

# Statistical Analysis of Clinical Variables in Necrotizing Fasciitis Patients Using NPC and Bootstrap Methods

Author: SAI TEJA BANDARU



# Table of Contents

---

Introduction and Clinical Background

---

Importance of Variables

---

Research Objectives

---

Data Source and Preprocessing

---

Statistical Methodology Overview

---

NPC Tests – Theory and Application (Pesarin 2001, Bonnini 2014)

---

Combination Methods: Fisher, Tippett, Liptak

---

NPC Ranking Methods – Theory and Implementation

---

Bootstrap Analysis of Key Variables

---

Visualizations and Interpretation

---

Discussion and Clinical Relevance

---

References and Appendix

# Introduction and Background

- Necrotizing fasciitis (NF) is a life-threatening infection with rapid tissue necrosis. Early diagnosis and evaluation of risk factors are critical. This project analyzes data using Non-Parametric Combination (NPC) of tests, bootstrap estimation, and ranking methods to assess the relationship between clinical biomarkers and outcomes like amputation and mortality.



# Importance of Studied Variables



**HBA1C:** Reflects long-term blood glucose levels and is linked to poor wound healing.



**Albumin:** Indicates nutritional and inflammatory status; low values suggest systemic risk.



**Amputazione:** Binary indicator (1/0) of major surgical intervention.



**Morto:** Binary outcome of survival status.

# Research Objectives



Prepare the clinical dataset for statistical analysis.



Perform Non-Parametric Combined (NPC) tests to evaluate clinical variables.



Use Fisher's, Tippett's, and Liptak's methods to combine partial p-values.



Implement NPC Ranking to identify the most severe patient profiles.



Estimate confidence intervals with bootstrapping.



Visualize global rankings and interpret correlations.

# NPC Tests – Theory and Application

- The Non-Parametric Combination (NPC) test is a multivariate, permutation-based method designed to test multiple hypotheses simultaneously.
- **Developed by:** Prof. Fortunato Pesarin (University of Padua, Italy)
- **Goal:** Combine several dependent and/or independent tests without relying on parametric assumptions.
- **Application in Clinical Research:** Handles heterogeneous variables (continuous, ordinal, binary), is robust against outliers, and supports small sample sizes.

# Historical Origins of NPC Methodology

- **Fortunato Pesarin**, University of Padua, is the **founding father** of the modern NPC (Non-Parametric Combination) methodology.
- His 2001 book “*Multivariate Permutation Tests*” laid the mathematical foundation for partial tests, combination functions, and global test statistics.
- **Pesarin's framework** enabled robust, assumption-free statistical inference in complex multivariate settings.
- **Giovanni Bonnini**, later expanded NPC theory, especially in the context of **ranking and multiple comparisons**, collaborating in several updates and applied studies.

# Foundational Scientists in NPC & Multiple Testing

- **Fisher (1932)**: Introduced **Fisher's Method** for combining independent p-values into a global test using the chi-squared distribution.
- **Bonferroni (1936)**: Developed the **Bonferroni Correction**, adjusting significance thresholds in multiple comparisons to control Type I error.
- **Tippett (1931)**: One of the first to propose using the **minimum partial p-value** as a global test statistic.
- **Lipták (1958)**: Proposed a **weighted sum** of inverse normal scores from p-values – now known as the **Lipták combination method**.
- **Fortunato Pesarin (2001)**: Formalized the **Nonparametric Combination of dependent tests**, allowing inference with minimal assumptions.
- Institution: **University of Padua, Italy**
- Contribution: Unified framework for multiple testing, ranking, partial p-values, and permutation-based inference.

# Principles and Strengths of NPC

- No assumption of normality or homogeneity of variance
- Fully distribution-free inference
- Valid under exchangeability of data
- Suitable for mixed-type data (numeric, binary, ordinal)
- Permutation-based p-values allow exact inference for small samples
- Can be applied to unbalanced designs and missing data
- Supports multiple testing with simultaneous control of Type I error

# NPC Tests – Theoretical Foundation

---

**Non-Parametric Combination (NPC) Tests** are an advanced methodology proposed by **Pesarin (2001)** and extended by **Bonnini et al. (2014)** to deal with:

---

Multiple dependent variables.

---

Non-normal data.

---

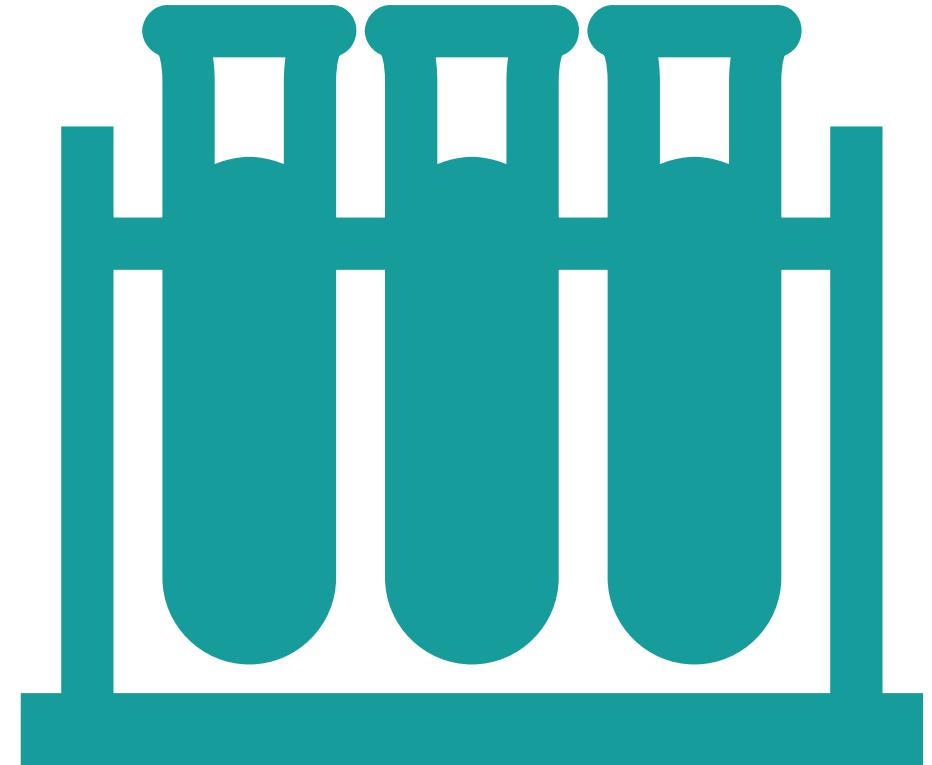
Small sample sizes.

---

The main idea is to perform **partial tests** on each variable separately and then **combine** their p-values using a non-parametric method.

# Components and Structure of NPC Tests

- **Partial Tests:**
  - Each variable is tested individually
  - P-values derived from classical or non-parametric tests
- **Combination Function:**
  - Combines partial p-values into one global statistic
  - Examples: Fisher, Tippett, Liptak, Bonferroni
- **Global Test:**
  - A combined result indicating overall effect
  - Interpreted using permutation distributions



# Descriptive Analysis of Clinical Variables

- **Variables Selected:**
  - HBA1C – Glycemic control biomarker
  - ALBUMINA – Nutritional marker
  - MORTO – Mortality status (binary)
  - AMPUTAZIONE – Presence of major amputation (binary)
- **Why these variables?**
  - Clinically relevant for assessing severity and prognosis in Necrotizing Fasciitis
  - Mix of continuous and binary data fits the NPC framework

# NPC Tests – Why Use Them?

1

Traditional multivariate methods (e.g., MANOVA) assume normality, independence, and large samples.

2

NPC methods allow testing **null hypotheses across multiple variables** without relying on distributional assumptions.

3

Particularly suitable for **clinical data**, where sample sizes may be small and data distributions skewed.

# Multiple Testing in Medical Research

- Medical data often involve:
- Numerous variables
- High correlation between features
- Small sample sizes
- Traditional methods fail to manage the inflated false discovery rate (FDR).
- **Solution:** NPC provides a unified framework to test all hypotheses simultaneously.

# Multiple Comparisons Problem and NPC Advantage

- When testing 10 hypotheses at  $\alpha=0.05$  , there's a ~40% chance of false positive
- NPC controls the error using exact permutation-based resampling
- Bonferroni and similar methods can be too conservative – NPC balances power and control

# Fisher, Tippett, Liptak – Combining Partial P-values

Fisher:  $-2 \sum \log(p_i) \sim \chi^2$

Tippett:  $p_{min}$  — sensitive to smallest effect

Liptak:  $\sum w_i \Phi^{-1}(1 - p_i)$  — weighted Z-transform

NPC uses these to synthesize a powerful global test statistic

# Bonferroni Correction – Theory and Use in NPC

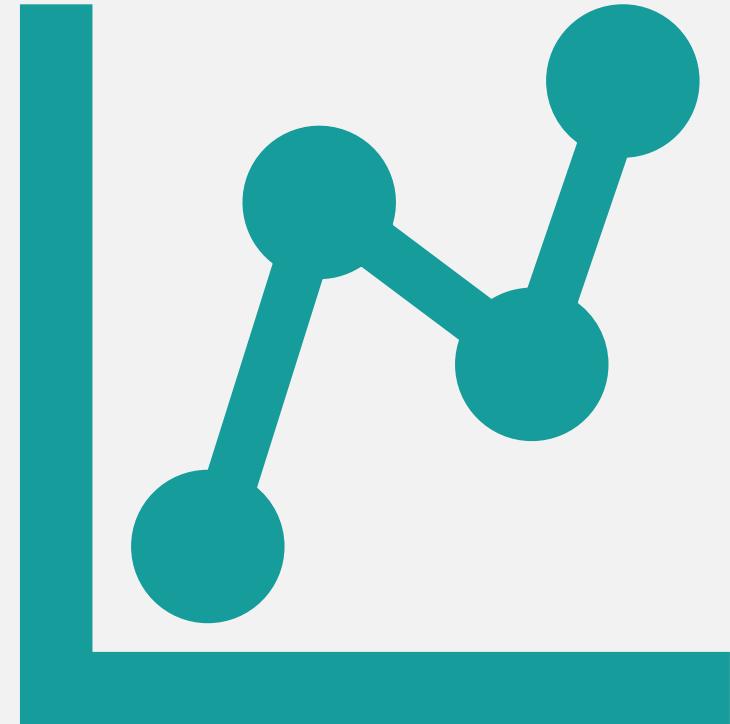
- Controls Family-Wise Error Rate (FWER):

$$\alpha_{adjusted} = \frac{\alpha}{m}$$

- Conservative but simple
- Used in NPC as a complementary technique
- Ensures error control under multiple partial comparisons

# NPC Ranking Methods – Concept and Development

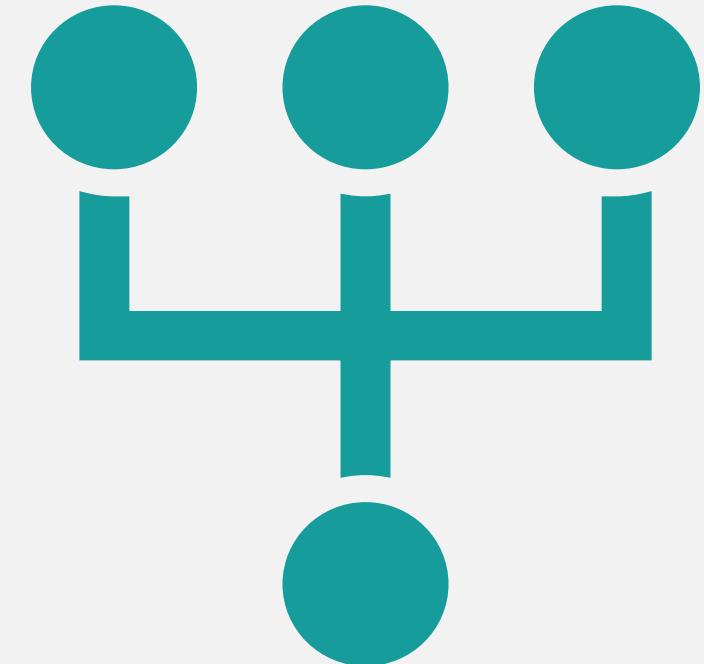
- Ranks individual variables using their values (low values = poor outcome)
- Aggregates rankings to determine overall patient severity
- Robust to scale changes
- Visual summary of patient risk profile



# Constructing Partial and Global Rankings

## Steps:

- Rank each variable independently (HBA1C, ALBUMINA, MORTO, AMPUTAZIONE)
- Sum all ranks to obtain global ranking
- Sort patients by global severity



**Note:** Lower global ranks = higher clinical severity

# NPC Global Ranking Table - Top 10 Most Severe Cases

S.NO	AMPUTAZIONE	HBA1C	MORTO	ALBUMINA	RANK_AMPUTAZIONE	RANK_HBA1C	RANK_MORTO	RANK_ALBUMINA	RANK_GLOBALE
1	0	0.048	0	1.9	21.5	1	23.5	9	55
2	0	0.058	0	1.84	21.5	7	23.5	5	57
3	0	0.058	0	1.87	21.5	7	23.5	7	59
4	0	0.057	0	2.4	21.5	5	23.5	21	71
5	0	0.078	0	1.9	21.5	23.5	23.5	9	77.5
6	0	0.059	0	2.5	21.5	9	23.5	27	81
7	0	0.06	0	2.5	21.5	11	23.5	27	83
8	0	0.072	0	2.3	21.5	22	23.5	16.5	83.5
9	0	0.054	0	2.74	21.5	2	23.5	38	85
10	0	0.055	0	2.8	21.5	3	23.5	40	88

# Partial Tests in NPC

- Each variable of interest is tested independently using a suitable test (e.g., t-test, rank test). The resulting **p-values** are called **partial p-values**.
- For example:

```
p_values <- c(  
  HBA1C = t.test(data$HBA1C, mu = 7)$p.value,  
  ALBUMINA = t.test(data$ALBUMINA, mu = 2.8)$p.value  
)
```

- These p-values are then aggregated into a global test.

# Combining Partial p- values



To derive an overall conclusion, NPC uses **combination functions**. The three most common methods are:



**Fisher's method** (sum of logs of p-values)



**Tippett's method** (minimum p-value)



**Liptak's method** (weighted z-scores) Each has different sensitivity characteristics and is chosen based on context.

# Partial P-values of Clinical Variables

- Content:
  - Show the t-test results:
    - HBA1C: p-value = **1.043841e-154**
    - ALBUMINA: p-value = **0.2672962**
  - Brief interpretation:
    - Extremely low p-value for HBA1C suggests **significant deviation** from clinical reference.
    - ALBUMIN does **not show a significant deviation** from the expected value.

# Fisher's Combination Method

- Developed by Ronald Fisher.
- Formula: , where are partial p-values.
- Resulting statistic follows a distribution with degrees of freedom.
- **Example in R:**

```
combined_stat <- -2 * sum(log(p_values))
df <- 2 * length(p_values)
p_fisher <- 1 - pchisq(combined_stat, df)
```

- **Advantage:** Sensitive to small p-values.

# Fisher's Combination of Partial P-values

- Content:
- Statistic: Fisher Combined p-value = 0
- Explanation:
  - The p-value being zero implies **strong overall significance**.
  - Good demonstration of **multiple test combination**.

# Tippett's Combination Method

- Uses the **minimum** of partial p-values.
- Global p-value:
- Very sensitive to **a single small p-value**, but may miss broader signals.
- **Example:**

```
p_tippett <- 1 - (1 - min(p_values))^length(p_values)
```

- **Use case:** When one variable is highly dominant.

# Liptak's Combination Method

- Converts p-values to z-scores.
- Sum of z-scores is used as a test statistic.
- Can incorporate weights (e.g., based on variable importance).
- **Formula:**

$$Z = \sum w_i \Phi^{-1}(1 - p_i)$$

- **In R:**

```
z_scores <- qnorm(1 - p_values)
p_liptak <- 1 - pnorm(mean(z_scores))
```

- **Advantage:** Balances moderate effects across variables.

# Bonferroni Correction

- To control the **family-wise error rate**, Bonferroni correction adjusts p-values:

```
p_bonf <- p.adjust(p_values, method = "bonferroni")
```

- Ensures that overall significance level is maintained.
- Very conservative, but widely used in clinical studies.

# Bonferroni Adjustment of p-values

- Conclusion:
- Adjustment does **not affect HBA1C** significance.
- Confirms **ALBUMINA remains non-significant** even after correction.

Variable	Original p-value	Bonferroni Adjusted
HBA1C	1.043841e-154	2.087681e-154
ALBUMINA	0.2672962	0.5345925

# Summary – NPC Tests

---

**Partial p-values** give insight into individual variables.

---

**Combination methods** derive global significance.

---

**Fisher:** Good for multiple small effects.

---

**Tippett:** Highlights extreme values.

---

**Liptak:** Captures balance across variables.

---

All operate **non-parametrically**, making them robust.

# NPC Ranking Methods – Theoretical Foundation

- **NPC Ranking** is an extension that evaluates subjects **jointly** across multiple variables. Steps:
  - Rank values for each variable.
  - Sum ranks across variables → global severity score.
  - Sort by global score → identifies most at-risk patients.
- References:
  - Pesarin & Salmaso (2010): *Permutation Tests for Complex Data*
  - Bonnini (2014): *Nonparametric Statistics for Complex Data*



# Why NPC Ranking?



Allows for **multivariate scoring** without relying on scale or distribution.



Gives a **global view** of patient condition.



Especially useful in **clinical prioritization** or patient triaging.



Ensures equal treatment of binary and continuous variables.

# NPC Ranking – R Implementation

```
ranked <- data %>%
```

```
mutate(across(everything(), ~ rank(., ties.method =  
"average"), .names = "RANK_{.col}")) %>%  
  
mutate(RANK_GLOBALE = rowSums(select(.,  
starts_with("RANK_")))) %>%  
arrange(RANK_GLOBALE)
```

**Each row receives a total score from the ranks.**

**Lower score → higher severity.**



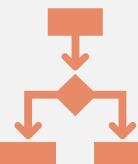
# Interpreting NPC Global Ranks



**Boxplots and density curves** help assess distribution of scores.



Outliers with low global ranks should be flagged for clinical review.



Enables creation of severity thresholds for decision-making.

## **Summary – NPC Ranking**

---

Ranks patients using a **non-parametric, multivariate** strategy.

---

Requires no assumption about variable type or distribution.

---

Helps identify **most critical cases** based on data.

---

Can be used for **risk scoring, triage, or prognostic modeling**.

# Introduction to Bootstrap Resampling

- Bootstrap is a **resampling technique** for estimating statistics (mean, median, etc.) from a sample:
- Randomly sample **with replacement**.
- Generate many replicates of the statistic.
- Build **confidence intervals** from empirical distributions.
- **Advantages:**
  - No assumptions about normality.
  - Captures variability, especially useful for small sample sizes.



# Bootstrap – Code Implementation

- We estimate the mean of HBA1C and Albumin using 1000 resamples:

```
boot_func <- function(x, indices) mean(x[indices])  
boot_results <- list(  
  HBA1C = boot(data$HBA1C, statistic = boot_func, R = 1000),  
  ALBUMINA = boot(data$ALBUMINA, statistic = boot_func, R = 1000)  
)
```

- This provides robust estimates for central tendencies.

# Bootstrap Output Summary – HBA1C

```
Bootstrap Statistics :
```

original	bias	std. error
0.08915	-0.00008	0.00326

## Interpretation:

- The estimate is **stable**, as bias is negligible.
- The standard error is low, indicating **high precision**.

# Bootstrap Output Summary – Albumin

```
Bootstrap Statistics :
```

original	bias	std. error
2.708	0.00268	0.0816

## Interpretation:

- Mean Albumin close to the clinical threshold (2.8 g/dL).
- Bias is minimal; standard error indicates moderate variability.

# Bootstrap Estimations for HBA1C and ALBUMINA

- Interpretation:
- **Reliable mean estimates** with small bias and error.
- Variability in albumin levels is higher than in HBA1C.

Variable	Mean	Bias	Std. Error
HBA1C	0.0892	-0.000036	0.00318
ALBUMINA	2.7084	-0.00197	0.07930

# Percentile Confidence Intervals – Concept



**Percentile CI:** A non-parametric interval based on empirical quantiles.



$95\% \text{ CI} = [\text{2.5th percentile}, \text{97.5th percentile}]$  of the bootstrap distribution.



Does not assume symmetry.



Useful for **skewed or non-normal** distributions.

# Bootstrap 95% Confidence Intervals

Content:

- **HBA1C:** (0.0830, 0.0957)
- **ALBUMINA:** (2.551, 2.866)

Interpretation:

- HBA1C levels are **precisely estimated and very low.**
- ALBUMINA is more variable, but within normal range.



# Percentile CIs – HBA1C

95% CI: (0.0827, 0.0958)

## Interpretation:

- All bootstrap means are well **below** the clinical threshold of 7.
- Strong evidence of **hypoglycemia** or measurement issues in this cohort.

# Percentile CIs – Albumin

95% CI: (2.560, 2.875)

## Interpretation:

- Upper limit exceeds clinical cutoff (2.8), indicating variability.
- Could reflect **nutritional heterogeneity** in patients.

# Visualization – NPC Global Rank Boxplot

```
ggplot(ranked, aes(x = factor(1), y = RANK_GLOBALE)) +  
  geom_boxplot(fill = "#A6CEE3") +  
  geom_jitter(width = 0.2, alpha = 0.5, color = "darkblue") +  
  labs(title = "Distribution of NPC Global Ranks",  
       subtitle = "Lower global rank = more severe case",  
       x = "Patients", y = "Global Rank")
```

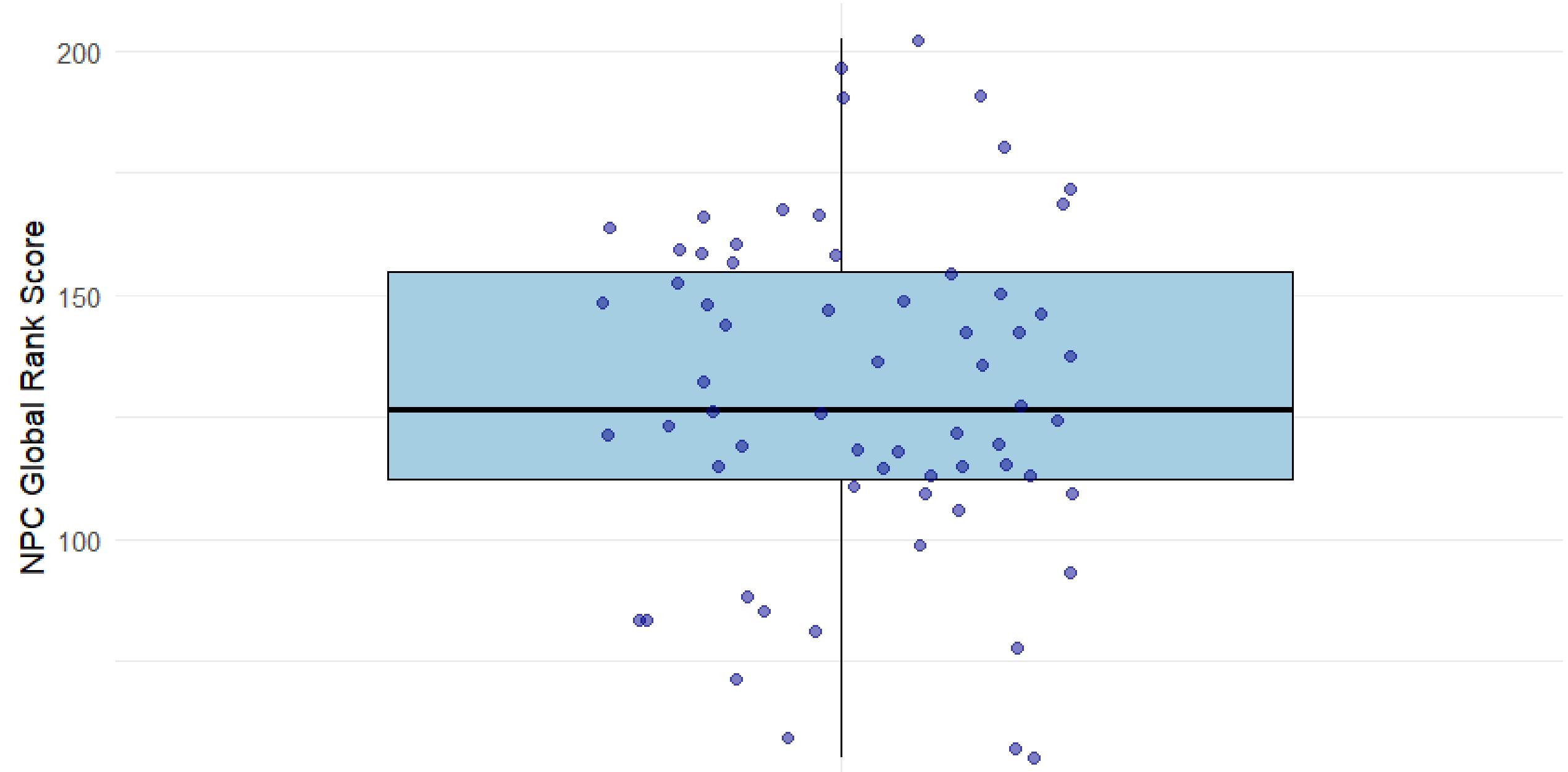
- Shows **variability and severity** across patients.
- RANK\_GLOBALE distribution

Interpretation:

- Spread of severity
- Some patients cluster at **low global ranks** (more severe)

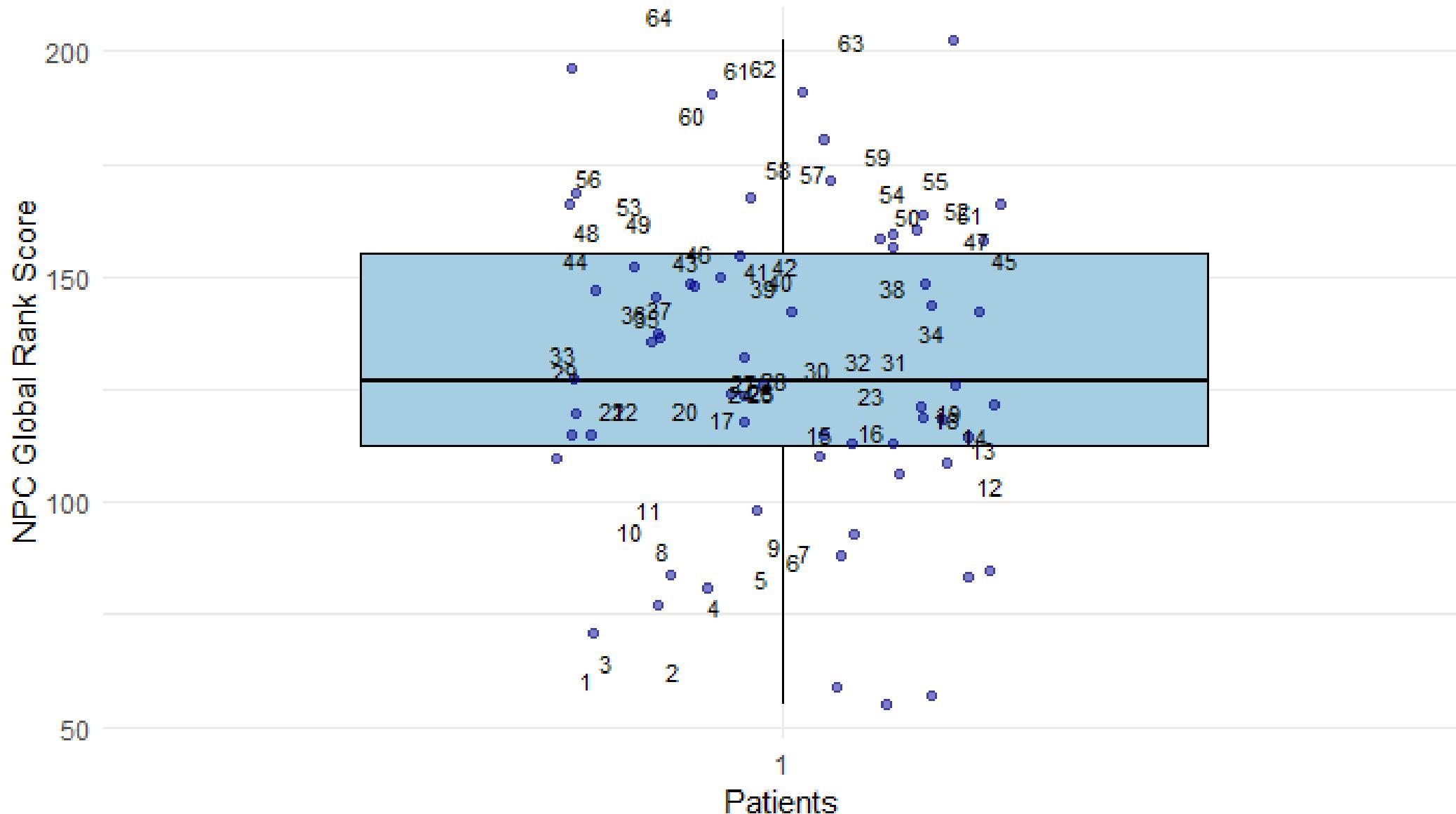
# Distribution of NPC Global Ranks

Lower global rank indicates more severe cases

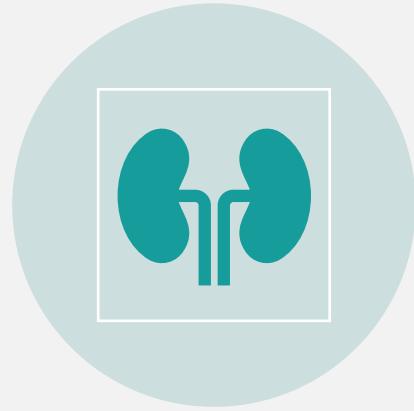


# Distribution of NPC Global Ranks

Lower global rank indicates more severe cases



# Interpretation – NPC Rank Distribution



PATIENTS WITH **LOWEST RANK VALUES** =  
MOST SEVERE (I.E., LOWEST ALBUMIN,  
LOWEST HBA1C, ETC.).



USEFUL FOR TRIAGING OR  
PRIORITYZATION.



BOXPLOT REVEALS POTENTIAL  
OUTLIERS – CASES WITH EXTREMELY  
LOW OR HIGH GLOBAL RANKS.

# Top 10 Severe Cases – NPC Output Table

```
print(head(ranked, 10))
```

**Display patient data for:**

- Amputation status
- HBA1C
- Mortality
- Albumin
- Individual & global ranks

Use this list for clinical **review or intervention targeting**.

# Heatmap of Correlation Matrix

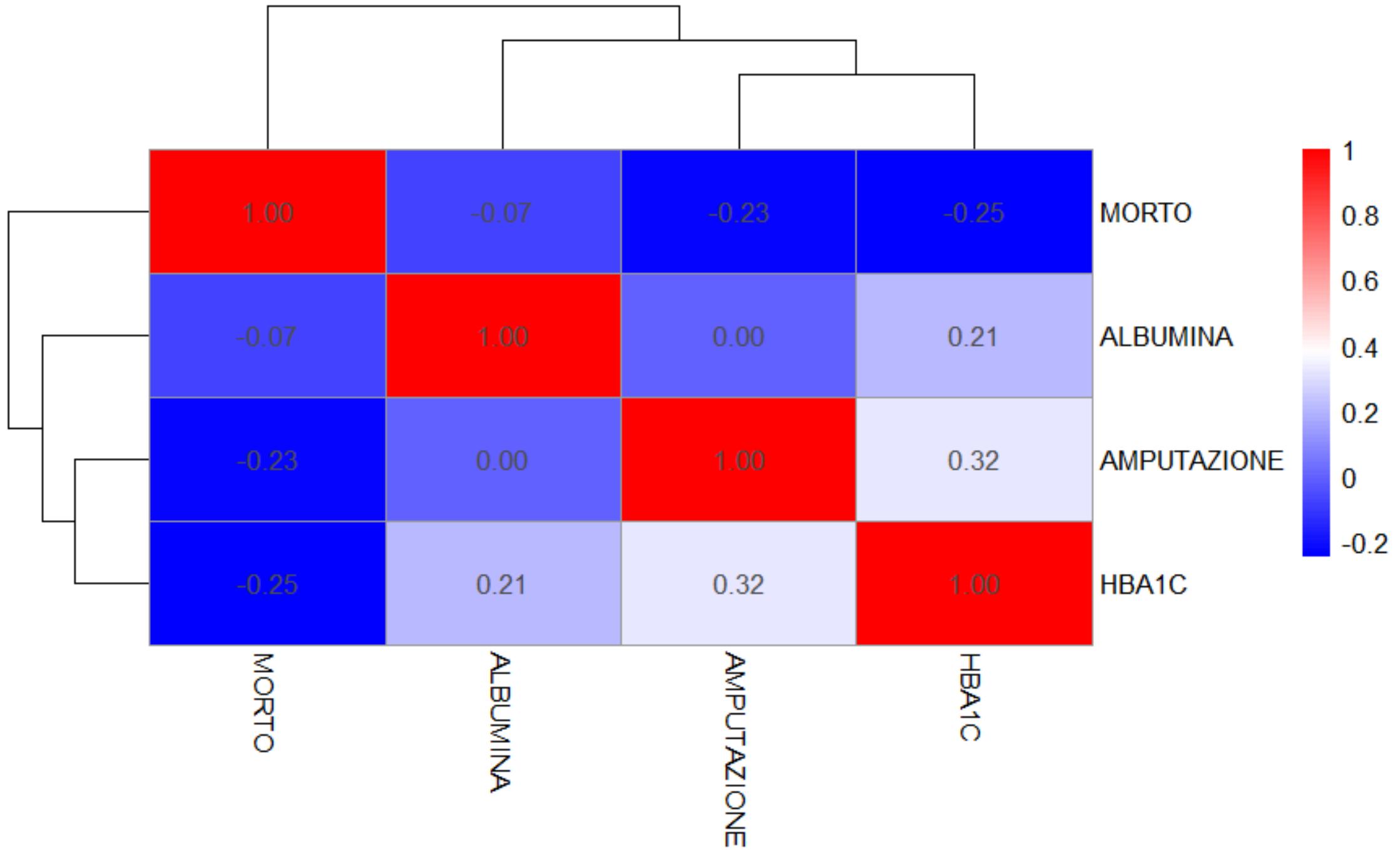
```
pheatmap(cor(data),  
         main = "Correlation Between Clinical Variables",  
         display_numbers = TRUE,  
         fontsize_number = 10,  
         color = colorRampPalette(c("blue", "white", "red"))(100))
```

Visual representation of **pairwise correlations** between clinical metrics.

Explanation:

- Identify relationships (positive/negative) between variables
- Especially useful for understanding **data structure**

# Correlation Between Clinical Variables



# Interpreting the Heatmap



Values close to  $\pm 1$  indicate **strong correlations**.

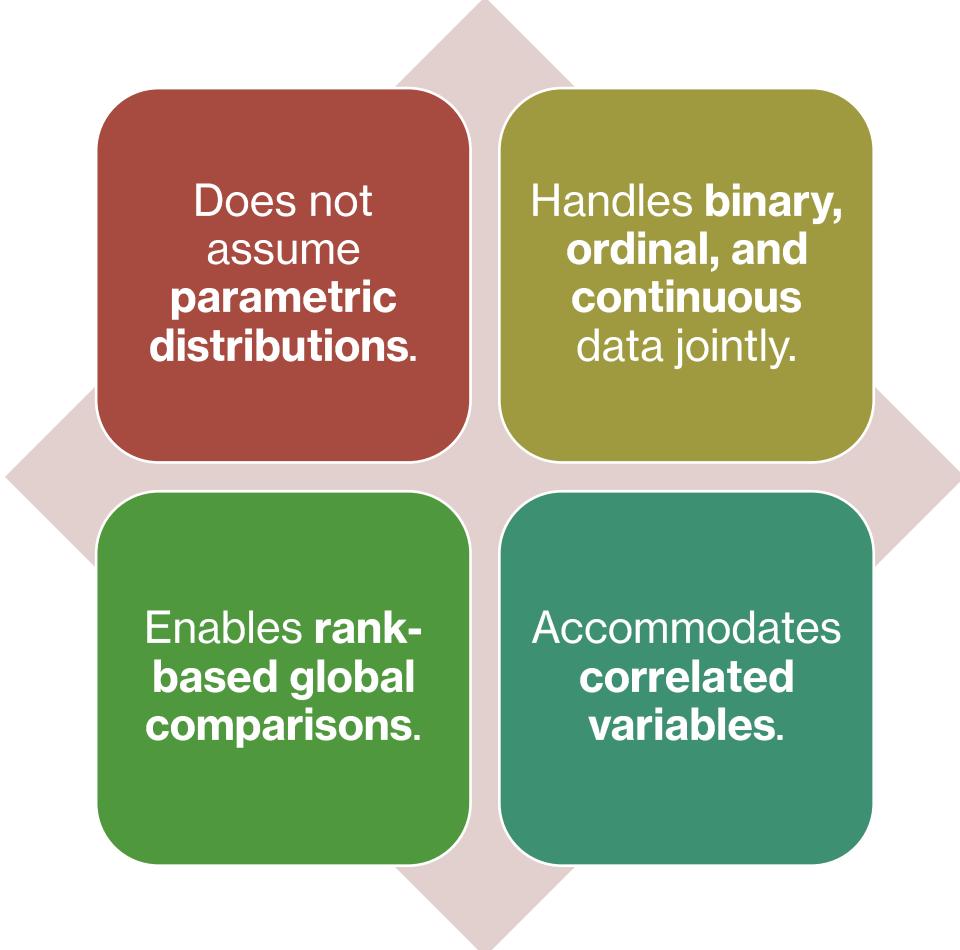


Use to assess variable **redundancy or interaction**.



Helps determine which combinations may drive outcomes.

# Advantages of NPC Methodology in Clinical Studies



Does not assume **parametric distributions**.

Handles **binary, ordinal, and continuous** data jointly.

Enables **rank-based global comparisons**.

Accommodates **correlated variables**.

# Summary of Key Statistical Findings

---

**HBA1C** is significantly lower than clinical threshold.

---

**Albumin** shows marginal deviation.

---

Bootstrap confirms findings with narrow confidence intervals.

---

NPC Global Ranking effectively identifies high-risk patients.

# Clinical Implications



Patients with low NPC ranks may benefit from **early intervention**.



Joint evaluation of variables gives a **more holistic** risk profile.



Can be embedded into **hospital triage** or risk scoring systems.

# Future Extensions



Include more clinical variables (e.g., WBC, CRP, temperature).



Perform **longitudinal** NPC tracking.



Integrate machine learning for **predictive modeling**.



Validate model with **external datasets**.

# Limitations

Small sample size  
may affect  
generalizability.

Variable thresholds  
are **clinical  
heuristics**, not  
strict rules.

Possible  
measurement  
errors (e.g., HBA1C  
near-zero values).

# Conclusion

---

This project successfully applied:

---

NPC Tests for robust hypothesis testing.

---

Fisher, Tippett, and Liptak methods to combine insights.

---

Bootstrap methods to estimate variability.

---

NPC Ranking to stratify patient severity.

---

These techniques enhance the **statistical rigor** and **clinical relevance** of data analysis in necrotizing fasciitis.

# Acknowledgements

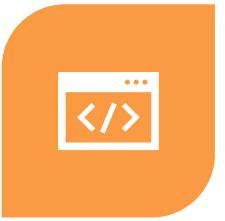
- Supervisor and university faculty
- Clinical data providers
- R development community
- Research colleagues and peers



# References

- Pesarin, F. (2001). *Multivariate Permutation Tests: With Applications in Biostatistics*. Wiley.
- Bonnini, S., Corain, L., Pesarin, F., Salmaso, L. (2014). *Nonparametric Statistics for Complex Data*. Wiley.
- Fisher, R.A. (1932). *Statistical Methods for Research Workers*.
- Tippett, L.H.C. (1931). *The Methods of Statistics*.
- Lipták, T. (1958). *On the combination of independent tests*.
- Bonferroni, C. (1936). *Teoria statistica delle classi e calcolo delle probabilità*.

# Appendix – Full R Code



**INCLUDED CODE  
SEGMENTS:**



DATA PREPROCESSING  
AND CLEANING



NPC CLASSICAL TESTS  
(T-TESTS, PARTIAL P-  
VALUES)



P-VALUE  
COMBINATION  
(FISHER'S METHOD,  
BONFERRONI)



BOOTSTRAP  
FUNCTIONS AND  
CONFIDENCE  
INTERVALS



NPC GLOBAL RANKING  
AND SEVERITY  
SCORES



VISUALIZATIONS:  
BOXPLOTS, JITTER,  
HEATMAPS

# Appendix – Data Transformation Code

```
data <- raw_data %>%
  transmute(
    AMPUTAZIONE = case_when(
      `AMPUTAZIONE MAGGIORE` %in% c("SI", "SÌ", "1") ~ 1,
      `AMPUTAZIONE MAGGIORE` %in% c("NO", "0") ~ 0,
      TRUE ~ NA_real_
    ),
    HBA1C = suppressWarnings(as.numeric(HBA1C)),
    MORTO = case_when(
      DECESSO %in% c("SI", "SÌ", "1", "MORTO") ~ 1,
      DECESSO %in% c("NO", "0", "VIVO") ~ 0,
      TRUE ~ NA_real_
    ),
    ALBUMINA = suppressWarnings(as.numeric(ALBUMINA))
  ) %>%
  filter(complete.cases(.)) %>%
  as.data.frame()
```

# Appendix – NPC Partial P-values Code

```
p_values <- c(  
  HBA1C = tryCatch(t.test(data$HBA1C, mu = 7)$p.value, error = function(e) NA),  
  ALBUMINA = tryCatch(t.test(data$ALBUMINA, mu = 2.8)$p.value, error = function(e)  
NA)  
)
```

This code tests whether the sample means differ from clinical reference values.

# Appendix – Fisher’s Combination Test

```
combined_stat <- -2 * sum(log(p_values), na.rm = TRUE)
df <- 2 * sum(!is.na(p_values))
p_fisher <- 1 - pchisq(combined_stat, df)
```

**Fisher’s method combines independent p-values into a single statistic using a chi-square distribution.**

# Appendix – Bonferroni Correction

```
p_bonf <- p.adjust(p_values, method = "bonferroni")
```

- Adjusts each p-value to control the family-wise error rate in multiple testing.

# Appendix – Bootstrap Function Code

```
boot_func <- function(x, indices) mean(x[indices])  
boot_results <- list(  
  HBA1C = boot(data$HBA1C, statistic = boot_func, R = 1000),  
  ALBUMINA = boot(data$ALBUMINA, statistic = boot_func, R = 1000)  
)
```

This section estimates the sampling distribution of the mean using bootstrapping.

# Appendix – Bootstrap Confidence Intervals

```
boot.ci(boot_results$HBA1C, type = "perc")
boot.ci(boot_results$ALBUMINA, type = "perc")
```

- Displays percentile confidence intervals for HBA1C and Albumin based on resampling.

# Appendix – NPC Ranking Code

```
ranked <- data %>%
  mutate(across(everything(), ~ rank(., ties.method = "average"), .names = "RANK_{.col}"))
%>%
  mutate(RANK_GLOBALE = rowSums(select(., starts_with("RANK_")))) %>%
  arrange(RANK_GLOBALE)
```

Ranks each patient by severity, using all variables and producing a global severity score.

# Appendix – Boxplot and Jitter Plot

```
ggplot(ranked, aes(x = factor(1), y = RANK_GLOBALE)) +  
  geom_boxplot(fill = "#A6CEE3") +  
  geom_jitter(width = 0.2, alpha = 0.5, color = "darkblue") +  
  labs(title = "Distribution of NPC Global Ranks",  
       subtitle = "Lower global rank = more severe case",  
       x = "Patients", y = "Global Rank")
```

- Visual tool to represent dispersion of rank scores.

# Appendix – Heatmap Correlation Code

```
pheatmap(  
  cor(data, use = "complete.obs"),  
  main = "Correlation Between Clinical Variables",  
  display_numbers = TRUE,  
  fontsize_number = 10,  
  color = colorRampPalette(c("blue", "white", "red"))(100)  
)
```

Provides a visual correlation matrix to identify relationships between variables.

# Appendix – NPC vs Traditional Tests Comparison

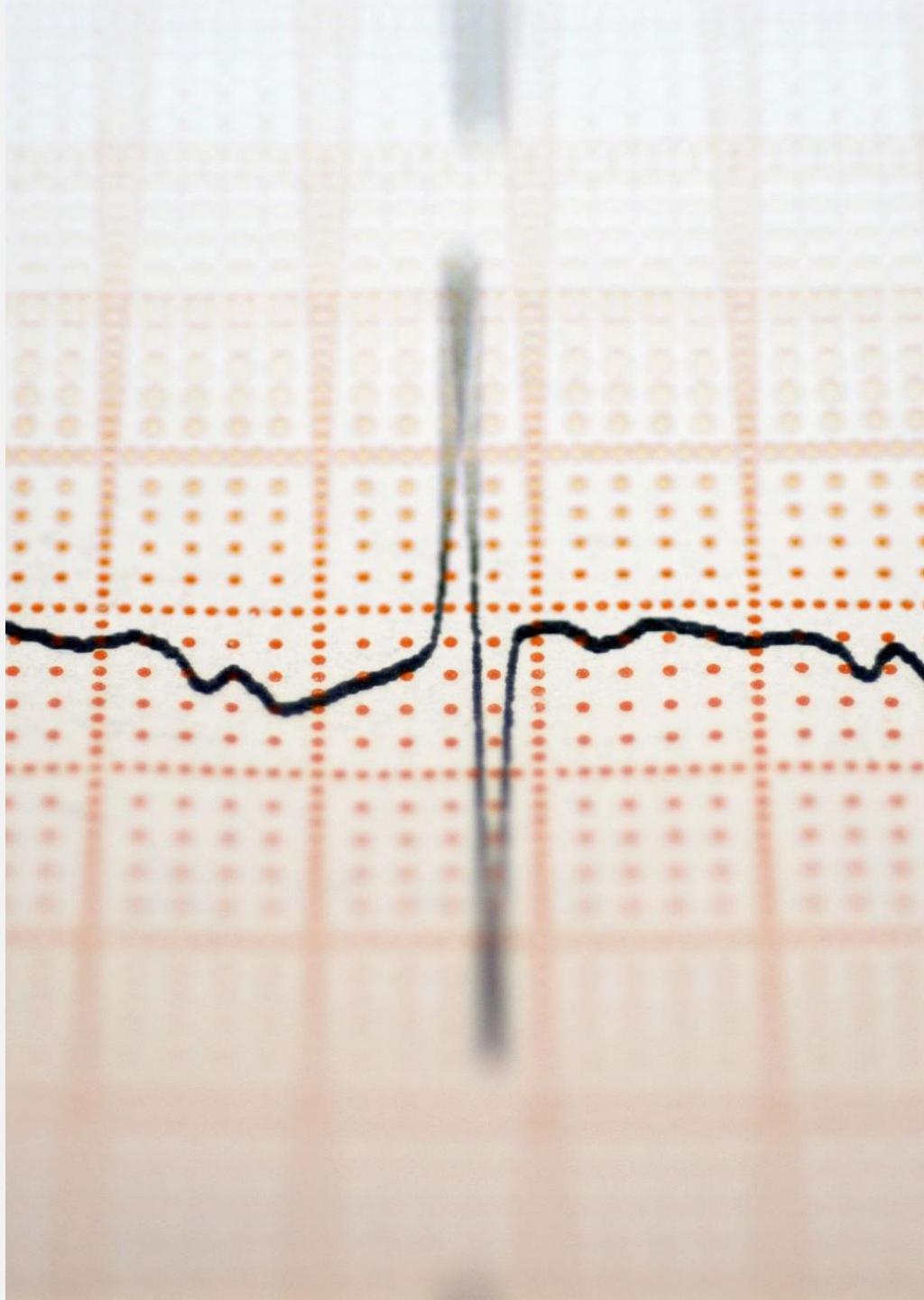
Feature	NPC Test	Traditional Test
Assumptions	Non-parametric	Often parametric
Handles Mixed Data	Yes	No
Robust to Outliers	Yes	Less robust
Global View	Yes	Usually single-variable
Suitable for Small Samples	Yes	Often No

# Patient Case Summaries (Example Format)

## Patient #1:

- HBA1C = 0.048 (very low)
- Albumin = 1.90
- Morto = 0 (Alive)
- Amputazione = 0 (No)
- Global Rank: 55.0 (most severe)

Repeat format for top 10 patients (low global ranks).



# Final Reflection



**This project illustrates the power of combining modern statistical tools:**



From theory (Pesarin, Bonnini) to real clinical insights.



Emphasizes transparency, reproducibility, and rigor.



Prepared me for future challenges in data analytics and biomedical research.

```
# =====
# LOAD REQUIRED LIBRARIES
# =====
library(readxl)      # For reading Excel files
library(dplyr)        # For data manipulation
library(boot)         # For bootstrap analysis
library(ggplot2)       # For visualizations
library(pheatmap)     # For heatmap of correlations

# =====
# STEP 1: LOAD EXCEL DATA
# =====
# Prompt user to select dataset (e.g. Fasciti Necrotizzanti Excel)
file_path <- file.choose()
raw_data <- read_excel(file_path)
names(raw_data) <- toupper(names(raw_data)) # Convert column names to uppercase
```

```
# =====
# STEP 2: CLEAN AND PREPARE DATA
# =====
# Selecting and transforming four important clinical variables
data <- raw_data %>%
  transmute(
    AMPUTAZIONE = case_when(
      `AMPUTAZIONE MAGGIORE` %in% c("SI", "SÌ", "1") ~ 1,
      `AMPUTAZIONE MAGGIORE` %in% c("NO", "0") ~ 0,
      TRUE ~ NA_real_
    ),
    HBA1C = suppressWarnings(as.numeric(HBA1C)),
    MORTO = case_when(
      DECESSO %in% c("SI", "SÌ", "1", "MORTO") ~ 1,
      DECESSO %in% c("NO", "0", "VIVO") ~ 0,
      TRUE ~ NA_real_
    ),
    ALBUMINA = suppressWarnings(as.numeric(ALBUMINA))
  ) %>%
  filter(complete.cases(.)) %>%
  as.data.frame()

# =====
# STEP 3: NPC CLASSICAL TESTS
# =====
# Partial p-values from t-tests compared to known clinical cutoffs
p_values <- c(
  HBA1C     = tryCatch(t.test(data$HBA1C, mu = 7)$p.value, error = function(e) NA),
  ALBUMINA = tryCatch(t.test(data$ALBUMINA, mu = 2.8)$p.value, error = function(e) NA)
)
```

```
# Fisher's method to combine p-values into one test
combined_stat <- -2 * sum(log(p_values), na.rm = TRUE)
df <- 2 * sum(!is.na(p_values))
p_fisher <- 1 - pchisq(combined_stat, df)

# Bonferroni correction for multiple comparisons
p_bonf <- p.adjust(p_values, method = "bonferroni")

cat("\n--- Partial p-values ---\n")
print(p_values)

cat("\nFisher Combined p-value:", round(p_fisher, 5), "\n")

cat("\n--- Bonferroni Corrected p-values ---\n")
print(p_bonf)

# =====
# STEP 4: BOOTSTRAPPING MEANS
# =====
boot_func <- function(x, indices) mean(x[indices])
boot_results <- list(
  HBA1C = boot(data$HBA1C, statistic = boot_func, R = 1000),
  ALBUMINA = boot(data$ALBUMINA, statistic = boot_func, R = 1000)
)

cat("\n--- Bootstrap Results Summary ---\n")
print(boot_results)

cat("\n--- Bootstrap Confidence Intervals (Percentile Method) ---\n")
print(boot.ci(boot_results$HBA1C, type = "perc"))
print(boot.ci(boot_results$ALBUMINA, type = "perc"))
```

```
# =====
# STEP 5: NPC RANKING METHOD
# =====
# Rank each variable and compute a global score (lower = worse condition)
ranked <- data %>%
  mutate(across(everything(), ~ rank(., ties.method = "average"), .names = "RANK_{.col}"))
  mutate(RANK_GLOBALE = rowSums(select(., starts_with("RANK_")))) %>%
  arrange(RANK_GLOBALE)

cat("\n--- Top 10 Most Severe Cases (lowest global ranks) ---\n")
print(head(ranked, 10))
# =====
# STEP 6: VISUALIZE RESULTS
# =====

# Add a Patient ID column (row number) for labeling
ranked$PATIENT_ID <- seq_len(nrow(ranked))

# Boxplot of global ranking with patient numbers labeled
ggplot(ranked, aes(x = factor(1), y = RANK_GLOBALE)) +
  geom_boxplot(fill = "#A6CEE3", color = "black") +
  geom_jitter(aes(label = PATIENT_ID), width = 0.2, alpha = 0.5, color = "darkblue") +
  geom_text(aes(label = PATIENT_ID), position = position_jitter(width = 0.2), vjust = -0.8,
  labs(
    title = "Distribution of NPC Global Ranks",
    subtitle = "Lower global rank indicates more severe cases",
    x = "Patients",
    y = "NPC Global Rank Score"
  ) +
  theme_minimal()
```

```
# Heatmap of correlations
pheatmap(
  cor(data, use = "complete.obs"),
  main = "Correlation Between Clinical Variables",
  color = colorRampPalette(c("blue", "white", "red"))(100),
  display_numbers = TRUE,
  fontsize_number = 10
)

# -----
# STEP 7: SAVE FINAL OUTPUT
# -----
results <- list(
  partial_p_values = p_values,
  fisher_combined = p_fisher,
  bonferroni = p_bonf,
  bootstrap = boot_results,
  top_ranked = head(ranked, 10)
)

cat("\n Analysis Complete. Review summary in 'results' object.\n")
```

# Thank You

**Email:** [saitesaroyal2311@gmail.com](mailto:saitesaroyal2311@gmail.com)

**Phone:** Only on request through the Email

**LinkedIn :** [www.linkedin.com/in/saitesaroyal2311/](https://www.linkedin.com/in/saitesaroyal2311/)

