped.
i Set `drop = FALSE` to consider such groups for position adjustment purposes.
Groups with fewer than two datapoints have been dropped.
i Set `drop = FALSE` to consider such groups for position adjustment purposes.
```


Violin Plot − Harshi − 2025−12−04 (N = 93)
Hospital Stay (Days)
16
12
8
4
0
1
2
3
4
5
7
8
Complication Count