Alternative Study Designs and Nonparametric Statistical Methods for Adaptive Management Studies of Invasive Plants

James N. McNair, Daniel Frobish, Emma K. Rice, and Ryan A. Thum

Supplementary Appendix S2: R Programs

This document presents a set of complete R programs intended to serve as examples that illustrate how various R statistical functions can be used to answer the research questions addressed in the main text. Output produced by each program is also included. Programs are identified by the research questions they address. Because the purpose of the programs is didactic, they are not intended to be used in their current form with other data sets and are presented without warranty of any kind.

The programs are written in an explicit style and include numerous comments, with the intent of making the underlying logic clear and obvious. User-defined functions are largely avoided, and loop-avoidance functions are avoided entirely. The programs are intended to be run from the R console, using the `source()` command. For users that prefer the informal approach of performing all statistical analyses interactively, the required code fragments can be extracted from the programs that are provided. In either case, make sure that required add-on packages are installed; these are listed near the beginning of each program, in a section titled "LOAD AND ATTACH PACKAGE(S)".

We recommend developing complete programs that can be run with the `source()` command. R's interpreter and superb facilities for exploring properties of data should be used interactively to develop the programs for a particular project. Once all programs are fully developed and debugged, their output can be used to prepare project reports and publications. The finalized programs and input data files should be archived with other project-related materials and can be used to quickly reproduce all reported results for a quality assurance officer during an audit.

1 Management Experiments Using Marked Plants

This section includes listings of example R programs that perform the nonparametric statistical procedures presented in section 3 of the main text. Required data are included in the programs; no external data files are used.

1.1 Assessing the efficacy of a single management treatment on a single species

Question 1: Program listing

```
# Title:    Question_01.R
# Purpose:  Compare success probabilities of three Norway maple sapling removal
#           methods in preventing resprouting using marked stumps.
# Command to run: source("Question_01.R")
```

```
## LOAD AND ATTACH PACKAGE
library(binom)

## ENTER DATA
# Management practice C (Cut trunk)
n.C <- 60
n.C.regrow <- 48
n.C.notregrow <- n.C - n.C.regrow
# Management practice T (cut trunk and apply Triclopyr to stump)
n.T <- 60
n.T.regrow <- 2
n.T.notregrow <- n.T - n.T.regrow

## ESTIMATE CONFIDENCE INTERVALS
# Management practice C
ci.out.C <- binom.confint(n.C.notregrow, n.C, methods=c("ac", "asymptotic", "wilson"))
cat("\nEstimates of PET and 95% confidence interval for treatment Cut:\n\n")
print(ci.out.C)
# Management practice T
ci.out.T <- binom.confint(n.T.notregrow, n.T, methods=c("ac", "asymptotic", "wilson"))
cat("\nEstimates of PET and 95% confidence interval for treatment Cut + Triclopyr:\n\n")
print(ci.out.T)
cat("\n")
```

Question 1: Output

```
> source("Question_01.R")

Estimates of PET and 95% confidence interval for treatment Cut:

         method  x  n mean     lower     upper
1 agresti-coull 12 60  0.2 0.1167618 0.3193413
2    asymptotic 12 60  0.2 0.0987879 0.3012121
3        wilson 12 60  0.2 0.1182851 0.3178181

Estimates of PET and 95% confidence interval for treatment Cut + Triclopyr:

         method  x  n      mean     lower     upper
1 agresti-coull 58 60 0.9666667 0.8796932 0.9974798
2    asymptotic 58 60 0.9666667 0.9212463 1.0120870
3        wilson 58 60 0.9666667 0.8863623 0.9908107
```

Question 2: Program listing

```
# Title:    Question_02.R
# Purpose:  Assess ability of two Norway maple sapling removal methods to achieve PET > p*.
# Command to run: source("Question_02.R")

## ENTER DATA
# Management practice C (Cut trunk)
n.C <- 60
n.C.regrow <- 48
n.C.notregrow <- n.C - n.C.regrow
# Management practice T (cut trunk and apply Triclopyr to stump)
n.T <- 60
n.T.regrow <- 2
n.T.notregrow <- n.T - n.T.regrow
# Management threshold (target)
p.star <- 0.8

## HYPOTHESIS TESTS
# Perform large-sample tests using prop.test()
cat("\nLarge-sample tests using prop.test() ", rep("-", 41), "\n\n", sep="")
cat("   H0: PET = p*,  H1: PET > p*\n\n")
# ... One-sided H1: p.C - p.star > 0
pt.out.greater.C <- prop.test(n.C.notregrow, n.C, p=0.8, correct=F, alternative="greater")
cat("Treatment: Cut\n")
print(pt.out.greater.C)
# ... One-sided H1: p.T - p.star > 0
cat("Treatment: Cut + Triclopyr\n")
pt.out.greater.T <- prop.test(n.T.notregrow, n.T, p=0.8, correct=F, alternative="greater")
print(pt.out.greater.T)
# Perform exact binomial tests using binom.test()
cat("\nExact binomial tests using binom.test() ", rep("-", 38), "\n\n", sep="")
cat("   H0: PET = p*,  H1: PET > p*\n\n")
# ... One-sided H1: p.C - p.star > 0
bt.out.greater.C <- binom.test(n.C.notregrow, n.C, p=0.8, alternative="greater")
cat("Treatment: Cut\n")
print(bt.out.greater.C)
# ... One-sided H1: p.T - p.star > 0
bt.out.greater.T <- binom.test(n.T.notregrow, n.T, p=0.8, alternative="greater")
cat("Treatment: Cut + Triclopyr\n")
print(bt.out.greater.T)
# Perform mid-P binomial tests using binom.test()
cat("\nExact mid-P binomial tests using pbinom() ", rep("-", 36), "\n\n", sep="")
cat("   H0: PET = p*,  H1: PET > p*\n\n")
# ... One-sided H1: p.C - p.star > 0
raw.p.mpbn.C <- pbinom(n.C.notregrow - 1, n.C, 0.8, lower.tail=F) - 0.5*dbinom(n.C.notregrow,
   n.C, 0.8)
cat("Treatment: Cut\n\n")
cat("Alternative hypothesis: True probability of success is greater than", p.star, "\n")
cat("   p-value =", raw.p.mpbn.C, "\n")
# ... One-sided H1: p.T - p.star > 0
raw.p.mpbn.T <- pbinom(n.T.notregrow - 1, n.T, 0.8, lower.tail=F) - 0.5*dbinom(n.T.notregrow,
   n.T, 0.8)
cat("\nTreatment: Cut + Triclopyr\n\n")
cat("Alternative hypothesis: True probability of success is greater than", p.star, "\n")
cat("   p-value =", raw.p.mpbn.T, "\n")
```

Question 2: Output

```
> source("Question_02.R")

Large-sample tests using prop.test() ----------------------------------------

   H0: PET = p*,  H1: PET > p*

Treatment: Cut

        1-sample proportions test without continuity correction

data:  n.C.notregrow out of n.C, null probability 0.8
X-squared = 135, df = 1, p-value = 1
alternative hypothesis: true p is greater than 0.8
95 percent confidence interval:
 0.1288546 1.0000000
sample estimates:
  p
0.2

Treatment: Cut + Triclopyr

        1-sample proportions test without continuity correction

data:  n.T.notregrow out of n.T, null probability 0.8
X-squared = 10.417, df = 1, p-value = 0.0006244
alternative hypothesis: true p is greater than 0.8
95 percent confidence interval:
 0.9041557 1.0000000
sample estimates:
        p
0.9666667


Exact binomial tests using binom.test() -------------------------------------

   H0: PET = p*,  H1: PET > p*

Treatment: Cut

        Exact binomial test

data:  n.C.notregrow and n.C
number of successes = 12, number of trials = 60, p-value = 1
alternative hypothesis: true probability of success is greater than 0.8
95 percent confidence interval:
 0.1196561 1.0000000
sample estimates:
probability of success
                  0.2

Treatment: Cut + Triclopyr

        Exact binomial test

data:  n.T.notregrow and n.T
number of successes = 58, number of trials = 60, p-value = 0.0001941
alternative hypothesis: true probability of success is greater than 0.8
95 percent confidence interval:
 0.8987641 1.0000000
sample estimates:
probability of success
```

```
             0.9666667


Exact mid-P binomial tests using pbinom() ----------------------------------

   H0: PET = p*,  H1: PET > p*

Treatment: Cut

Alternative hypothesis: True probability of success is greater than 0.8
   p-value = 1

Treatment: Cut + Triclopyr

Alternative hypothesis: True probability of success is greater than 0.8
   p-value = 0.0001092861
```

1.2 Comparing treatment efficacy in pairs of treatments or species

Question 3: Program listing

```
# Title:    Question_03.R
# Purpose:  Compare success of two Norway maple sapling removal methods (cut down, Cut down
#           then apply triclopyr to the stump) in preventing resprouting from marked stumps.
# Command to run: source("Question_03.R")

## LOAD AND ATTACH PACKAGE
library(Exact)

## CREATE DATA MATRIX
# Method C (Cut trunk)
n.C.notregrow <- 12
n.C.regrow    <- 48
n.C <- n.C.notregrow + n.C.regrow
# Method T (cut trunk and apply Triclopyr to stump)
n.T.notregrow <- 58
n.T.regrow    <-  2
n.T <- n.T.notregrow + n.T.regrow
# Set up N.mat matrix (treatment "success" = NOT regrow)
#     Row 1: Test area, Row 2: Reference area
#     Col 1: treatment successes, Col 2: treatment failures
N.mat <- matrix(c(n.T.notregrow, n.T.regrow, n.C.notregrow, n.C.regrow), ncol=2, byrow=T)

## HYPOTHESIS TESTS
# Perform large-sample test using prop.test()
# Large-sample test H0: p.CT = p.C versus H1: p.CT > p.C using prop.test()
pt.out.greater <- prop.test(x=N.mat, correct=F, alternative="greater")
cat("\nLarge-sample test using prop.test() ", rep("-", 42), "\n\n", sep="")
cat("   H0: p.CT = p.C\n")
cat("   H1: p.CT > p.C\n")
print(pt.out.greater)
# Perform exact unconditional test using exact.test() with method="csm"
#   (This is Barnard's original method)
cat("\nBarnard's test using exact.test() with method='csm' ", rep("-", 26), "\n\n", sep="")
cat("   H0: p.CT = p.C\n")
cat("   H1: p.CT > p.C\n")
et.out.greater <- exact.test(data=N.mat, alternative="greater", method="csm", to.plot=F)
print(et.out.greater)
```

Question 3: Output

```
> source("Question_03.R")

Large-sample test using prop.test() ------------------------------------------

   H0: p.CT = p.C
   H1: p.CT > p.C

        2-sample test for equality of proportions without continuity correction

data:  N.mat
X-squared = 72.549, df = 1, p-value < 2.2e-16
alternative hypothesis: greater
95 percent confidence interval:
 0.6735659 1.0000000
sample estimates:
   prop 1    prop 2
0.9666667 0.2000000


Barnard's test using exact.test() with method='csm' --------------------------

   H0: p.CT = p.C
   H1: p.CT > p.C

        CSM Exact Test

data:  58 out of 60 vs. 12 out of 60
test statistic = NA, first sample size = 60, second sample size = 60,
p-value < 2.2e-16
alternative hypothesis: true difference in proportion is greater than 0
sample estimates:
difference in proportion
               0.7666667
```

Question 4: Program listing

```
# Title:   Question_04.R
# Purpose: Compare success of three Norway maple sapling removal methods in preventing
#          resprouting using marked stumps.
# Command to run: source("Question_04.R")

## LOAD AND ATTACH PACKAGES
library(pairwiseCI)
library(ExactCIdiff)

## CREATE DATA FRAME
method  <- c("CT", "C")
success <- c(58, 12)
failure <- c(2, 48)
data.df <- data.frame(removal.method=method, not.resprouted=success, resprouted=failure)

## NEWCOMBE HYBRID SCORE INTERVAL
pci <- pairwiseCI(cbind(not.resprouted, resprouted) ~ removal.method, data=data.df,
   method="Prop.diff", CImethod="NHS")
pci.sum <- summary(pci)
pci.table <- cbind(pci.sum$estimate, pci.sum $conf.int)
rownames(pci.table) <- "p_CT - p_C"
cat("\n")
cat("95% confidence interval:", pci$byout[[1]]$method, "\n\n")
print(pci.table)
cat("\n")

## AGRESTI-CAFFO INTERVAL
pci <- pairwiseCI(cbind(not.resprouted, resprouted) ~ removal.method, data=data.df,
   method="Prop.diff", CImethod="AC")
pci.sum <- summary(pci)
pci.table <- cbind(pci.sum$estimate, pci.sum$conf.int)
rownames(pci.table) <- "p_CT - p_C"
cat("\n")
cat("95% confidence interval:", pci$byout[[1]]$method, "\n\n")
print(pci.table)
cat("\n")

## WANG EXACT UNCONDITIONAL INTERVAL
CT.tot  <- sum(data.df[1, 2:3])
C.tot   <- sum(data.df[2, 2:3])
CT.succ <- data.df[1, 2] # successes (not.resprouted)
C.succ  <- data.df[2, 2] # failure   (resprouted)
cat("\n...Computing exact unconditional confidence interval. This will take a while...")
cat("\n")
xcidiff.out <- BinomCI(CT.tot, C.tot, CT.succ, C.succ, CItype="Two.sided")
cat("\n\n95% confidence interval: Wang interval for difference of proportions\n\n")
cat("           estimate    lower    upper\n")
str.1 <- formatC(xcidiff.out$estimate, digits=4, width=8, format="f")
str.2 <- formatC(xcidiff.out$ExactCI[1], digits=4, width=8, format="f")
str.3 <- formatC(xcidiff.out$ExactCI[2], digits=4, width=8, format="f")
cat("p_CT - p_C", str.1, str.2, str.3, "\n\n")
```

Question 4: Output

```
> source("Question_04.R")
Loading required package: MCPAN
Loading required package: coin
Loading required package: survival

95% confidence interval: Newcombes Hybrid Score interval for the difference of proportions

           estimate  lower  upper
p_CT - p_C   0.7667 0.6241 0.8519


95% confidence interval: Agresti-Caffo interval for the difference of proportions

           estimate  lower  upper
p_CT - p_C   0.7667 0.6274 0.8565


...Computing exact unconditional confidence interval. This will take a while...


95% confidence interval: Wang interval for difference of proportions

           estimate    lower    upper
p_CT - p_C   0.7667   0.6350   0.8597
```

2 Management Experiments Using Point Intercept Surveys

This section includes listings of example R programs that perform the nonparametric statistical procedures presented in section 4 of the main text, plus an extra R program that creates two sets of simulated data required by the statistical programs. Each statistical program is identified by the research question it addresses. The programs are intended to be run from the R console, using the `source()` command. Make sure required packages are installed. We recommend placing all programs (as plain text files) in a single directory, then making that the working directory after starting R. The data simulator program should be run first, since the data files it creates are required by the statistical programs.

Point intercept data simulator: Program listing

```
# Title:    PI_data_simulator.R
# Purpose:  Creates two data files containing simulated binary and quantitative point
#           intercept data.
# Command to run: source("PI_data_simulator.R")

## User-defined function ------------------------------------------------------------------
rbern <- function(n, p) {
   # Create n pseudo-random Bernoulli variates with success probability p
   return( rbinom(n=n, size=1, prob=p) )
}
## --------------------------------------------------------------------------------------

## PARAMETER VALUES
# Number of survey points in each sampling site; must be a multiple of 10 (= # of columns)
n <- 60
if(n %% 10 != 0)
   stop("*** Number n of survey points must be divisible by 10 ***\n")
n.cols <- 10
n.rows <- n/10
# Various event probabilities
p.present.test <- 0.5
p.present.ref  <- 0.5
p.mort.test    <- 0.98
p.mort.ref     <- 0.1
p.repro.test   <- 0.0
p.repro.ref    <- 0.5
p.immig.test   <- 0.1
p.immig.ref    <- 0.1
# Pseud-random number generator seed
set.seed(3)

## CREATE PRE-TREATMENT DATA
# Test site
I.present.test <- rbern(n, p=p.present.test)
N.pre.test     <- I.present.test*rbinom(n, size=50, prob=0.3)
# Reference site
I.present.ref  <- rbern(n, p=p.present.ref)
N.pre.ref      <- I.present.ref*rbinom(n, size=50, prob=0.3)

## CREATE POST-TREATMENT DATA
I.immig.test <- rbern(n, p=p.immig.test)
n.immig.test <- rbinom(n, size=2, prob=0.5)
I.immig.ref  <- rbern(n, p=p.immig.ref)
n.immig.ref  <- rbinom(n, size=2, prob=0.5)
N.post.test  <- rep(NA, times=n)
N.post.ref   <- rep(NA, times=n)
for(i in 1:n) {
```

```
   # Test site
   I.mort.test    <- rbern(N.pre.test[i], p=p.mort.test)
   I.repro.test   <- rbern(N.pre.test[i], p=p.repro.test)
   n.repro.test   <- rbinom(N.pre.test[i], size=2, prob=0.25)
   N.post.test[i] <- sum((1 - I.mort.test)*(1 + I.repro.test*n.repro.test)) +
      I.immig.test[i]*n.immig.test[i]
   # Reference site
   I.mort.ref     <- rbern(N.pre.ref[i], p=p.mort.ref)
   I.repro.ref    <- rbern(N.pre.ref[i], p=p.repro.ref)
   n.repro.ref    <- rbinom(N.pre.ref[i], size=2, prob=0.25)
   N.post.ref[i]  <- sum((1 - I.mort.ref)*(1 + I.repro.ref*n.repro.ref)) +
      I.immig.ref[i]*n.immig.ref[i]
}
PA.pre.test  <- ifelse(N.pre.test > 0, 1, 0)
PA.post.test <- ifelse(N.post.test > 0, 1, 0)
PA.pre.ref   <- ifelse(N.pre.ref > 0, 1, 0)
PA.post.ref  <- ifelse(N.post.ref > 0, 1, 0)

## CREATE X AND Y SPATIAL COORDINATES FOR THE N SURVEY POINTS
## Note: These coordinates are included to show the recommended structure of the data
##       sets; they are not used in the example statistical programs we have provided.
# Survey point coordinates
x.coords <- seq(0, ((n.cols - 1)*50), by=50)
y.coords <- seq(0, ((n.rows - 1)*50), by=50)
# Re-label by row (e.g., north to south) and column (e.g., east to west)
x <- rep(NA, times=(n.rows*n.cols))
y <- rep(NA, times=(n.rows*n.cols))
for(row in 1:n.rows) {
   for (col in 1:n.cols) {
      i    <- (row - 1)*n.cols + col
      y[i] <- y.coords[row]
      x[i] <- x.coords[col]
   }
}

## CREATE DATA FRAMES AND SAVE
# Quantitative data
data.df.test <- data.frame(site="Test", y=y, x=x, pre.abund=N.pre.test, post.abund=N.post.test)
data.df.ref  <- data.frame(site="Ref",  y=y, x=x, pre.abund=N.pre.ref,  post.abund=N.post.ref)
data.df <- rbind(data.df.test, data.df.ref)
write.table(data.df, file="Faux_quantitative_data.csv", row.names=F, sep=",")
# Binary data
data.df.test <- data.frame(site="Test", y=y, x=x, pre.PA=PA.pre.test, post.PA=PA.post.test)
data.df.ref  <- data.frame(site="Ref",  y=y, x=x, pre.PA=PA.pre.ref, post.PA=PA.post.ref)
data.df <- rbind(data.df.test, data.df.ref)
write.table(data.df, file="Faux_binary_data.csv", row.names=F, sep=",")
# Message to user
cat("\n")
cat("*** Faux data saved as CSV files in current working directory ***\n\n")
cat("    Binary data:       Faux_binary_data.csv\n")
cat("    Quantitative data: Faux_quantitative_data.csv\n\n")
```

Point intercept data simulator: Output

```
> source("PI_data_simulator.R")

*** Faux data saved as CSV files in current working directory ***

    Binary data:       Faux_binary_data.csv
    Quantitative data: Faux_quantitative_data.csv
```

2.1 Assessing the efficacy of a single management treatment on a single population

2.1.1 Methods based on binary data

Question 7: Program listing

```
# Title:    Question_07.R
# Purpose:  Estimate PDS change and 95% confidence interval for Test and Ref sites.
# Command to run: source("Question_07.R")

## LOAD AND ATTACH PACKAGE
library(PropCIs)

## READ IN DATA
data.df <- read.table("Faux_binary_data.csv", header=T, sep=",")

## TEST SITE
# Determine Nij and n for pre and post treatment
Test.df <- subset(data.df, site == "Test")
pre.PA  <- Test.df$pre.PA
post.PA <- Test.df$post.PA
N11 <- sum((pre.PA == 0) & (post.PA == 0))
N12 <- sum((pre.PA == 0) & (post.PA == 1))
N21 <- sum((pre.PA == 1) & (post.PA == 0))
N22 <- sum((pre.PA == 1) & (post.PA == 1))
n <- N11 + N12 + N21 + N22
# Agresti-Min CI for P1' - P1
diffprop.out <- diffpropci.mp(N12, N21, n, 0.95)
cat("\nP1' - P1 for Test site, pre v. post:\n\n")
cat("   Estimate of P1' - P1 as difference of ML estimates:", (N21 - N12)/n, "\n")
cat("   Estimate of P1' - P1  as midpoint of conf. int.:   ", diffprop.out$estimate, "\n")
cat("   95% confidence interval:", diffprop.out$conf.int, "\n\n")

## REF SITE
# Determine Nij and n for pre and post treatment
Ref.df <- subset(data.df, site == "Ref")
pre.PA  <- Ref.df$pre.PA
post.PA <- Ref.df$post.PA
N11 <- sum((pre.PA == 0) & (post.PA == 0))
N12 <- sum((pre.PA == 0) & (post.PA == 1))
N21 <- sum((pre.PA == 1) & (post.PA == 0))
N22 <- sum((pre.PA == 1) & (post.PA == 1))
n <- N11 + N12 + N21 + N22
# Agresti-Min CI for P1' - P1
diffprop.out <- diffpropci.mp(N12, N21, n, 0.95)
cat("\nP1' - P1 for Ref site, pre v. post:\n\n")
cat("   Estimate of P1' - P1 as difference of ML estimates:", (N21 - N12)/n, "\n")
cat("   Estimate of P1' - P1  as midpoint of conf. int.:   ", diffprop.out$estimate, "\n")
cat("   95% confidence interval:", diffprop.out$conf.int, "\n\n")
```

Question 7: Output

```
> source("Question_07.R")

P1' - P1 for Test site, pre v. post:

   Estimate of P1' - P1 as difference of ML estimates: 0.2666667
   Estimate of P1' - P1  as midpoint of conf. int.:    0.2580645
   95% confidence interval: 0.1136444 0.4024846


P1' - P1 for Ref site, pre v. post:

   Estimate of P1' - P1 as difference of ML estimates: -0.05
   Estimate of P1' - P1  as midpoint of conf. int.:    -0.0483871
   95% confidence interval: -0.1104539 0.01367973
```

Question 8: Program listing

```
# Title:    Question_08.R
# Purpose:  Test H0: P1' - P1 = 0 v. H1: P1' - P1 > 0
# Command to run: source("Question_08.R")

## READ IN DATA
data.df <- read.table("Faux_binary_data.csv", header=T, sep=",")

## Mid-P McNemar test ----------------------------------------------------------------------
cat("\nmid-P McNemar test ", rep("=", 70), "\n\n", sep="")

## TEST SITE
# Determine Nij and n for pre and post treatment
Test.df <- subset(data.df, site == "Test")
pre.PA  <- Test.df$pre.PA
post.PA <- Test.df$post.PA
N11 <- sum((pre.PA == 0) & (post.PA == 0))
N12 <- sum((pre.PA == 0) & (post.PA == 1))
N21 <- sum((pre.PA == 1) & (post.PA == 0))
N22 <- sum((pre.PA == 1) & (post.PA == 1))
n.star <- N12 + N21
cat("Test site:\n\n")
cat("N11 =", N11, "\tN12 =", N12, "\n")
cat("N21 =", N21, "\tN22 =", N22,  "\n\n")
cat("n* =", n.star, "\n\n")
# Perform mid-p McNemar test
P.val <- pbinom(N21 - 1, n.star, 0.5, lower.tail=F) - 0.5*dbinom(N12, n.star, 0.5)
cat("mid-P P-value for test of H0: P1'-P1 = 0 v. H1: P1'-P1 > 0 for Test site:\n")
cat("   P =", P.val, "\n\n")

## REFERENCE SITE
# Determine Nij and n for pre and post treatment
Ref.df  <- subset(data.df, site == "Ref")
pre.PA  <- Ref.df$pre.PA
post.PA <- Ref.df$post.PA
N11 <- sum((pre.PA == 0) & (post.PA == 0))
N12 <- sum((pre.PA == 0) & (post.PA == 1))
N21 <- sum((pre.PA == 1) & (post.PA == 0))
N22 <- sum((pre.PA == 1) & (post.PA == 1))
n.star <- N12 + N21
cat("Reference site:\n\n")
cat("N11 =", N11, "\tN12 =", N12, "\n")
cat("N21 =", N21, "\tN22 =", N22,  "\n\n")
cat("n* =", n.star, "\n\n")
# Perform mid-p McNemar test
P.val <- pbinom(N21 - 1, n.star, 0.5, lower.tail=F) - 0.5*dbinom(N12, n.star, 0.5)
cat("mid-P P-value for test of H0: P1'-P1 = 0 v. H1: P1'-P1 > 0 for Ref site:\n")
cat("   P =", P.val, "\n")

## Exact conditional McNemar test ----------------------------------------------------------
cat("\nExact conditional McNemar test ", rep("=", 58), "\n\n", sep="")

## TEST SITE
# Determine Nij and n for pre and post treatment
Test.df <- subset(data.df, site == "Test")
pre.PA  <- Test.df$pre.PA
post.PA <- Test.df$post.PA
N11 <- sum((pre.PA == 0) & (post.PA == 0))
N12 <- sum((pre.PA == 0) & (post.PA == 1))
N21 <- sum((pre.PA == 1) & (post.PA == 0))
N22 <- sum((pre.PA == 1) & (post.PA == 1))
n.star <- N12 + N21
```

```
# Perform exact conditional McNemar test
P.val <- pbinom(N21 - 1, n.star, 0.5, lower.tail=F)
cat("Test site:\n\n")
cat("Exact conditional P-value for test of H0: P1'-P1 = 0 v. H1: P1'-P1 > 0 for Test site:\n")
cat("   P =", P.val, "\n\n")

## REFERENCE SITE
# Determine Nij and n for pre and post treatment
Ref.df  <- subset(data.df, site == "Ref")
pre.PA  <- Ref.df$pre.PA
post.PA <- Ref.df$post.PA
N11 <- sum((pre.PA == 0) & (post.PA == 0))
N12 <- sum((pre.PA == 0) & (post.PA == 1))
N21 <- sum((pre.PA == 1) & (post.PA == 0))
N22 <- sum((pre.PA == 1) & (post.PA == 1))
n.star <- N12 + N21
# Perform exact conditional McNemar test
P.val <- pbinom(N21 - 1, n.star, 0.5, lower.tail=F)
cat("Reference site:\n\n")
cat("Exact conditional P-value for test of H0: P1'-P1 = 0 v. H1: P1'-P1 > 0 for Ref site:\n")
cat("   P =", P.val, "\n\n")
```

Question 8: Output

```
> source("Question_08.R")

mid-P McNemar test ================================================================

Test site:

N11 = 29        N12 = 4
N21 = 20        N22 = 7

n* = 24

mid-P P-value for test of H0: P1'-P1 = 0 v. H1: P1'-P1 > 0 for Test site:
   P = 0.0004552603

Reference site:

N11 = 24        N12 = 3
N21 = 0         N22 = 33

n* = 3

mid-P P-value for test of H0: P1'-P1 = 0 v. H1: P1'-P1 > 0 for Ref site:
   P = 0.9375

Exact conditional McNemar test ====================================================

Test site:

Exact conditional P-value for test of H0: P1'-P1 = 0 v. H1: P1'-P1 > 0 for Test site:
   P = 0.0007719398

Reference site:

Exact conditional P-value for test of H0: P1'-P1 = 0 v. H1: P1'-P1 > 0 for Ref site:
   P = 1
```

Question 8a: Program listing

```
# Title:    Question_08a.R
# Purpose:  Estimate p_2|1 and p_1|2 and their 95% confidence intervals
# Command to run: source("Question_09.R")

## LOAD AND ATTACH PACKAGE
library(binom)

## READ IN DATA
data.df <- read.table("Faux_binary_data.csv", header=T, sep=",")

## TEST SITE
# Determine Nij and n for pre and post treatment
Test.df <- subset(data.df, site == "Test")
pre.PA  <- Test.df$pre.PA
post.PA <- Test.df$post.PA
N11 <- sum((pre.PA == 0) & (post.PA == 0))
N12 <- sum((pre.PA == 0) & (post.PA == 1))
N21 <- sum((pre.PA == 1) & (post.PA == 0))
N22 <- sum((pre.PA == 1) & (post.PA == 1))
N1.pre <- N11 + N12
N2.pre <- N21 + N22
n  <- N1.pre  + N2.pre
# Maximum likelihood estimates of p_2|1 and p_1|2
p.hat_2given1 <- N12/N1.pre
p.hat_1given2 <- N21/N2.pre
# Wilson and Agresti-Coull confidence intervals
cat("\nTest site ", rep("=", 70), "\n\n", sep="")
ci.p1given2.out <- binom.confint(N21, N2.pre, methods=c("ac", "wilson"))
cat("95% confidence intervals for p_{1|2}:\n\n")
cat("          Method   N21 N2.pre   mean    lower    upper\n")
cat("   ---------------------------------------------------\n")
cat("          Wilson    ", formatC(N21, format="d", width=2),
   formatC(N2.pre, format="d", width=5),
   formatC(ci.p1given2.out$mean[2], format="f", digits=5, width=9),
   formatC(ci.p1given2.out$lower[2], format="f", digits=5, width=9),
   formatC(ci.p1given2.out$upper[2], format="f", digits=5, width=9), sep="", "\n")
cat("   Agresti-Coull    ", formatC(N21, format="d", width=2),
   formatC(N2.pre, format="d", width=5),
   formatC(ci.p1given2.out$mean[1], format="f", digits=5, width=9),
   formatC(ci.p1given2.out$lower[1], format="f", digits=5, width=9),
   formatC(ci.p1given2.out$upper[1], format="f", digits=5, width=9), sep="", "\n\n")
ci.p2given1.out <- binom.confint(N12, N1.pre, methods=c("ac", "wilson"))
cat("95% confidence intervals for p_{2|1}:\n\n")
cat("          Method   N12 N1.pre   mean    lower    upper\n")
cat("   ---------------------------------------------------\n")
cat("          Wilson    ", formatC(N12, format="d", width=2),
   formatC(N1.pre, format="d", width=5),
   formatC(ci.p2given1.out$mean[2], format="f", digits=5, width=9),
   formatC(ci.p2given1.out$lower[2], format="f", digits=5, width=9),
   formatC(ci.p2given1.out$upper[2], format="f", digits=5, width=9), sep="", "\n")
cat("   Agresti-Coull    ", formatC(N12, format="d", width=2),
   formatC(N1.pre, format="d", width=5),
   formatC(ci.p2given1.out$mean[1], format="f", digits=5, width=9),
   formatC(ci.p2given1.out$lower[1], format="f", digits=5, width=9),
   formatC(ci.p2given1.out$upper[1], format="f", digits=5, width=9), sep="", "\n\n")

## REFERENCE SITE
# Determine Nij and n for pre and post treatment
Ref.df  <- subset(data.df, site == "Ref")
pre.PA  <- Ref.df$pre.PA
post.PA <- Ref.df$post.PA
```

```
N11 <- sum((pre.PA == 0) & (post.PA == 0))
N12 <- sum((pre.PA == 0) & (post.PA == 1))
N21 <- sum((pre.PA == 1) & (post.PA == 0))
N22 <- sum((pre.PA == 1) & (post.PA == 1))
N1.pre <- N11 + N12
N2.pre <- N21 + N22
n  <- N1.pre  + N2.pre
# Maximum likelihood estimates of p_2|1 and P_1|2
p.hat_2given1 <- N12/N1.pre
p.hat_1given2 <- N21/N2.pre
# Wilson and Agresti-Coull confidence intervals
cat("\nReference site ", rep("=", 64), "\n\n", sep="")
ci.p1given2.out <- binom.confint(N21, N2.pre, methods=c("ac", "wilson"))
cat("\n95% confidence intervals for p_{1|2}:\n\n")
cat("          Method   N21 N2.pre   mean    lower    upper\n")
cat("   ---------------------------------------------------\n")
cat("          Wilson    ", formatC(N21, format="d", width=2),
   formatC(N2.pre, format="d", width=5),
   formatC(ci.p1given2.out$mean[2], format="f", digits=5, width=9),
   formatC(ci.p1given2.out$lower[2], format="f", digits=5, width=9),
   formatC(ci.p1given2.out$upper[2], format="f", digits=5, width=9), sep="", "\n")
cat("   Agresti-Coull    ", formatC(N21, format="d", width=2),
   formatC(N2.pre, format="d", width=5),
   formatC(ci.p1given2.out$mean[1], format="f", digits=5, width=9),
   formatC(ci.p1given2.out$lower[1], format="f", digits=5, width=9),
   formatC(ci.p1given2.out$upper[1], format="f", digits=5, width=9), sep="", "\n\n")
ci.p2given1.out <- binom.confint(N12, N1.pre, methods=c("ac", "wilson"))
cat("95% confidence intervals for p_{2|1}:\n\n")
cat("          Method   N12 N1.pre   mean    lower    upper\n")
cat("   ---------------------------------------------------\n")
cat("          Wilson    ", formatC(N12, format="d", width=2),
   formatC(N1.pre, format="d", width=5),
   formatC(ci.p2given1.out$mean[2], format="f", digits=5, width=9),
   formatC(ci.p2given1.out$lower[2], format="f", digits=5, width=9),
   formatC(ci.p2given1.out$upper[2], format="f", digits=5, width=9), sep="", "\n")
cat("   Agresti-Coull    ", formatC(N12, format="d", width=2),
   formatC(N1.pre, format="d", width=5),
   formatC(ci.p2given1.out$mean[1], format="f", digits=5, width=9),
   formatC(ci.p2given1.out$lower[1], format="f", digits=5, width=9),
   formatC(ci.p2given1.out$upper[1], format="f", digits=5, width=9), sep="", "\n\n")
```

Question 8a: Output

```
> source("Question_08a.R")

Test site ======================================================================

95% confidence intervals for p_{1|2}:

          Method   N21 N2.pre   mean    lower    upper
   -------------------------------------------------
          Wilson    20   27  0.74074  0.55321  0.86830
   Agresti-Coull    20   27  0.74074  0.55074  0.87077

95% confidence intervals for p_{2|1}:

          Method   N12 N1.pre   mean    lower    upper
   -------------------------------------------------
          Wilson     4   33  0.12121  0.04816  0.27326
   Agresti-Coull     4   33  0.12121  0.04212  0.27930


Reference site =================================================================


95% confidence intervals for p_{1|2}:

          Method   N21 N2.pre   mean    lower    upper
   -------------------------------------------------
          Wilson     0   33  0.00000  0.00000  0.10427
   Agresti-Coull     0   33  0.00000 -0.01965  0.12392

95% confidence intervals for p_{2|1}:

          Method   N12 N1.pre   mean    lower    upper
   -------------------------------------------------
          Wilson     3   27  0.11111  0.03852  0.28058
   Agresti-Coull     3   27  0.11111  0.03031  0.28879
```

Question 8b: Program listing

```
# Title:    Question_8b.R
# Purpose:  Test HO: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0
# Command to run: source("Question_8b.R")

## LOAD AND ATTACH PACKAGE
library(Exact)

## READ IN DATA
data.df <- read.table("Faux_binary_data.csv", header=T, sep=",")

## TEST SITE
# Determine Nij and n for pre and post treatment
Test.df <- subset(data.df, site == "Test")
pre.PA  <- Test.df$pre.PA
post.PA <- Test.df$post.PA
# We want the "good" transition to be state 2 (presence, X=1) to state 1 (absence, X=0)
#     Group A: X = 2 (present before treatment)
#     Group B: X = 1 (absent before treatment)
nA1 <- sum((pre.PA == 1) & (post.PA == 0)) # presence --> absence;  transition "success" for Group A
nA2 <- sum((pre.PA == 1) & (post.PA == 1)) # presence --> presence; transition "failure" for Group A
nB1 <- sum((pre.PA == 0) & (post.PA == 1)) # absence  --> presence; transition "success" for Group B
nB2 <- sum((pre.PA == 0) & (post.PA == 0)) # absence  --> absence;  transition "failure" for Group B
nA <- nA1 + nA2
nB <- nB1 + nB2
#n  <- nA  + nB
Test.OC.mat <- matrix(c(nA1, nA - nA1, nB1, nB - nB1), ncol=2, byrow=T)
rownames(Test.OC.mat) <- c("Group A: Transition 2 -> 1", "Group B: Transition 1 -> 2")
colnames(Test.OC.mat) <- c("Success", "Failure")
cat("\nTest site ", rep("=", 70), "\n\n", sep="")
cat("Observed outcomes for Test site:\n\n")
print(Test.OC.mat)
cat("\n")
# Maximum likelihood estimates of p_2|1 and p_1|2
#  *** Note: "success" here is defined as changing state (1 --> 2 or 2 --> 1)
p.hat_1given2 <- nA1/nA # 2 --> 1
p.hat_2given2 <- 1 - p.hat_1given2
p.hat_2given1 <- nB1/nB # 1 --> 2
p.hat_1given1 <- 1 - p.hat_2given1
p.hat <- (nA1 + nB1)/(nA  + nB)  # estimated prob. of a change in state under H0
# Perform statistical tests of H0 (no difference) v. H1 (p for Group 1 < p for Group 2)
# Use prop.test()
#     Note: Hollander et al. (2014: p. 506) discuss a paper by Storer & Kim (1990), who argue that
#     Yates's continuity correction should NOT be applied.
Test.prop.test.out <- prop.test(x=Test.OC.mat, correct=F, alternative="greater")
cat("Results for prop.test of HO: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0 ------------------\n")
cat("   Estimate:", Test.prop.test.out$estimate[1] - Test.prop.test.out$estimate[2], "\n")
cat("   P-value: ", Test.prop.test.out$p.value, "\n\n")
# Use fisher.test() (exact conditional binomial test)
Test.fisher.test.out <- fisher.test(Test.OC.mat, alternative="greater")
cat("Results for fisher.test of HO: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0 ----------------\n")
cat("   Estimate:", p.hat_1given2 - p.hat_2given1, "\n")
cat("   P-value: ", Test.fisher.test.out$p.value, "\n\n")
# Barnard's exact unconditional test
Test.exact.test.out <- exact.test(data=Test.OC.mat, method="csm", alternative="greater", to.plot=F)
cat("Results for exact.test of HO: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0 ----------------\n")
cat("   Estimate:", Test.exact.test.out$estimate, "\n")
cat("   P-value: ", Test.exact.test.out$p.value, "\n\n")

## REFERENCE SITE
# Determine Nij and n for pre and post treatment
Ref.df <- subset(data.df, site == "Ref")
```

```
pre.PA  <- Ref.df$pre.PA
post.PA <- Ref.df$post.PA
# We want the "good" transition to be state 2 (presence, X=1) to state 1 (absence, X=0)
#     Group A: X = 2 (present before treatment)
#     Group B: X = 1 (absent before treatment)
nA1 <- sum((pre.PA == 1) & (post.PA == 0)) # presence --> absence;  transition "success" for Group A
nA2 <- sum((pre.PA == 1) & (post.PA == 1)) # presence --> presence; transition "failure" for Group A
nB1 <- sum((pre.PA == 0) & (post.PA == 1)) # absence  --> presence; transition "success" for Group B
nB2 <- sum((pre.PA == 0) & (post.PA == 0)) # absence  --> absence;  transition "failure" for Group B
nA <- nA1 + nA2
nB <- nB1 + nB2
#n  <- n1  + n2
Ref.OC.mat <- matrix(c(nA1, nA - nA1, nB1, nB - nB1), ncol=2, byrow=T)
rownames(Ref.OC.mat) <- c("Group A: Transition 2 -> 1", "Group B: Transition 1 -> 2")
colnames(Ref.OC.mat) <- c("Success", "Failure")
cat("\nReference site ", rep("=", 65), "\n\n", sep="")
cat("Observed outcomes for Reference site:\n\n")
print(Ref.OC.mat)
cat("\n")
# Maximum likelihood estimates of p_2|1 and P_1|2
#  *** Note: "success" here is defined as changing state (1 --> 2 or 2 --> 1)
p.hat_1given2 <- nA1/nA # 2 --> 1
p.hat_2given1 <- nB1/nB # 1 --> 2
p.hat_2given2 <- 1 - p.hat_1given2
p.hat_1given1 <- 1 - p.hat_2given1
p.hat <- (nA1 + nB1)/(nA  + nB)  # estimated prob. of a change in state under H0
# Perform statistical tests of H0 (no difference) v. H1 (p for Group 1 < p for Group 2)
# Use prop.test() (exact conditional binomial test)
#     Note: Hollander et al. (2014: p. 506) discuss a paper by Storer & Kim (1990), who argue that
#     Yates's continuity correction should NOT be applied.
Ref.prop.test.out <- prop.test(x=Ref.OC.mat, correct=F, alternative="greater")
cat("Results for prop.test of H0: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0:\n")
cat("   Estimate:", Ref.prop.test.out$estimate[1] - Ref.prop.test.out$estimate[2], "\n")
cat("   P-value: ", Ref.prop.test.out$p.value, "\n\n")
# Use fisher.test() (exact conditional binomial test)
Ref.fisher.test.out <- fisher.test(Ref.OC.mat, alternative="greater")
cat("Results for fisher.test of H0: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0:\n")
cat("   Estimate:", p.hat_1given2 - p.hat_2given1, "\n")
cat("   P-value: ", Ref.fisher.test.out$p.value, "\n\n")
# Barnard's exact unconditional test
Ref.exact.test.out <- exact.test(data=Ref.OC.mat, method="csm", alternative="greater", to.plot=F)
cat("Results for exact.test of H0: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0:\n")
cat("   Estimate:", Ref.exact.test.out$estimate, "\n")
cat("   P-value: ", Ref.exact.test.out$p.value, "\n\n")
```

Question 8b: Output

```
> source("Question_8b.R")

Test site ========================================================================

Observed outcomes for Test site:

                           Success Failure
Group A: Transition 2 -> 1      20       7
Group B: Transition 1 -> 2       4      29

Results for prop.test of H0: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0 ------------------
   Estimate: 0.6195286
   P-value:  5.488768e-07

Results for fisher.test of H0: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0 ----------------
   Estimate: 0.6195286
   P-value:  1.053923e-06

Results for exact.test of H0: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0 -----------------
   Estimate: 0.6195286
   P-value:  2.888001e-07


Reference site ===================================================================

Observed outcomes for Reference site:

                           Success Failure
Group A: Transition 2 -> 1       0      33
Group B: Transition 1 -> 2       3      24

Results for prop.test of H0: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0:
   Estimate: -0.1111111
   P-value:  0.9752697

Results for fisher.test of H0: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0:
   Estimate: -0.1111111
   P-value:  1

Results for exact.test of H0: p_1|2 - p_2|1 = 0 v. H1: p1|2 - p2|1 > 0:
   Estimate: -0.1111111
   P-value:  1

Warning message:
In prop.test(x = Ref.OC.mat, correct = F, alternative = "greater") :
  Chi-squared approximation may be incorrect
```

Question 8c: Program listing

```
# Title:    Question_8c.R
# Purpose:  Estimate p_1|2 - p_2|1 and its 95% confidence interval
# Command to run: source("Question_8c.R")

## LOAD AND ATTACH PACKAGES
library(pairwiseCI)
library(ExactCIdiff)

## READ IN DATA
data.df <- read.table("Faux_binary_data.csv", header=T, sep=",")

## SPECIFY THE CONFIDENCE LEVEL FOR THE CIS
CL <- 0.95
cat("\n\nCONFIDENCE INTERVALS FOR THE DIFFERENCE BETWEEN TWO SUCCESS PROBABILITIES\n\n")
cat("   Confidence level =", CL, "\n\n")
cat("   X below is the state of a survey point before treatment.\n\n")
cat("      X=1: Desirable state, invasive plant absent\n")
cat("      X=2: Undesirable state, invasive plant present\n\n")
cat("   p_(X=i) is the success probability for plants in pre-treatment state X=i,\n")
cat("      where success means changing state from i to j != i following treatment\n")

## TEST SITE
cat("\n\nTest Site ", rep("=", 68), "\n\n", sep="")
# Determine Nij and n for pre and post treatment
Test.df <- subset(data.df, site == "Test")
pre.PA  <- Test.df$pre.PA
post.PA <- Test.df$post.PA
nA1 <- sum((pre.PA == 1) & (post.PA == 0)) # presence --> absence;  transition "success" for Group A
nA2 <- sum((pre.PA == 1) & (post.PA == 1)) # presence --> presence; transition "failure" for Group A
nB1 <- sum((pre.PA == 0) & (post.PA == 1)) # absence  --> presence; transition "success" for Group B
nB2 <- sum((pre.PA == 0) & (post.PA == 0)) # absence  --> absence;  transition "failure" for Group B
nA <- nA1 + nA2
nB <- nB1 + nB2
# Note: pairwiseCI determines order of proportions alphabetically by group names
N.df <- data.frame(group=c("p_(X=2)", "p_(X=1)"), success=c(nA1, nB1), failure=c(nA2, nB2))
# Newcombe Hybrid Score interval
pci <- pairwiseCI(cbind(success, failure) ~ group, data=N.df, method="Prop.diff", CImethod="NHS",
   conf.level=CL)
pci.sum <- summary(pci)
cat("\n")
cat("Newcombe hybrid score interval\n\n")
cat("                  estimate    lower    upper\n")
str.1 <- formatC(as.numeric(pci.sum$estimate), digits=3, width=8, format="f")
str.2 <- formatC(as.numeric(pci.sum$conf.int[1]), digits=3, width=8, format="f")
str.3 <- formatC(as.numeric(pci.sum$conf.int[2]), digits=3, width=8, format="f")
cat(N.df$group[1], "-", N.df$group[2], str.1, str.2, str.3, "\n\n")
# Agresti-Caffo interval
pci <- pairwiseCI(cbind(success, failure) ~ group, data=N.df, method="Prop.diff", CImethod="AC",
   conf.level=CL)
pci.sum <- summary(pci)
cat("\n")
cat("Agresti-Caffo interval\n\n")
cat("                  estimate    lower    upper\n")
str.1 <- formatC(as.numeric(pci.sum$estimate), digits=3, width=8, format="f")
str.2 <- formatC(as.numeric(pci.sum$conf.int[1]), digits=3, width=8, format="f")
str.3 <- formatC(as.numeric(pci.sum$conf.int[2]), digits=3, width=8, format="f")
cat(N.df$group[1], "-", N.df$group[2], str.1, str.2, str.3, "\n\n")
# Wang exact unconditional interval
X2.tot  <- sum(N.df[1, 2:3])
X1.tot  <- sum(N.df[2, 2:3])
X2.succ <- N.df[1, 2] # successes (not.resprouted)
```

```
X1.succ <- N.df[2, 2] # failure   (resprouted)
cat("\nComputing exact unconditional confidence interval. This will take a while...")
xcidiff.out <- BinomCI(X2.tot, X1.tot, X2.succ, X1.succ, CItype="Two.sided", conf.level=CL)
cat("\n\nWang interval\n\n")
cat("                  estimate    lower    upper\n")
str.1 <- formatC(xcidiff.out$estimate, digits=3, width=8, format="f")
str.2 <- formatC(xcidiff.out$ExactCI[1], digits=3, width=8, format="f")
str.3 <- formatC(xcidiff.out$ExactCI[2], digits=3, width=8, format="f")
cat("p_(X=2) - p_(X=1)", str.1, str.2, str.3, "\n\n")

## REFERENCE SITE
cat("\nReference Site ", rep("=", 69), "\n\n", sep="")
# Determine Nij and n for pre and post treatment
Ref.df <- subset(data.df, site == "Ref")
pre.PA  <- Ref.df$pre.PA
post.PA <- Ref.df$post.PA
nA1 <- sum((pre.PA == 1) & (post.PA == 0)) # presence --> absence;  transition "success" for Group A
nA2 <- sum((pre.PA == 1) & (post.PA == 1)) # presence --> presence; transition "failure" for Group A
nB1 <- sum((pre.PA == 0) & (post.PA == 1)) # absence  --> presence; transition "success" for Group B
nB2 <- sum((pre.PA == 0) & (post.PA == 0)) # absence  --> absence;  transition "failure" for Group B
nA <- nA1 + nA2
nB <- nB1 + nB2
# Note: pairwiseCI determines order of proportions alphabetically by group names
N.df <- data.frame(group=c("p_(X=2)", "p_(X=1)"), success=c(nA1, nB1), failure=c(nA2, nB2))
# Newcombe Hybrid Score interval
pci <- pairwiseCI(cbind(success, failure) ~ group, data=N.df, method="Prop.diff", CImethod="NHS",
   conf.level=CL)
pci.sum <- summary(pci)
cat("\n")
cat("Newcombe hybrid score interval\n\n")
cat("                  estimate    lower    upper\n")
str.1 <- formatC(as.numeric(pci.sum$estimate), digits=3, width=8, format="f")
str.2 <- formatC(as.numeric(pci.sum$conf.int[1]), digits=3, width=8, format="f")
str.3 <- formatC(as.numeric(pci.sum$conf.int[2]), digits=3, width=8, format="f")
cat(N.df$group[1], "-", N.df$group[2], str.1, str.2, str.3, "\n\n")
# Agresti-Caffo interval
pci <- pairwiseCI(cbind(success, failure) ~ group, data=N.df, method="Prop.diff", CImethod="AC",
   conf.level=CL)
pci.sum <- summary(pci)
cat("\n")
cat("Agresti-Caffo interval\n\n")
cat("                  estimate    lower    upper\n")
str.1 <- formatC(as.numeric(pci.sum$estimate), digits=3, width=8, format="f")
str.2 <- formatC(as.numeric(pci.sum$conf.int[1]), digits=3, width=8, format="f")
str.3 <- formatC(as.numeric(pci.sum$conf.int[2]), digits=3, width=8, format="f")
cat(N.df$group[1], "-", N.df$group[2], str.1, str.2, str.3, "\n\n")
# Wang exact unconditional interval
X2.tot  <- sum(N.df[1, 2:3])
X1.tot  <- sum(N.df[2, 2:3])
X2.succ <- N.df[1, 2] # successes (not.resprouted)
X1.succ <- N.df[2, 2] # failure   (resprouted)
cat("\nComputing exact unconditional confidence interval. This will take a while...")
xcidiff.out <- BinomCI(X2.tot, X1.tot, X2.succ, X1.succ, CItype="Two.sided", conf.level=CL)
cat("\n\nWang interval\n\n")
cat("                  estimate    lower    upper\n")
str.1 <- formatC(xcidiff.out$estimate, digits=3, width=8, format="f")
str.2 <- formatC(xcidiff.out$ExactCI[1], digits=3, width=8, format="f")
str.3 <- formatC(xcidiff.out$ExactCI[2], digits=3, width=8, format="f")
cat("p_(X=2) - p_(X=1)", str.1, str.2, str.3, "\n\n")
```

Question 8c: Output

```
> source("Question_8c.R")


CONFIDENCE INTERVALS FOR THE DIFFERENCE BETWEEN TWO SUCCESS PROBABILITIES

   Confidence level = 0.95

   X below is the state of a survey point before treatment.

      X=1: Desirable state, invasive plant absent
      X=2: Undesirable state, invasive plant present

   p_(X=i) is the success probability for plants in pre-treatment state X=i,
      where success means changing state from i to j != i following treatment


Test Site ======================================================================


Newcombe hybrid score interval

                  estimate    lower    upper
p_(X=2) - p_(X=1)    0.620    0.378    0.766


Agresti-Caffo interval

                  estimate    lower    upper
p_(X=2) - p_(X=1)    0.620    0.382    0.781


Computing exact unconditional confidence interval. This will take a while...

Wang interval

                  estimate    lower    upper
p_(X=2) - p_(X=1)    0.620    0.380    0.798


Reference Site =================================================================


Newcombe hybrid score interval

                  estimate    lower    upper
p_(X=2) - p_(X=1)   -0.111   -0.281    0.016


Agresti-Caffo interval

                  estimate    lower    upper
p_(X=2) - p_(X=1)   -0.111   -0.246    0.028


Computing exact unconditional confidence interval. This will take a while...

Wang interval

                  estimate    lower    upper
p_(X=2) - p_(X=1)   -0.111   -0.292    0.012
```

2.1.2 Methods based on quantitative data

Question 9: Program listing

```
# Title:    Question_9.R
# Purpose:  Estimate mean pre- and post-treatment local density and 95% confidence interval
#           for Test and Reference sites
# Command to run: source("Question_9.R")

## LOAD AND ATTACH PACKAGE
library(MKinfer)

## READ IN DATA
data.df <- read.table("Faux_quantitative_data.csv", header=T, sep=",")
# Specify radius of disk around each survey point
r_m <- 2 # radius in meters; implies means and CIs have units of no. per meter^2

## TEST SITE
cat("\n\nTest Site ", rep("=", 73), "\n\n", sep="")
# Extract pre and post abundance data for Test site and convert to density
Test.df <- subset(x=data.df, subset=(site == "Test"))
Area <- pi*r_m^2
density.pre  <- Test.df$pre.abund/Area
density.post <- Test.df$post.abund/Area
# Bootstrap estimates of mean density and 95% confidence interval for pre- and post-treatment data
meanCI.out <- meanCI(x=density.pre, boot=T, alternative="two.sided", bootci.type="all")
cat("Mean and 2-sided bootstrap confidence intervals for pre-treatment data: -----------\n")
print(meanCI.out)
meanCI.out <- meanCI(x=density.post, boot=T, alternative="two.sided", bootci.type="all")
cat("Mean and 2-sided bootstrap confidence intervals for post-treatment data:  ---------\n")
print(meanCI.out)

## REFERENCE SITE
cat("\nReference Site ", rep("=", 70), "\n\n", sep="")
# Extract pre and post abundance data for Ref site and convert to density
Ref.df <- subset(x=data.df, subset=(site == "Ref"))
density.pre  <- Ref.df$pre.abund/(pi*r_m^2)
density.post <- Ref.df$post.abund/(pi*r_m^2)
# Bootstrap estimates of mean density and 95% confidence interval
meanCI.out <- meanCI(x=density.pre, boot=T, alternative="two.sided", bootci.type="all")
cat("Mean and 2-sided bootstrap confidence intervals for pre-treatment data: -----------\n")
print(meanCI.out)
meanCI.out <- meanCI(x=density.post, boot=T, alternative="two.sided", bootci.type="all")
cat("Mean and 2-sided bootstrap confidence intervals for post-treatment data:  ---------\n")
print(meanCI.out)
```

Question 9: Output

```
> source("Question_9.R")


Test Site ==============================================================================

Mean and 2-sided bootstrap confidence intervals for pre-treatment data:

        Bootstrap confidence interval(s)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 9999 bootstrap replicates

CALL :
boot.ci(boot.out = boot.out, conf = 1 - alpha, type = bootci.type)

Intervals :
Level      Normal              Basic             Studentized
95%   ( 0.3854,  0.7040 )   ( 0.3833,  0.7016 )   ( 0.3867,  0.7149 )

Level     Percentile            BCa
95%   ( 0.3886,  0.7069 )   ( 0.3913,  0.7082 )
Calculations and Intervals on Original Scale

sample estimates:
     mean        sd
0.5451057 0.6277616

Mean and 2-sided bootstrap confidence intervals for post-treatment data:

        Bootstrap confidence interval(s)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 9999 bootstrap replicates

CALL :
boot.ci(boot.out = boot.out, conf = 1 - alpha, type = bootci.type)

Intervals :
Level      Normal              Basic             Studentized
95%   ( 0.0081,  0.0293 )   ( 0.0066,  0.0279 )   ( 0.0094,  0.0338 )

Level     Percentile            BCa
95%   ( 0.0093,  0.0305 )   ( 0.0093,  0.0305 )
Calculations and Intervals on Original Scale

sample estimates:
      mean         sd
0.01856808 0.04237945


Reference Site =========================================================================

Mean and 2-sided bootstrap confidence intervals for pre-treatment data:

        Bootstrap confidence interval(s)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 9999 bootstrap replicates

CALL :
boot.ci(boot.out = boot.out, conf = 1 - alpha, type = bootci.type)
```

```
Intervals :
Level      Normal              Basic             Studentized
95%   ( 0.5065,  0.8210 )   ( 0.5080,  0.8223 )   ( 0.5017,  0.8275 )

Level     Percentile            BCa
95%   ( 0.5066,  0.8210 )   ( 0.5000,  0.8157 )
Calculations and Intervals on Original Scale

sample estimates:
     mean        sd
0.6644719 0.6222664

Mean and 2-sided bootstrap confidence intervals for post-treatment data:

        Bootstrap confidence interval(s)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 9999 bootstrap replicates

CALL :
boot.ci(boot.out = boot.out, conf = 1 - alpha, type = bootci.type)

Intervals :
Level      Normal              Basic             Studentized
95%   ( 0.5902,  0.9484 )   ( 0.5915,  0.9456 )   ( 0.5850,  0.9512 )

Level     Percentile            BCa
95%   ( 0.5929,  0.9470 )   ( 0.5955,  0.9496 )
Calculations and Intervals on Original Scale

sample estimates:
     mean        sd
0.7692489 0.7168214
```

Question 10

```
# Title:    Question_10.R
# Purpose:  Given target density mu_D*, test H0: mu_D' = mu_D* v. H1: mu_D' < mu_D*
# Command to run: source("Question_10.R")

## LOAD AND ATTACH PACKAGE
library(MKinfer)

## READ IN DATA
data.df <- read.table("Faux_quantitative_data.csv", header=T, sep=",")
# Specify radius of disk around each survey point
r_m <- 2 # radius in meters; implies means and CIs have units of no. per meter^2
# Specify target density
mu.D.target <- 0.5 # Units: plants/m^2

## TEST SITE
cat("\n\nTest Site ", rep("=", 73), "\n\n", sep="")
# Extract post-treatment abundance data from site Test and convert to density
Test.df <- subset(x=data.df, subset=(site == "Test"))
density.post <- Test.df$post.abund/(pi*r_m^2)
# Boot t-test
boot.out <- boot.t.test(x=density.post, alternative="less", mu=mu.D.target)
target <- formatC(mu.D.target, digits=1, width=3, format="f")
cat("Target density:", target, "\n")
print(boot.out)

## REFERENCE SITE
cat("\nReference Site ", rep("=", 70), "\n\n", sep="")
# Extract post-treatment abundance data from site Ref and convert to density
Ref.df <- subset(x=data.df, subset=(site == "Ref"))
density.post <- Ref.df$post.abund/(pi*r_m^2)
# Boot t-test
boot.out <- boot.t.test(x=density.post, alternative="less", mu=mu.D.target)
target <- formatC(mu.D.target, digits=1, width=3, format="f")
cat("Target density:", target, "\n")
print(boot.out)
```

Question 10: Output

```
> source("Question_10.R")


Test Site ======================================================================

Target density: 0.5

        Bootstrap One Sample t-test

data:  density.post
bootstrap p-value < 2.2e-16
bootstrap mean of x (SE) = 0.01860296 (0.005357992)
95 percent bootstrap percentile confidence interval:
       -Inf 0.02785212

Results without bootstrap:
t = -87.994, df = 59, p-value < 2.2e-16
alternative hypothesis: true mean is less than 0.5
95 percent confidence interval:
      -Inf 0.0277109
sample estimates:
 mean of x
0.01856808


Reference Site =================================================================

Target density: 0.5

        Bootstrap One Sample t-test

data:  density.post
bootstrap p-value = 0.9975
bootstrap mean of x (SE) = 0.7700209 (0.09175041)
95 percent bootstrap percentile confidence interval:
      -Inf 0.9204461

Results without bootstrap:
t = 2.9095, df = 59, p-value = 0.9975
alternative hypothesis: true mean is less than 0.5
95 percent confidence interval:
      -Inf 0.9238939
sample estimates:
mean of x
0.7692489
```

Questions 11-12: Program listing

```
# Title:    Question_11-12.R
# Purpose:  Test H0: mu_D' = mu_D v. H1: mu_D' < mu_D, and
#           estimate 95% confidence interval for mu_D' - mu_D
# Command to run: source("Question_11-12.R")

## LOAD AND ATTACH PACKAGE
library(MKinfer)

## READ IN DATA
data.df <- read.table("Faux_quantitative_data.csv", header=T, sep=",")
# Specify radius of disk around each survey point (in meters)
r_m <- 2
cat("\n\n***** Note: In output below, x refers to post-treatment and y refers to pre-treatment\n")
cat("***** H0: mu_D'(x) - mu_D(y) = 0,  H1: mu_D'(x) - mu_D(y) < 0 (i.e., mu_D'(x) < mu_D(y))\n\n")

## TEST SITE
cat("\nTest site ", rep("=", 73), "\n\n", sep="")
# Extract pre- and post-treatment abundance data from Test site and convert to density
Test.df <- subset(x=data.df, subset=(site == "Test"))
pre.density <- Test.df$pre.abund/(pi*r_m^2)
post.density <- Test.df$post.abund/(pi*r_m^2)
# Boot t-test
# Note: in boot.t.test(x, y, ...), one-sided H1s are x < y and x > y, with x and y in that order
boot.out   <- boot.t.test(x=post.density, y=pre.density, alternative="less", paired=T)
boot.out.2 <- boot.t.test(x=post.density, y=pre.density, alternative="two.sided", paired=T)
cat("One-sided bootstrap t-test:\n")
print(boot.out)
cat("Two-sided bootstrap t-test to get a two-sided confidence interval:\n")
print(boot.out.2)
# Permutation t-test
perm.out   <- perm.t.test(x=post.density, y=pre.density, alternative="less", paired=T)
perm.out.2 <- perm.t.test(x=post.density, y=pre.density, alternative="two.sided", paired=T)
cat("One-sided permutation t-test:\n")
print(perm.out)
cat("Two-sided permutation t-test to get a two-sided confidence interval:\n")
print(perm.out.2)

## REFERENCE SITE
cat("\nReference site ", rep("=", 68), "\n\n", sep="")
# Extract pre- and post-treatment abundance data from site Ref and convert to density
Ref.df <- subset(x=data.df, subset=(site == "Ref"))
pre.density <- Ref.df$pre.abund/(pi*r_m^2)
post.density <- Ref.df$post.abund/(pi*r_m^2)
# Boot t-test
boot.out   <- boot.t.test(x=post.density, y=pre.density, alternative="less", paired=T)
boot.out.2 <- boot.t.test(x=post.density, y=pre.density, alternative="two.sided", paired=T)
cat("One-sided bootstrap t-test:\n")
print(boot.out)
cat("Two-sided bootstrap t-test to get a two-sided confidence interval:\n")
print(boot.out.2)
# Permutation t-test
perm.out   <- perm.t.test(x=post.density, y=pre.density, alternative="less", paired=T)
perm.out.2 <- perm.t.test(x=post.density, y=pre.density, alternative="two.sided", paired=T)
cat("One-sided permutation t-test:\n")
print(perm.out)
cat("Two-sided permutation t-test to get a two-sided confidence interval:\n")
print(perm.out.2)
```

Questions 11-12: Output

```
> source("Question_11-12.R")


***** Note: In output below, x refers to post-treatment and y refers to pre-treatment
***** H0: mu_D'(x) - mu_D(y) = 0,  H1: mu_D'(x) - mu_D(y) < 0 (i.e., mu_D'(x) < mu_D(y))


Test site ================================================================================

One-sided bootstrap t-test:

        Bootstrap Paired t-test

data:  post.density and pre.density
bootstrap p-value < 2.2e-16
bootstrap mean of the differences (SE) = -0.5256734 (0.08001393)
95 percent bootstrap percentile confidence interval:
       -Inf -0.3965611

Results without bootstrap:
t = -6.5214, df = 59, p-value = 8.689e-09
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
       -Inf -0.3916136
sample estimates:
mean of the differences
             -0.5265376

Two-sided bootstrap t-test to get a two-sided confidence interval:

        Bootstrap Paired t-test

data:  post.density and pre.density
bootstrap p-value < 2.2e-16
bootstrap mean of the differences (SE) = -0.5260615 (0.07998695)
95 percent bootstrap percentile confidence interval:
 -0.6830400 -0.3726878

Results without bootstrap:
t = -6.5214, df = 59, p-value = 1.738e-08
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.6880979 -0.3649773
sample estimates:
mean of the differences
             -0.5265376

One-sided permutation t-test:

        Permutation Paired t-test

data:  post.density and pre.density
(Monte-Carlo) permutation p-value < 2.2e-16
permutation mean of the differences (SE) = -0.5265812 (0.07995951)
95 percent (Monte-Carlo) permutation percentile confidence interval:
       -Inf -0.3938996

Results without permutation:
t = -6.5214, df = 59, p-value = 8.689e-09
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
```

```
       -Inf -0.3916136
sample estimates:
mean of the differences
             -0.5265376

Two-sided permutation t-test to get a two-sided confidence interval:

        Permutation Paired t-test

data:  post.density and pre.density
(Monte-Carlo) permutation p-value < 2.2e-16
permutation mean of the differences (SE) = -0.5259834 (0.07994753)
95 percent (Monte-Carlo) permutation percentile confidence interval:
 -0.6833494 -0.3657425

Results without permutation:
t = -6.5214, df = 59, p-value = 1.738e-08
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.6880979 -0.3649773
sample estimates:
mean of the differences
             -0.5265376


Reference site ====================================================================

One-sided bootstrap t-test:

        Bootstrap Paired t-test

data:  post.density and pre.density
bootstrap p-value = 1
bootstrap mean of the differences (SE) = 0.1047656 (0.02108308)
95 percent bootstrap percentile confidence interval:
      -Inf 0.1419132

Results without bootstrap:
t = 4.8868, df = 59, p-value = 1
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf 0.1406066
sample estimates:
mean of the differences
               0.104777

Two-sided bootstrap t-test to get a two-sided confidence interval:

        Bootstrap Paired t-test

data:  post.density and pre.density
bootstrap p-value < 2.2e-16
bootstrap mean of the differences (SE) = 0.1051272 (0.02116088)
95 percent bootstrap percentile confidence interval:
 0.06498827 0.14854461

Results without bootstrap:
t = 4.8868, df = 59, p-value = 8.213e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.06187406 0.14767995
sample estimates:
mean of the differences
```

```
               0.104777

One-sided permutation t-test:

        Permutation Paired t-test

data:  post.density and pre.density
(Monte-Carlo) permutation p-value = 1
permutation mean of the differences (SE) = 0.1048184 (0.02121177)
95 percent (Monte-Carlo) permutation percentile confidence interval:
      -Inf 0.1399679

Results without permutation:
t = 4.8868, df = 59, p-value = 1
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf 0.1406066
sample estimates:
mean of the differences
               0.104777

Two-sided permutation t-test to get a two-sided confidence interval:

        Permutation Paired t-test

data:  post.density and pre.density
(Monte-Carlo) permutation p-value < 2.2e-16
permutation mean of the differences (SE) = 0.1051346 (0.02121059)
95 percent (Monte-Carlo) permutation percentile confidence interval:
 0.06401565 0.14638497

Results without permutation:
t = 4.8868, df = 59, p-value = 8.213e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.06187406 0.14767995
sample estimates:
mean of the differences
               0.104777
```

2.2 Comparing treatment efficacy in pairs of treatments or species

Question 13: Program listing

```
# Title:    Question_13.R
# Purpose:  Test H0: (mu_D - mu_D')_Test = (mu_D - mu_D')_Ref versus
#           H1: (mu_D - mu_D')_Test > (mu_D - mu_D')_Ref  (i.e., determine if there is strong
#           evidence that mean density decreased more in the Test site than in the Ref site)
# Command to run: source("Question_13.R")

## LOAD AND ATTACH PACKAGE
library(MKinfer)

## READ IN DATA
data.df <- read.table("Faux_quantitative_data.csv", header=T, sep=",")

## PRE-PROCESS DATA
# Specify radius of disk around each survey point (in meters)
r_m <- 2
cat("\n\n*** Note: In output below, x and y refer to Test and Ref density decreases\n")
cat("*** H0: (mu - mu')_Test = (mu - mu')_Ref,  H1: (mu - mu')_Test > (mu - mu')_Ref\n\n")
# Extract pre- and post-treatment abundance data from site Test and convert to density
Test.df <- subset(data.df, subset=(site == "Test"))
pre.density <- Test.df$pre.abund/(pi*r_m^2)
post.density <- Test.df$post.abund/(pi*r_m^2)
# Create vector of density decreases for site Test
Test.density.decrease <- pre.density - post.density  # Positive values ==> DECREASE
# Extract abundance data from site Ref for 2017 and 2018 and convert to density
Ref.df <- subset(data.df, subset=(site == "Ref"))
pre.density <- Ref.df$pre.abund/(pi*r_m^2)
post.density <- Ref.df$post.abund/(pi*r_m^2)
# Create vector of density decreases for site Ref
Ref.density.decrease <- pre.density - post.density  # Positive values ==> DECREASE

## STATISTICAL TESTS
# Bootstrap t-test
boot.out <- boot.t.test(Test.density.decrease, Ref.density.decrease, alternative="greater")
print(boot.out)
cat("Estimated difference (Test decrease) - (Ref decrease) between pre- and post-treatment means =",
   boot.out$estimate[1] - boot.out$estimate[2], "\n")
# Permutation t-test
perm.out <- perm.t.test(Test.density.decrease, Ref.density.decrease, alternative="greater")
print(perm.out)
cat("Estimated difference (Test decrease) - (Ref decrease) between pre- and post-treatment means =",
   perm.out$estimate[1] - perm.out$estimate[2], "\n\n")
```

Question 13: Output

```
> source("Question_13.R")


*** Note: In output below, x and y refer to Test and Ref density decreases
*** H0: (mu - mu')_Test = (mu - mu')_Ref,  H1: (mu - mu')_Test > (mu - mu')_Ref


        Bootstrap Welch Two Sample t-test

data:  Test.density.decrease and Ref.density.decrease
bootstrap p-value < 2.2e-16
bootstrap difference of means (SE) = 0.632209 (0.0827818)
95 percent bootstrap percentile confidence interval:
 0.4985529       Inf

Results without bootstrap:
t = 7.5572, df = 67.28, p-value = 7.39e-11
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 0.4919878       Inf
sample estimates:
 mean of x  mean of y
 0.5265376 -0.1047770

Estimated difference (Test decrease) - (Ref decrease) between pre- and post-treatment means = 0.6313146

        Permutation Welch Two Sample t-test

data:  Test.density.decrease and Ref.density.decrease
(Monte-Carlo) permutation p-value < 2.2e-16
permutation difference of means (SE) = 0.6327124 (0.1013441)
95 percent (Monte-Carlo) permutation percentile confidence interval:
 0.4642019       Inf

Results without permutation:
t = 7.5572, df = 67.28, p-value = 7.39e-11
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 0.4919878       Inf
sample estimates:
 mean of x  mean of y
 0.5265376 -0.1047770

Estimated difference (Test decrease) - (Ref decrease) between pre- and post-treatment means = 0.6313146
```