Statistical Analysis of Nodal Network Structures

2025-03-22

Dataset Information

Let's read in two datasets containing graph theory measures of patients who were sleeping while undergoing EEG. The target variable is Dreams, a binary (0,1) value that indicates if a patient reported a conscious experience. The data is messy, let's read it in, inspect it, and do some pre-processing. We will need a non-prametric test to compare the two groups, assumption violations are showcased on GitHub: .

```
suppressPackageStartupMessages({
  library(readr)
  library(dplyr)
  library(tidyr)
})
```

```
## Warning: package 'tidyr' was built under R version 4.4.3
```

```
df1 <- read_csv("C:/Users/User/Desktop/group1.csv", show_col_types = FALSE)
df2 <- read_csv("C:/Users/User/Desktop/group2.csv", show_col_types = FALSE)
#head(df1, 3);
head(df2, 3)
```

```
## # A tibble: 3 x 10
##   Subject  Band  threshold wGlobEff wTrans mean_wBetw bTrans mean_bBetw Dreams
##   <chr>    <chr>     <dbl>    <dbl>  <dbl>      <dbl>  <dbl>      <dbl>  <dbl>
## 1 sub_2012 delta     0.202    0.125  0.105       58.3  0.439       151.      0
## 2 sub_2010 beta      0.181    0.164  0.125      132.   0.309       201.      0
## 3 sub_2010 alpha     0.294   NA      0.151       47.8  0.302       233.      0
## # i 1 more variable: degree <dbl>
```

```
process_combined_data <- function(df1, df2) {
  # Combine both data frames in a row-wise manner
  combined <- rbind(df1, df2)
  # Number of rows before dropping NA values
  n_before <- nrow(combined)
  # Remove rows with any NA values
  combined_clean <- tidyr::drop_na(combined)
  # Number of rows after dropping NA values
  n_after <- nrow(combined_clean)
  # print the total number of rows dropped
  cat("Dropped", n_before - n_after, "rows containing NA values.\n")
  # Convert the Dreams column: 0 -> "No Dreams", 1 -> "Dreaming"
  combined_clean$Dreams <- factor(
    combined_clean$Dreams,
```

```
    levels = c(0, 1),
    labels = c("No Dreams", "Dreaming")
  )
  # Create a new column 'obs' to track observation number
  combined_clean$obs <- seq_len(nrow(combined_clean))
  # Return the cleaned, merged dataset
  return(combined_clean)
}
processed_data <- process_combined_data(df1, df2)
```

```
## Dropped 103 rows containing NA values.
```

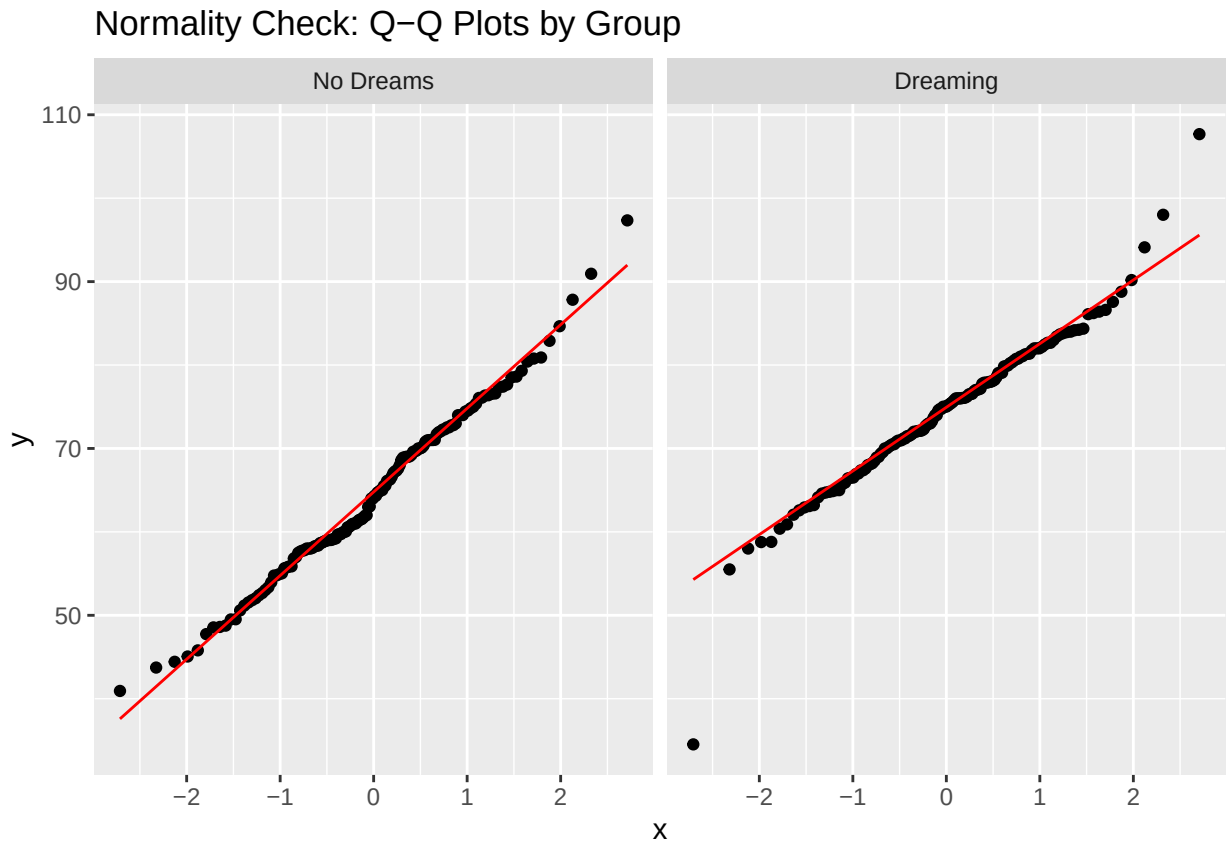
Statistical Assumption Checking

Normality Tests

```
# Shapiro-Wilk test for normality by group
shapiro_results <- processed_data %>%
  group_by(Dreams) %>%
  summarise(
    statistic = shapiro.test(degree)$statistic,
    p.value = shapiro.test(degree)$p.value
  )
print(shapiro_results)
```

```
## # A tibble: 2 x 3
##   Dreams    statistic  p.value
##   <fct>         <dbl>    <dbl>
## 1 No Dreams     0.990 0.375
## 2 Dreaming      0.959 0.000253
```

```
# QQ Plots for further normality tests
suppressPackageStartupMessages(library(ggplot2))
ggplot(processed_data, aes(sample = degree)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  facet_wrap(~Dreams) +
  ggtitle("Normality Check: Q-Q Plots by Group")
```



Shapiro-Wilk test showed significant evidence against the null hypothesis of normality, the powerful non-normality is evidenced by p-values **(p = 0.00025)** in the dreaming group, the non-dreaming group appears normal with **(p = 0.3752400104)**. This test is further supported by the Dreaming group's qq-plot showing heavier tails or skewness compared to a normal distribution via the upward curve in the high x-range.

Homogeneity of Variance test

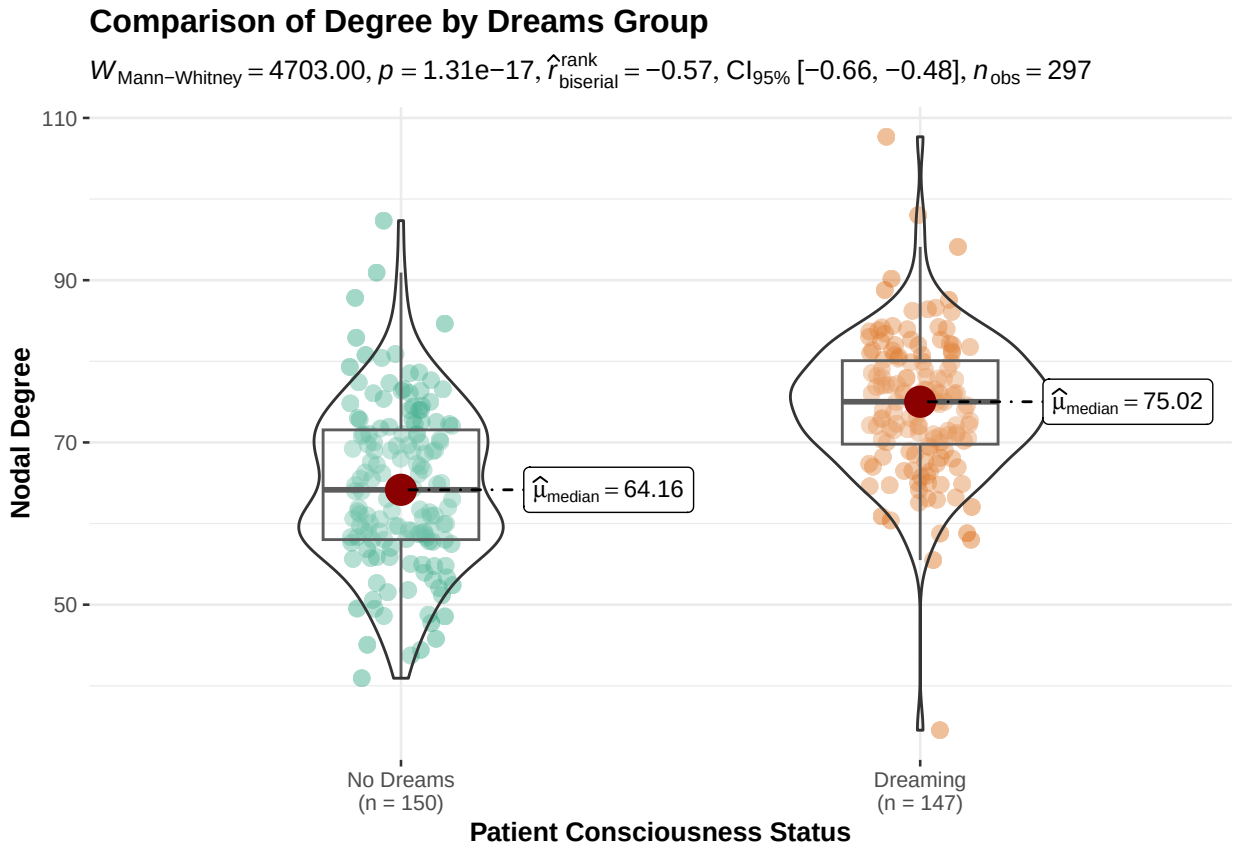
```
# Levene's Test to see how to set the eq.val argument in ggbetweenstats()
suppressPackageStartupMessages(library(car))
leveneTest(degree ~ Dreams, data = processed_data, center = median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value   Pr(>F)
## group   1  6.9179 0.008982 **
##       295
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Levene's test rejects homogeneity of variance, and assumes heterogeneity of variances **(p = 0.009)**

Based on the prior statistical tests, we will use Mann-Whitney U test. The Mann-Whitney U test does not assume normality or equal variances and is robust to outliers and skewed distributions. This matches the data'a characteristics

```
suppressPackageStartupMessages(library(ggstatsplot))
ggbetweenstats(
  data = processed_data,
  x = Dreams,              # 'Dreams' is a factor that indicates group membership
  y = degree,              # 'degree' is the graph measure to compare the two groups
  type = "nonparametric", # Nonparametric test is selected
  bf.message = FALSE,      # set to TRUE for postier probabilities
  xlab = "Patient Consciousness Status",
  ylab = "Nodal Degree",
  title = "Comparison of Degree by Dreams Group",
  messages = FALSE
)
```



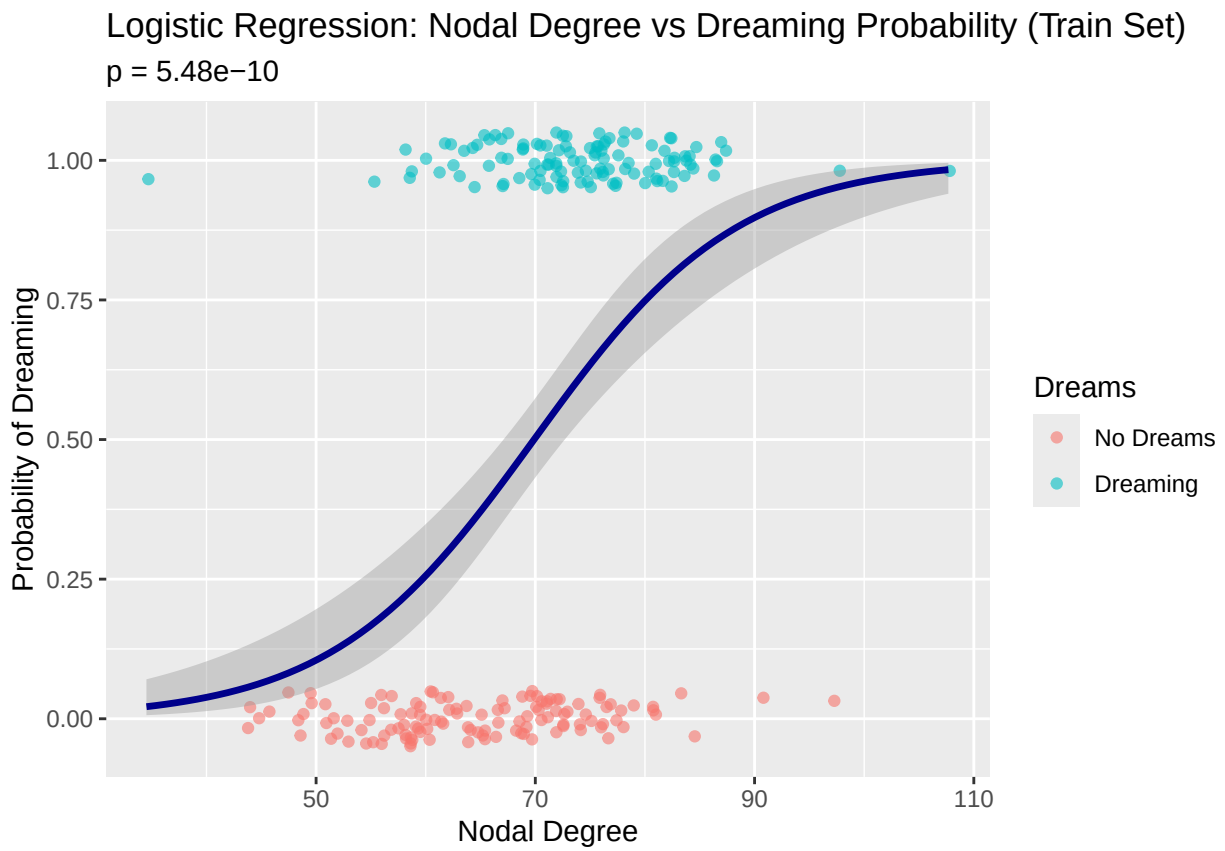
Patients who reported dreaming had significantly lower nodal degree (a network connectivity measure) compared to those who did not, with a clinically meaningful effect, as evidenced by a large effect size **(r = -0.57)**, non-overlapping confidence intervals **(CI: [-0.66, -0.48])**, and extreme statistical significance **($p < 0.05$)**

While statistical differences don't always translate to predictive power, I want to see how a model might do, let's try logistic regression.

```
suppressPackageStartupMessages({library(pROC)})
# 1) Split the data
set.seed(222)  # I'll set a seed for hold-out reproducibility
train_index <- sample(seq_len(nrow(processed_data)), size = 0.8 * nrow(processed_data)) # 80% train
train_data  <- processed_data[train_index, ]
test_data   <- processed_data[-train_index, ] # 20% test
```

```r
logit_model <- glm(
  Dreams ~ degree, # Dream is target, degree is our predictor
  data   = train_data,
  family = binomial() # Sets glm() to a logistic regression
)
model_summary <- summary(logit_model)
# Lets visualize the logistic (sigmoid) curve and p-value
ggplot(train_data, aes(x = degree, y = as.numeric(Dreams == "Dreaming"))) +
  # Using geom_jitter to offset points to avoid clutter
  geom_jitter(aes(color = Dreams), height = 0.05, width = 0.5, alpha = 0.6) +
  geom_smooth(
    method       = "glm",
    method.args  = list(family = "binomial"),
    se           = TRUE,
    color        = "darkblue",
    linewidth    = 1.2
  ) +
  labs(
    x       = "Nodal Degree",
    y       = "Probability of Dreaming",
    title   = "Logistic Regression: Nodal Degree vs Dreaming Probability (Train Set)",
    # Extract p-value from coefficient #2 (degree)
    subtitle = paste("p =", format.pval(model_summary$coefficients[2, 4], digits = 3))
  )
```

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

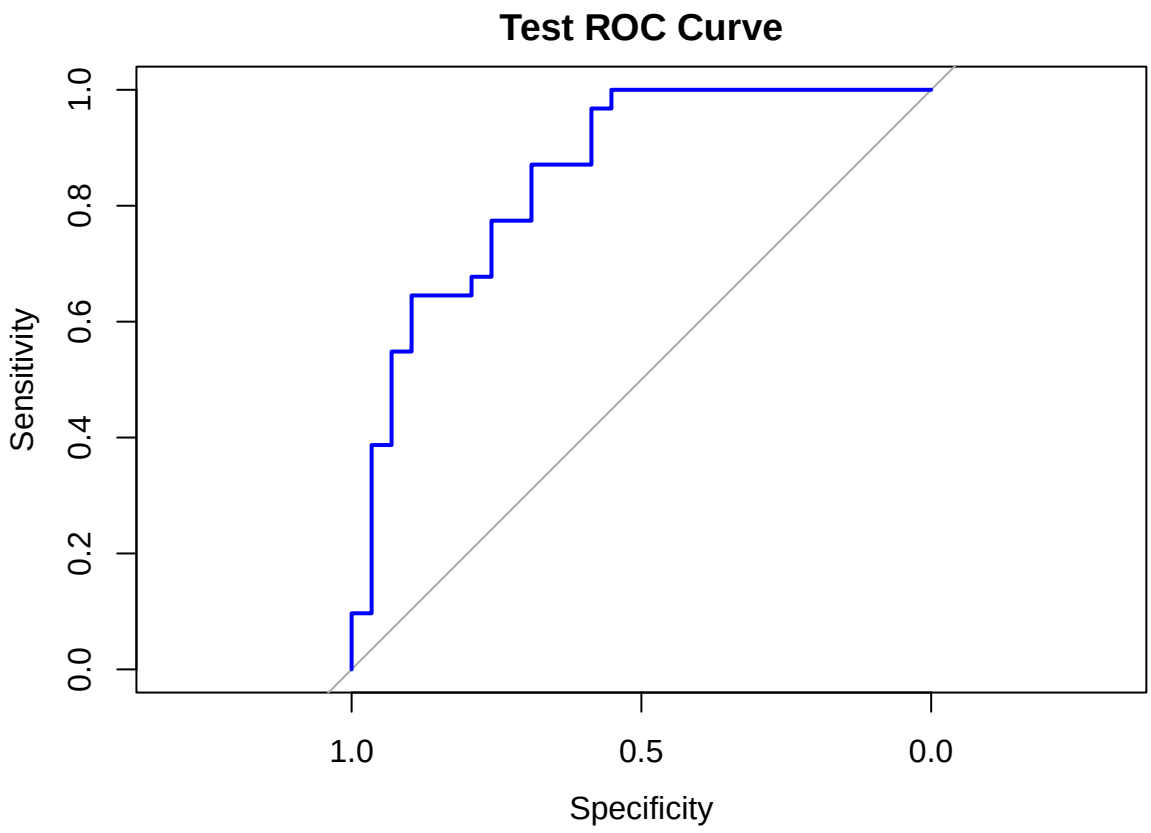



```r
# Evaluation on test data
test_predictions <- predict(logit_model, newdata = test_data, type = "response")
test_roc <- roc(
  response  = as.numeric(test_data$Dreams == "Dreaming"),
  predictor = test_predictions
)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```r
plot(test_roc, main = "Test ROC Curve", col = "blue")
```




```
cat("Test AUC:", auc(test_roc), "\n")
```

```
## Test AUC: 0.8542825
```

```
# ROC of 0.85 indicates a classifier discriminating much better than a random-guessing model
#(illustrated via diagonal line)
```