41_Statistical_Analysis

2025-12-06

```r
# Data loading
df_reg <- readr::read_csv("Regular Season.csv")
```

```
## New names:
## Rows: 570 Columns: 15
## -- Column specification
## ---------------------------------------------------------- Delimiter: "," chr
## (2): Player, Team dbl (13): ...1, Age, GP, W, L, Min, PTS, FGM, FGA, FG%, 3PM/,
## 3PA, 3P%
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```r
df_playoff <- readr::read_csv("Playoffs.csv")
```

```
## New names:
## Rows: 219 Columns: 15
## -- Column specification
## ---------------------------------------------------------- Delimiter: "," chr
## (2): Player, Team dbl (13): ...1, Age, GP, W, L, Min, PTS, FGM, FGA, FG%, 3PM/,
## 3PA, 3P%
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```r
# Identify playoff teams
# Get a vector of unique team codes that appear in the playoff data
playoff_teams <- unique(df_playoff$Team)

# Aggregate regular season data to team level
df_reg_team <- df_reg %>%
  group_by(Team) %>%
  summarise(
    FGM_total = sum(FGM, na.rm = TRUE),
    FGA_total = sum(FGA, na.rm = TRUE),
    TPM_total = sum(`3PM/`, na.rm = TRUE),
    TPA_total = sum(`3PA`, na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  mutate(
    # Calculate team FG%
    FG_percent_team = (FGM_total / FGA_total) * 100,
    # Calculate team 3P%
```

```r
    TP_percent_team = ifelse(
      TPA_total > 0,
      (TPM_total / TPA_total) * 100,
      0
    )
  )

# Classify teams
df_comparison <- df_reg_team %>%
  mutate(
    Team_Status = ifelse(
      Team %in% playoff_teams,
      "Playoff",
      "Non-Playoff"
    )
  ) %>%
  select(Team, Team_Status, FG_percent_team, TP_percent_team)

assign("df_comparison", df_comparison, envir = .GlobalEnv)

# Cohen's d Calculation
# Define a function to calculate Cohen's d for two independent samples
cohen_d_independent <- function(group1, group2) {
  n1 <- length(group1)
  n2 <- length(group2)
  m1 <- mean(group1, na.rm = TRUE)
  m2 <- mean(group2, na.rm = TRUE)
  s1 <- sd(group1, na.rm = TRUE)
  s2 <- sd(group2, na.rm = TRUE)

  # Pooled Standard Deviation
  sp <- sqrt(((n1 - 1) * s1^2 + (n2 - 1) * s2^2) / (n1 + n2 - 2))

  # Cohen's d
  d <- (m1 - m2) / sp
  return(d)
}

# Prepare data for testing
# Separate the data into two groups for FG%
fg_playoff <- df_comparison %>% filter(Team_Status == "Playoff") %>% pull(FG_percent_team)
fg_non_playoff <- df_comparison %>% filter(Team_Status == "Non-Playoff") %>% pull(FG_percent_team)


# Separate the data into two groups for 3P%
tp_playoff <- df_comparison %>% filter(Team_Status == "Playoff") %>% pull(TP_percent_team)
tp_non_playoff <- df_comparison %>% filter(Team_Status == "Non-Playoff") %>% pull(TP_percent_team)

# Execute Statistical Comparisons (Unpaired T-tests)

# T-test for Field Goal Percentage (FG%)
# We use var.equal = FALSE for Welch's t-test, which assumes unequal variances.
t_test_fg <- t.test(fg_playoff, fg_non_playoff, var.equal = FALSE)
```

```r
cohen_d_fg <- cohen_d_independent(fg_playoff, fg_non_playoff)

# T-test for Three Point Percentage (3P%)
t_test_tp <- t.test(tp_playoff, tp_non_playoff, var.equal = FALSE)
cohen_d_tp <- cohen_d_independent(tp_playoff, tp_non_playoff)

# Print Results
cat("--- Field Goal Percentage (FG%) Comparison ---\n")
```

```
## --- Field Goal Percentage (FG%) Comparison ---
```

```r
cat("Playoff Mean (N=", length(fg_playoff), "): ", round(mean(fg_playoff), 2), "%\n", sep="")
```

```
## Playoff Mean (N=16): 45.65%
```

```r
cat("Non-Playoff Mean (N=", length(fg_non_playoff), "): ", round(mean(fg_non_playoff), 2), "%\n", sep="
```

```
## Non-Playoff Mean (N=14): 44.27%
```

```r
print(t_test_fg)
```

```
##
##  Welch Two Sample t-test
##
## data:  fg_playoff and fg_non_playoff
## t = 1.4963, df = 27.867, p-value = 0.1458
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5107985  3.2773919
## sample estimates:
## mean of x mean of y
##  45.64983  44.26653
```

```r
cat("Cohen's d: ", round(cohen_d_fg, 3), "\n\n")
```

```
## Cohen's d:  0.545
```

```r
cat("--- Three Point Percentage (3P%) Comparison ---\n")
```

```
## --- Three Point Percentage (3P%) Comparison ---
```

```r
cat("Playoff Mean (N=", length(tp_playoff), "): ", round(mean(tp_playoff), 2), "%\n", sep="")
```

```
## Playoff Mean (N=16): 33.78%
```

```r
cat("Non-Playoff Mean (N=", length(tp_non_playoff), "): ", round(mean(tp_non_playoff), 2), "%\n", sep="
```

```
## Non-Playoff Mean (N=14): 32.69%
```

```r
print(t_test_tp)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  tp_playoff and tp_non_playoff
## t = 1.0121, df = 23.373, p-value = 0.3218
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.143411  3.337883
## sample estimates:
## mean of x mean of y
##  33.78347  32.68623
```

```r
cat("Cohen's d: ", round(cohen_d_tp, 3), "\n")
```

```
## Cohen's d:  0.379
```

```r
write_csv(df_comparison, "team_accuracy_comparison_R.csv")
```