

# Hypothesis Testing

## Slides

Chapter slides [here](#). (To convert html to pdf, press E → Print → Destination: Save to pdf)

## R code

```
#####  
# This code generates the numerical results in chapter 2      #  
#####  
  
# load the survival package  
library(survival)  
  
# install and load the WR package for  
# 1. dataset hfaction_cpx9;  
# 2. function WRrec() for win ratio test (of recurrent events and death)  
# 3. functions base() and WRSS() for sample size calculation  
# install.packages("WR")  
library(WR)  
library(tidyverse) # for data wrangling (dplyr, ggplot2, etc.)  
  
##### Read in HF-ACTION DATA #####  
# same as rmt::hfaction used in chap 1  
# (except for status coding)  
data(hfaction_cpx9)  
hfaction <- hfaction_cpx9  
head(hfaction)  
#> Shows the first few rows of hfaction_cpx9 dataset  
  
# count unique patients in each arm  
hfaction |>  
  group_by(trt_ab) |>
```

```

distinct(patid) |>
count(trt_ab)
#> This gives the number of unique patients (patid) by treatment arm (trt_ab)

#### demo #####
# WRrec() fits the recurrent event plus death model (Win Ratio approach)
obj <- WRrec(
  ID = hfaction$patid,
  time = hfaction$time,
  status = hfaction$status,
  trt = hfaction$trt_ab,
  strata = hfaction$age60,
  naive = TRUE
)
# summary results
obj
#> Displays the main results, including win ratio estimates for each method.

# LWR
beta <- obj$log.WR # log-win ratio for LWR
se <- obj$se       # standard error for log-win ratio (LWR)
# test
pval <- 2 * (1 - pnorm(abs(beta / se)))
pval
#> Two-sided p-value for LWR

# NWR
beta.naive <- obj$log.WR.naive # log-win ratio for naive WR (NWR)
se.naive <- obj$se.naive       # its standard error
# test
pval.naive <- 2 * (1 - pnorm(abs(beta.naive / se.naive)))
pval.naive
#> Two-sided p-value for NWR

# FWR
beta.FI <- obj$log.WR.FI # log-win ratio for Fisher's information-based WR (FWR)
se.FI <- obj$se.FI       # standard error for log(FWR)
# test
pval.FI <- 2 * (1 - pnorm(abs(beta.FI / se.FI)))
pval.FI
#> Two-sided p-value for FWR

```

```
#####
# Win ratio analyses: tabulate #
#####

data <- hfaction
### create a dataset with only the first hospitalization -> data.H1

# hospitalization data
tmpH <- data[data$status == 2, ]
# get the first record of each id
o <- order(tmpH$patid, tmpH$time)
tmpH <- tmpH[o, ]
tmpFH <- tmpH[!duplicated(tmpH$patid), ]

# combine it with mortality data
data.H1 <- rbind(tmpFH, data[data$status != 2, ])
o <- order(data.H1$patid, data.H1$time)
data.H1 <- data.H1[o, ]

# Function to create a summary table for
# PWR, NWR, FWR, and LWR with their 95% CI and p-values
# ind: index (logical) for rows in the main 'data'
# ind1: index (logical) for rows in 'data.H1'
# r: number of decimals for rounding in the output
gwr.fun = function(ind, ind1, r = 2) {

  # fit NWR, FWR, and LWR to original data (multiple events)
  obj <- WRrec(
    ID = data$patid[ind],
    time = data$time[ind],
    status = data$status[ind],
    trt = data$trt_ab[ind],
    strata = data$age60[ind],
    naive = TRUE
  )

  # fit sWR (PWR) to dataset with first hospitalization only
  # This typically addresses "semi-competing" event structure
  obj1 <- WRrec(
    ID = data.H1$patid[ind1],
    time = data.H1$time[ind1],
    status = data.H1$status[ind1],
```

```

    trt = data.H1$trt_ab[ind1],
    strata = data.H1$age60[ind1],
    naive = FALSE
  )

# critical value for a 95% confidence interval
za <- qnorm(0.975)

## LWR results
beta <- obj$log.WR
se <- obj$se
theta <- obj$theta # proportion of pairwise comparisons that are wins/losses

# Format: percentage of wins, percentage of losses,
#         win ratio & 95% CI, p-value
r4 <- c(
  paste0(round(100 * theta[1], 1), "%"), # Win
  paste0(round(100 * theta[2], 1), "%"), # Loss
  paste0(
    round(exp(beta), r), " (",
    round(exp(beta - za * se), r), ", ",
    round(exp(beta + za * se), r), ")"
  ),
  round(1 - pchisq((beta / se)^2, 1), 3)
)

## PWR results
beta1 <- obj1$log.WR
se1 <- obj1$se
theta1 <- obj1$theta

r1 <- c(
  paste0(round(100 * theta1[1], 1), "%"),
  paste0(round(100 * theta1[2], 1), "%"),
  paste0(
    round(exp(beta1), r), " (",
    round(exp(beta1 - za * se1), r), ", ",
    round(exp(beta1 + za * se1), r), ")"
  ),
  round(1 - pchisq((beta1 / se1)^2, 1), 3)
)

```

```

## NWR results
beta.naive <- obj$log.WR.naive
se.naive <- obj$se.naive
theta.naive <- obj$theta.naive

r2 <- c(
  paste0(round(100 * theta.naive[1], 1), "%"),
  paste0(round(100 * theta.naive[2], 1), "%"),
  paste0(
    round(exp(beta.naive), r), " (",
    round(exp(beta.naive - za * se.naive), r), ", ",
    round(exp(beta.naive + za * se.naive), r), ")"
  ),
  round(1 - pchisq((beta.naive / se.naive)^2, 1), 3)
)

## FWR results
beta.FI <- obj$log.WR.FI
se.FI <- obj$se.FI
theta.FI <- obj$theta.FI

r3 <- c(
  paste0(round(100 * theta.FI[1], 1), "%"),
  paste0(round(100 * theta.FI[2], 1), "%"),
  paste0(
    round(exp(beta.FI), r), " (",
    round(exp(beta.FI - za * se.FI), r), ", ",
    round(exp(beta.FI + za * se.FI), r), ")"
  ),
  round(1 - pchisq((beta.FI / se.FI)^2, 1), 3)
)

# Combine rows into a single table
result <- rbind(r1, r2, r3, r4)
rownames(result) <- c("PWR", "NWR", "FWR", "LWR")

return(result)
}

# Create table
## Age <= 60 years
ind <- (data$age60 == 0)

```

```

ind1 <- (data.H1$age60 == 0)
result.lt60 <- gwr.fun(ind, ind1, r = 2)

## Age > 60 years
ind <- (data$age60 == 1)
ind1 <- (data.H1$age60 == 1)
result.ge60 <- gwr.fun(ind, ind1, r = 2)

## overall
ind <- rep(TRUE, nrow(data))
ind1 <- rep(TRUE, nrow(data.H1))
result.all <- gwr.fun(ind, ind1, r = 2)

# combine results
results <- rbind(result.lt60, result.ge60, result.all)
colnames(results) <- c("Win", "Loss", "Win ratio (95% CI)", "p-value")
noquote(results)
#> Final table of all 4 measures (PWR, NWR, FWR, LWR) across strata.

#####
#                               Sample size calculation                               #
#####

# get training arm data
pilot <- hfaction |>
  filter(trt_ab == 1)
# number of subjects
pilot |>
  distinct(patid) |>
  count()
#> This indicates how many unique subjects were in the training arm

##### estimate parameters #####
# Get the variables from pilot dataset
# to estimate baseline parameters
# lambda_D, lambda_H, kappa

outcome_base <- gumbel.est(pilot$patid, pilot$time / 12, pilot$status)

lambda_D <- outcome_base$lambda_D
lambda_H <- outcome_base$lambda_H
kappa <- outcome_base$kappa

```

```

lambda_D
lambda_H
kappa
#> Baseline hazards for death/hospitalization and the gumbel 'kappa' parameter

## Kendall's rank correlation
1 - 1/kappa
#> [1] 0.360812
#> This measures correlation between timing of repeated events

### demo #####
# set design parameters
tau_b <- 3 # time from baseline to start of follow-up for base() computation
tau <- 4 # total follow-up (in years)
lambda_L <- 0.01 # Additional parameter used in the base() function

# use base() function to compute zeta2 and delta
## may take up to 30s
bparam <- base(lambda_D, lambda_H, kappa, tau_b, tau, lambda_L)
#> bparam includes the baseline rates and distribution shape
#> used for sample size calculations

# compute sample size under HRs 0.8 and 0.9
# for death and nonfatal event, respectively
obj <- WRSS(
  xi = log(c(0.9, 0.8)),
  bparam = bparam,
  q = 0.5,
  alpha = 0.05,
  power = 0.8
)
obj$n
#> The required sample size for the given HRs at 80% power

## effect size specification
thetaD <- seq(0.6, 0.95, by = 0.05) # hazard ratio for death
thetaH <- seq(0.6, 0.95, by = 0.05) # hazard ratio for hospitalization

## create a matrix "SS08" for sample size powered at 80%
## under each combination of thetaD and thetaH
mD <- length(thetaD)
mH <- length(thetaH)

```

```

SS08 <- matrix(NA, mD, mH)
rownames(SS08) <- thetaD
colnames(SS08) <- thetaH

## fill in the computed sample size values
for (i in 1:mD) {
  for (j in 1:mH) {
    ## sample size under hazard ratios thetaD[i] for death
    ## and thetaH[j] for hospitalization
    SS08[i, j] <- WRSS(
      xi = log(c(thetaD[i], thetaH[j])),
      bparam = bparam,
      q = 0.5,
      alpha = 0.05,
      power = 0.8
    )$n
  }
}
## print the calculated sample sizes
print(SS08)
#> Shows how sample size changes under different hazard ratios for death/hosp

## repeating the same calculation for power = 90%
SS09 <- matrix(NA, mD, mH)
rownames(SS09) <- thetaD
colnames(SS09) <- thetaH # As in original code; sets the colnames to the sequence of thetaH

## fill in the computed sample size values
for (i in 1:mD) {
  for (j in 1:mH) {
    ## sample size under hazard ratios thetaD[i] for death
    ## and thetaH[j] for hospitalization
    SS09[i, j] <- WRSS(
      xi = log(c(thetaD[i], thetaH[j])),
      bparam = bparam,
      q = 0.5,
      alpha = 0.05,
      power = 0.9
    )$n
  }
}
## print the calculated sample sizes

```



```

print(SS09)
#> Sample sizes under 90% power requirements

oldpar <- par(mfrow = par("mfrow"))
par(mfrow = c(1, 2))

persp(
  thetaD, thetaH, SS08 / 1000,
  theta = 50, phi = 15, expand = 0.8, col = "gray",
  ltheta = 180, lphi = 180, shade = 0.75,
  ticktype = "detailed",
  xlab = "\n HR on Death", ylab = "\n HR on Hospitalization",
  zlab = paste0("\n Sample Size (10e3)",
  main = "Power = 80%",
  zlim = c(0, 26)
)
#> 3D perspective plot of sample size (in thousands) for power=80%
#> over various hazard ratios for death/hosp

persp(
  thetaD, thetaH, SS09 / 1000,
  theta = 50, phi = 15, expand = 0.8, col = "gray",
  ltheta = 180, lphi = 180, shade = 0.75,
  ticktype = "detailed",
  xlab = "\nHR on Death", ylab = "\nHR on Hospitalization",
  zlab = paste0("\n Sample Size (10e3)",
  main = "Power = 90%",
  zlim = c(0, 26)
)
#> Similar 3D perspective for power=90%

```

## Win Ratio: Definitions and Properties

The win ratio (WR) is a pairwise, nonparametric method for comparing composite outcomes in a prioritized manner. Each subject in the treatment arm is compared to each subject in the control arm over a shared follow-up window. A win is declared if the treated subject experiences a better outcome—typically defined as longer survival or, if tied on survival, fewer or later nonfatal events.