**1. 접속하는 방법**

[인터넷으로 접속: 컴퓨터에 R 프로그램을 설치할 필요가 없음.]

URL: <https://rosabaya93-project1.shinyapps.io/Narzio/>

[R 프로그램]

해당 컴퓨터에서 처음 실행하는 경우, 아래 코드를 입력하여 패키지를 깔아준다.

install.packages("shiny"); install.packages("shinydashboard"); install.packages("readxl")

install.packages("dplyr"); install.packages("plyr"); install.packages("ggplot2");install.packages("lazyeval")

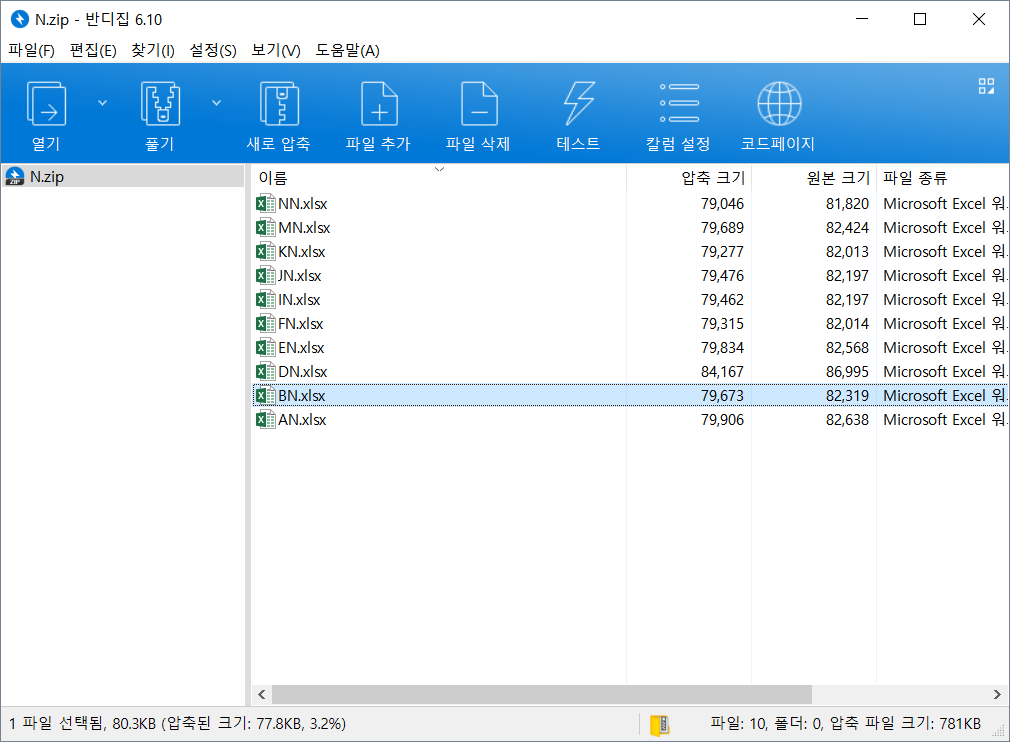
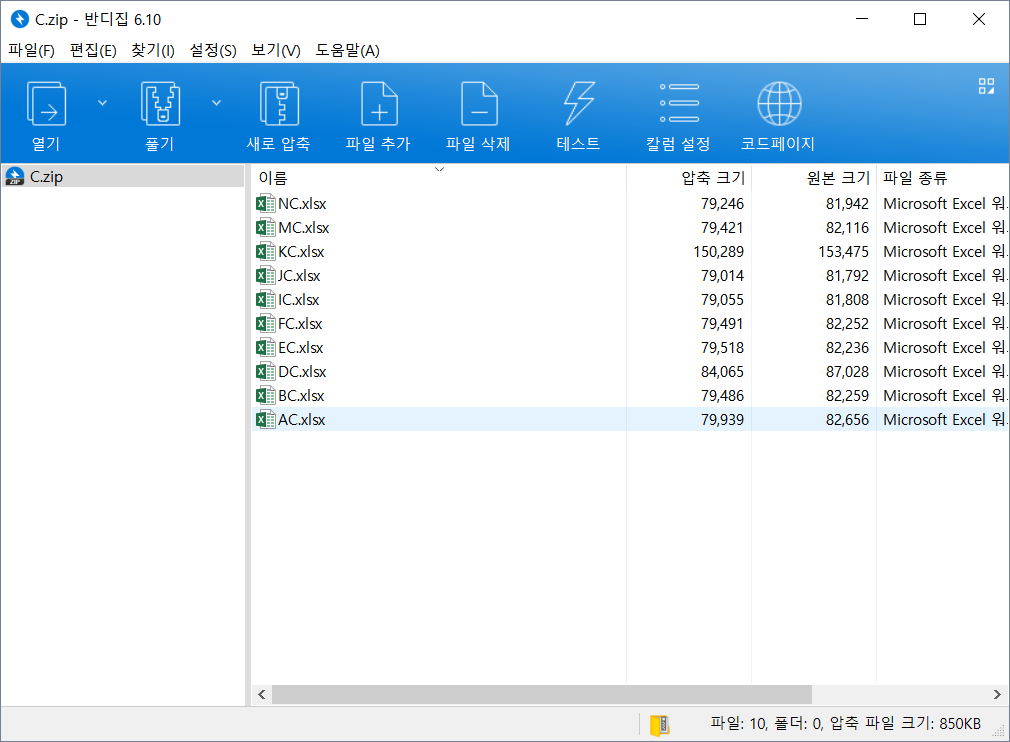
**2. 데이터 불러오기**

Narzio와 Control 데이터를 각각 zip 파일로 압축하여 불러온다.

[주의] 1) 엑셀파일의 확장자가 .xlsx임을 확인한다.

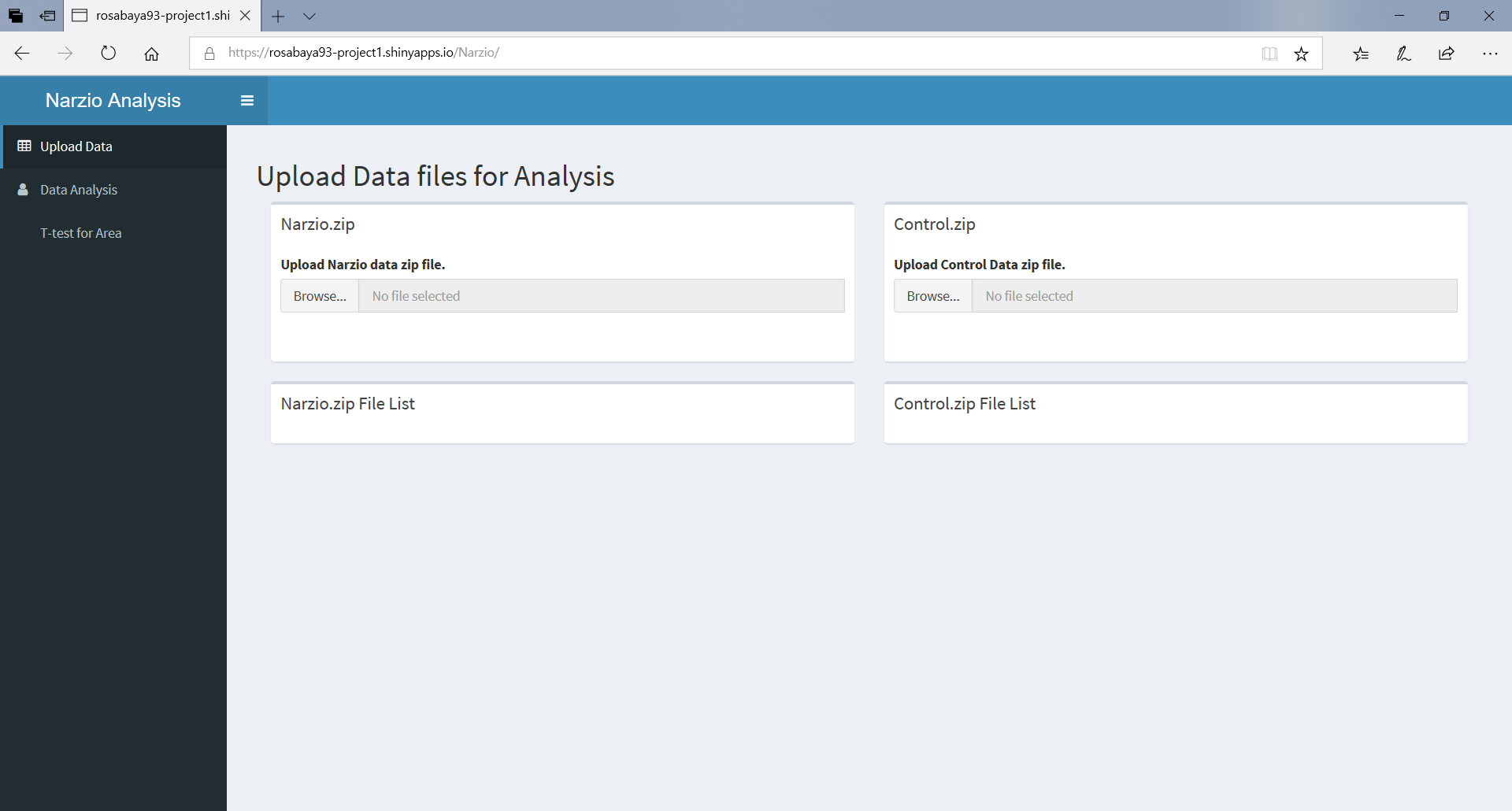
2) 엑셀파일 이름에 특수기호들을 제거한다.

3) 각각의 압축파일은 동일한 사람 순서대로 정렬한다. (ex. 1번 사람의 데이터는 각각 압축파일에서 첫번째에 위치)

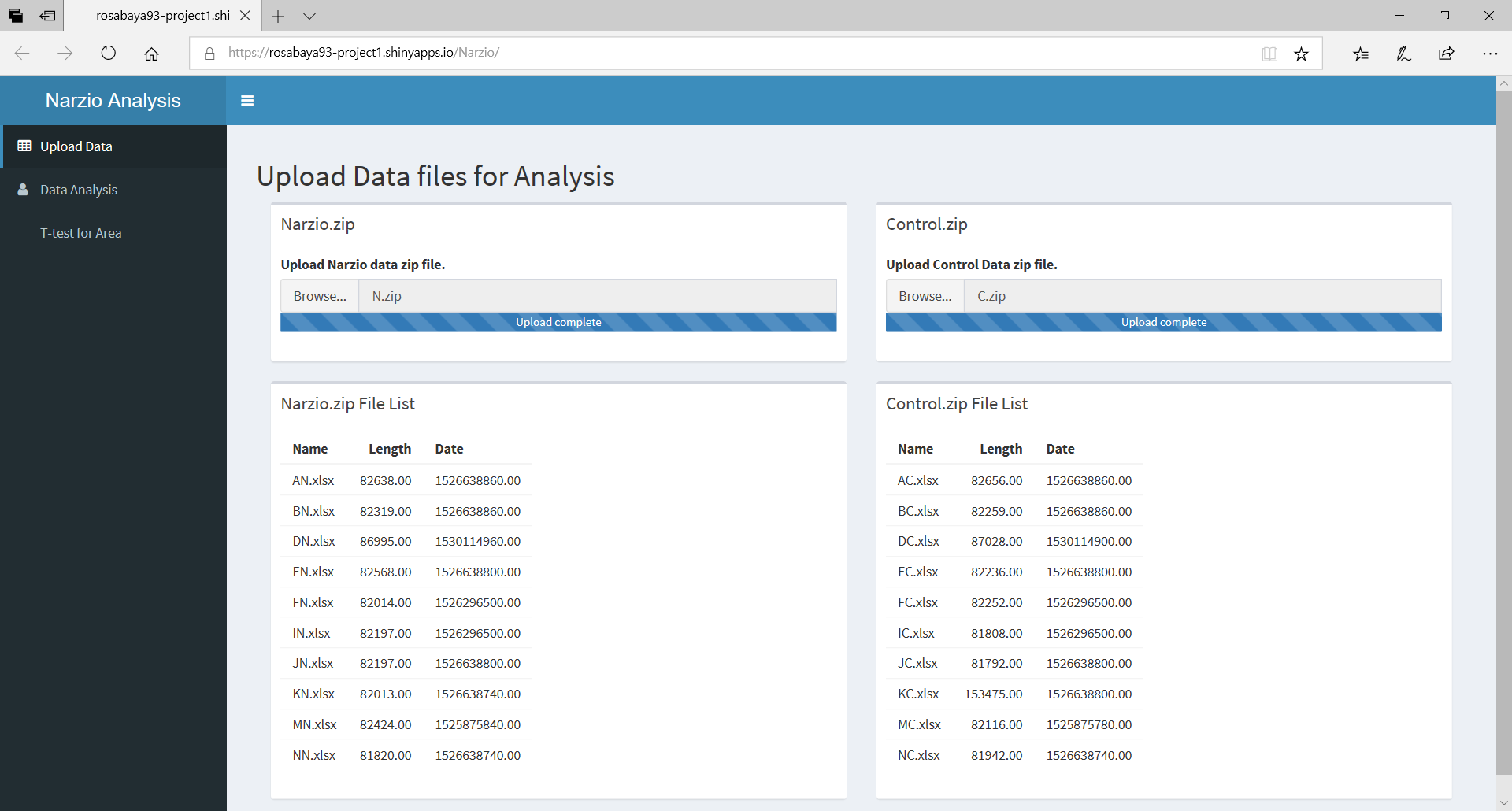
 

붉은 상자 안의 Browse를 클릭하여 Narzio 데이터 압축파일을 업로드한다.

동일하게 파란색 상자에는 Control 데이터 압축파일을 업로드한다.



압축해제 결과 출력: 데이터가 적절하게 들어갔는지 확인한다. **반드시 순서를 확인한다.**



**3. 분석 진행하기**

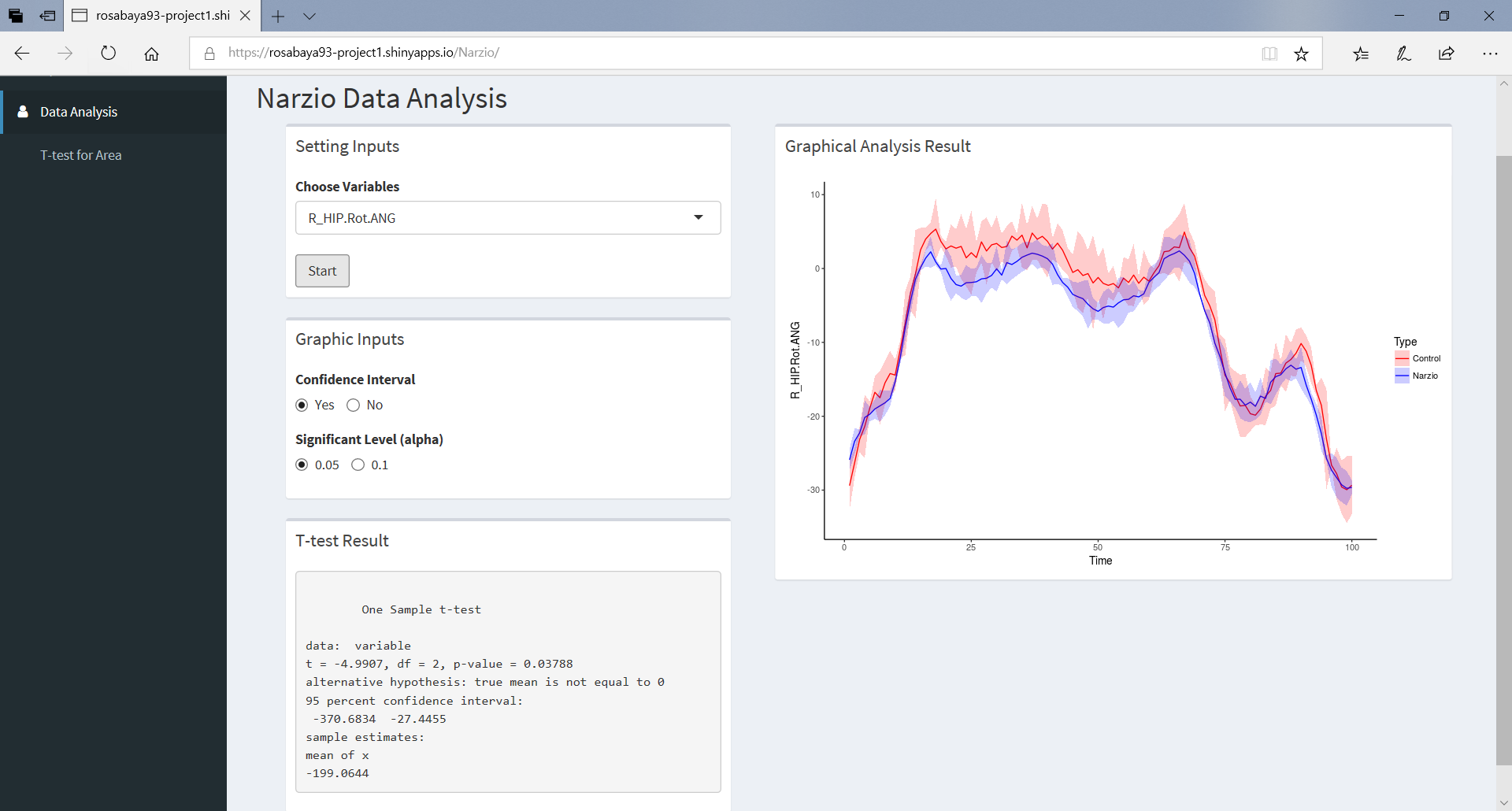
1) 분석을 진행하고자 하는 변수(붉은색 상자)를 선택한다.

2) 그래프에 출력되는 결과물에 관한 옵션(녹색 상자)을 설정한다.

Confidence Interval (신뢰구간): 도표에 신뢰구간을 표시하는지 여부

Significant Level (유의수준): 신뢰구간에 영향을 주는 수치로 클수록 신뢰구간이 좁아지나, 통계적 유의성이 떨어진다.

3) 옵션 설정을 완료한 후, [Start] 버튼을 누른다.



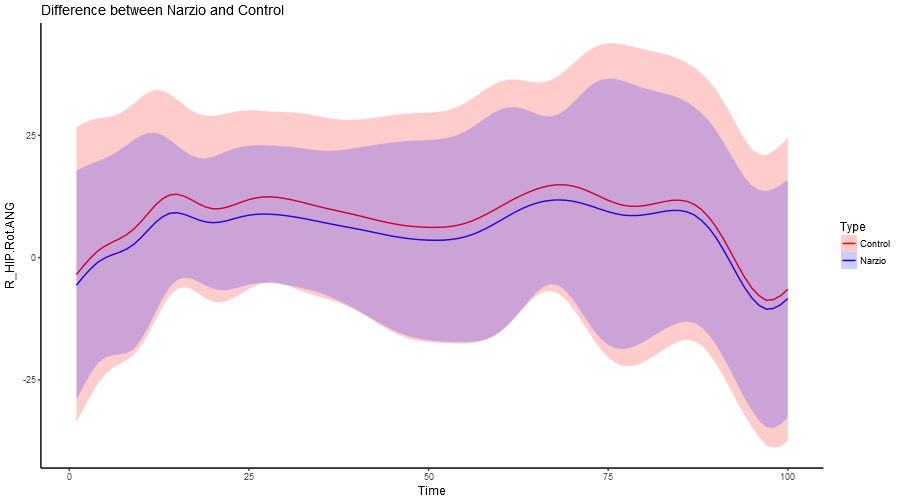
**4. 출력 결과 분석하기**

[Graphical Analysis Result] 도표는 오른쪽 버튼을 이용하여 저장/복사가 가능하다.

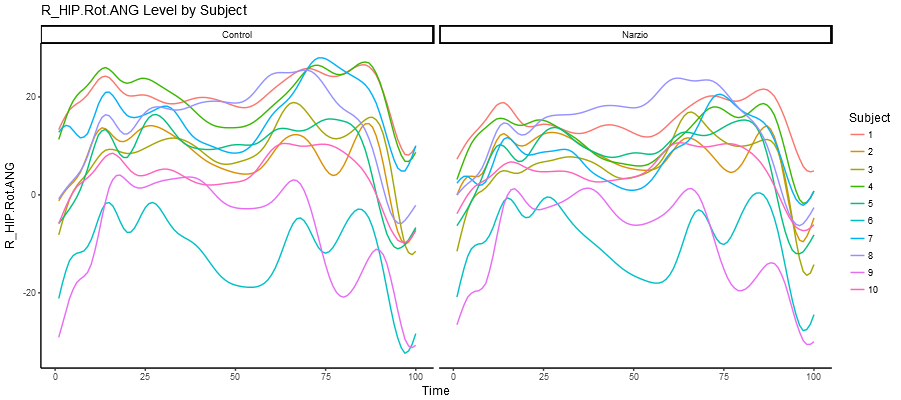
아래 도표에서 x축은 시간, y축은 R\_HIP ROT ANG 변수이다.

실선은 각각 Narzio와 Control 그룹에 있는 15명의 R\_HIP ROT ANG 변수의 시간(Time)별 평균을 의미한다.

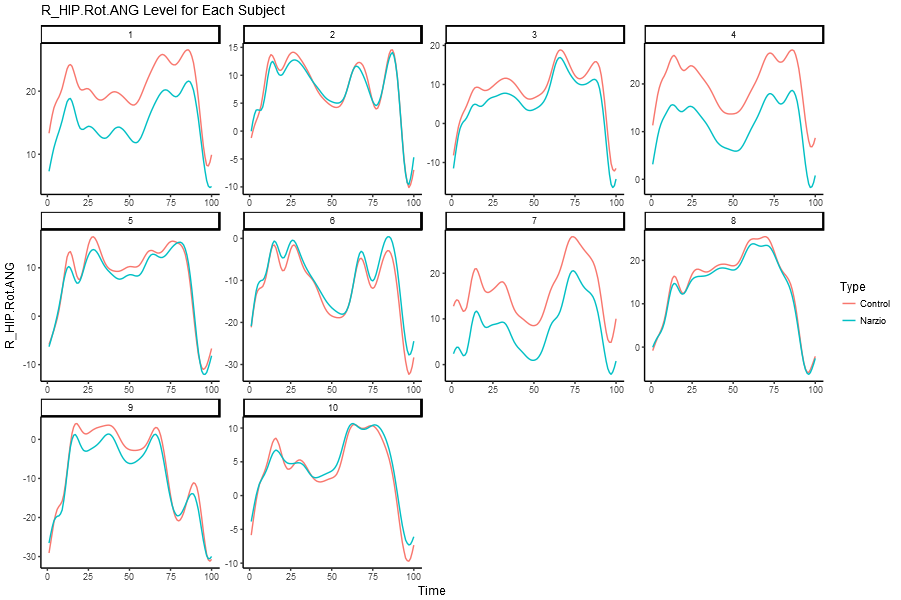
면적은 각각 Narzio와 Control 그룹에 있는 15명의 R\_HIP ROT ANG 변수의 시간(Time)별 신뢰구간을 의미한다.



그러나 본 데이터에서, 신뢰구간을 이용한 분석은 어렵다. 아래 도표에서 개인간 변동이 매우 강하며 패턴에 큰 차이를 보인다. 이로 인하여 신뢰구간이 과도하게 넓어지며, 나르지오 신발과 일반 신발의 차이를 효과적으로 검정할 수 없다.



또한 하단의 도표는 각 피험자의 특정변수(R\_HIP Rot ANG) 수준을 보여준다. 전체적으로 나르지오 신발의 수준이 일반 신발보다 낮거나 비슷한 추세를 확인할 수 있다. 이러한 추세는 1번, 4번, 7번 실험자에서 잘 나타난다. 또한 개인별 걸음걸이의 습관 차이로 인해 고유의 패턴을 확인할 수 있다. 만약 큰 차이가 보이는 피험자의 공통적인 특성(성별, 나이, 직업, 키 등)을 추출할 수 있다면 추가적인 연구를 통해, 나르지오 신발의 효과가 잘 나타나는 이용자를 특정할 수 있을 것이다.

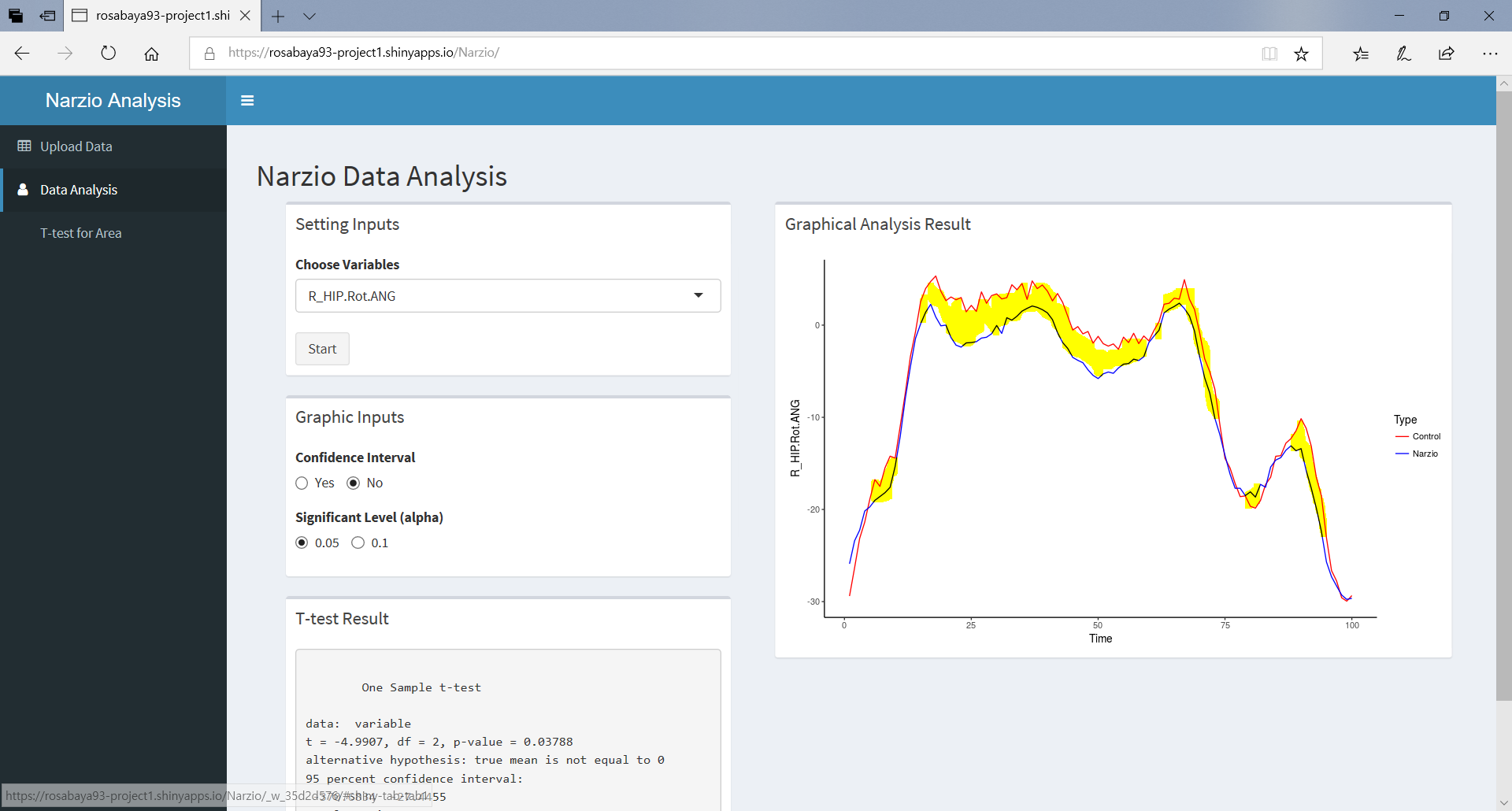


[t-test Result]

(분석의 개요)

아래 도표에서 노란색 부분의 면적은 Control과 Narzio의 차이를 보여주며, 다음과 같은 귀무가설을 설정한다.

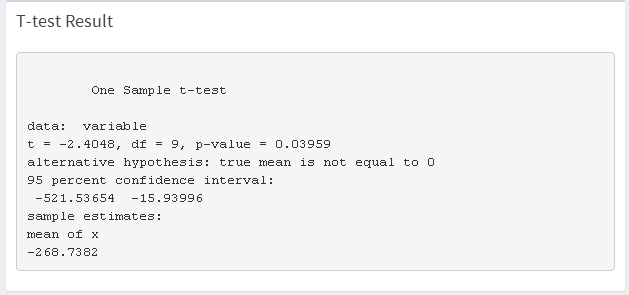
귀무가설 H0: 나르지오와 컨트롤은 차이가 없다. vs. 대립가설 H1: 나르지오와 컨트롤은 차이가 있다.



노란색 면적(A)을 새롭게 정의하면 아래와 같다.

결과적으로 피실험자 15명으로부터 관심변수(R\_HIP ROT ANG)의 15개 면적 데이터를 얻을 수 있다.

(결과 해석) 아래는 임의의 데이터 3개로 분석한 결과로 실제와는 다를 수 있다.



1) **p-value**: 유의수준(significant level: alpha)은 0.05 혹은 0.1을 이용하며 아래와 같은 결론을 내릴 수 있다.

일반적으로

🡪 귀무가설을 기각한다. 🡪 나르지오 신발과 일반 신발의 차이가 통계적으로 유의하다.

🡪 귀무가설을 기각할 수 없다 🡪 나르지오와 일반 신발의 차이가 통계적으로 유의하다고 할 수 없다.

2) **test statistic** (검정통계량): 귀무가설이 기각된 경우, 면적 A의 정의에 의해 아래와 같은 결론을 내릴 수 있다.

🡪 : Control 신발이 Narzio 신발보다 관심변수(R\_HIP ROT ANG)에서 그 크기가 크다.

🡪 : Control 신발이 Narzio 신발보다 관심변수(R\_HIP ROT ANG)에서 그 크기가 작다.

**R Codes**

library(shiny);library(shinydashboard);library(readxl);library(dplyr);library(plyr);library(ggplot2);library(lazyeval)

options(shiny.maxRequestSize=30\*1024^2)

header <- dashboardHeader(title = "Narzio Analysis")

sidebar <- dashboardSidebar(

sidebarMenu(

menuItem("Upload Data", tabName = "tab1", icon = icon("table")),

menuItem("Data Analysis", icon = icon("user", lib="font-awesome"), tabName = "tab2"),

menuItem("Additional(?)", icon = icon("chart-area", lib='font-awesome'), tabName = "tab3")

))

body <- dashboardBody(

tabItems(

## tab1

tabItem("tab1",

fluidPage(

h2("Upload Data files for Analysis"),

box(title="Narzio.zip", width=6, fileInput("narzio", "Upload Narzio data zip file.", accept='.zip')),

box(title="Control.zip", width=6, fileInput("control", "Upload Control Data zip file.", accept='.zip')),

box(title="Narzio.zip File List", width=6, tableOutput("narzio\_list")),

box(title="Control.zip File List", width=6, tableOutput("control\_list"))

)),

## tab2

tabItem("tab2",

fluidPage(

h2("Narzio Data Analysis"),

column(5,

box(title="Setting Inputs", width=12, uiOutput('choose\_columns'), actionButton("button1", "Start")),

box(title="Graphic Inputs", width=12,

radioButtons("ci", "Confidence Interval", choices=c("Yes", "No"), select="Yes", inline=TRUE),

radioButtons("alpha", "Significant Level (alpha)", choices=c("0.05", "0.1"), select="0.05", inline=TRUE)),

box(title="T-test Result", width=12, verbatimTextOutput("text1"))

),

tabBox(title="Graphical Analysis Result", width=7, height="620px",

tabPanel("Graph #1", plotOutput("plot1", height="500px")),

tabPanel("Graph #2", plotOutput("plot2",height="400px")),

tabPanel("Graph #3", plotOutput("plot3",height="600px")))

)),

## tab3

tabItem("tab3",

fluidPage(

)

)))

ui <- dashboardPage(header, sidebar, body)

server <- function(input, output, session){

narzio\_name <- eventReactive(input$narzio,{

data1 <- input$narzio

validate(need(!is.null(input$narzio), 'Plaese Upload Narzio.zip'))

narzio <- unzip(data1$datapath, list=TRUE, exdir=getwd())

return(narzio)

})

output$narzio\_list <- renderTable({narzio\_name()})

narzio\_df <- eventReactive(input$narzio,{

withProgress(message="Merging Narzio Data", value=0, {

incProgress(1/4, detail="Unzip file")

data1 <- input$narzio

narzio\_name <- unzip(data1$datapath, list=TRUE, exdir=getwd())$Name

unzip(data1$datapath, files=narzio\_name, exdir=getwd(), overwrite=TRUE)

incProgress(1/4, detail="Loading Unzipped Files")

narzio\_path <- paste(getwd(), narzio\_name, sep='/')

narzio\_dat <- lapply(narzio\_path, function(x) data.frame(read\_xlsx(x, range=cell\_rows(24:124))[,c(1:9, 13:21,40:87)]))

incProgress(1/4, detail="Bulid Data Frame")

narzio <- ldply(narzio\_dat, data.frame)

narzio$Subject <- rep(1:length(narzio\_name), each=100)

narzio$Type <- rep("Narzio", nrow(narzio))

narzio$Time <- rep(1:100, length=nrow(narzio))

})

return(narzio)

})

control\_name <- eventReactive(input$control,{

data1 <- input$control

validate(need(!is.null(input$control), 'Plaese Upload Control.zip'))

control <- unzip(data1$datapath, list=TRUE, exdir=getwd())

return(control)

})

output$control\_list <- renderTable({control\_name()})

control\_df <- eventReactive(input$control,{

withProgress(message="Merging Control Data", value=0, {

incProgress(1/4, detail="Unzip file")

data1 <- input$control

control\_name <- unzip(data1$datapath, list=TRUE, exdir=getwd())$Name

unzip(data1$datapath, files=control\_name, exdir=getwd(), overwrite=TRUE)

control\_path <- paste(getwd(), control\_name, sep='/')

incProgress(1/4, detail="Loading Unzipped Files")

control\_dat <- lapply(control\_path, function(x) read\_xlsx(x, range=cell\_rows(24:124))[,c(1:9, 13:21,40:87)] )

incProgress(1/4, detail="Bulid Data Frame")

control <- ldply(control\_dat, data.frame)

control$Subject <- rep(1:length(control\_name), each=100)

control$Type <- rep("Control", nrow(control))

control$Time <- rep(1:100, length=nrow(control))

})

return(control)

})

col\_list <- eventReactive(input$narzio, {

colnames <- colnames(narzio\_df())

return(colnames)

})

output$choose\_columns <- renderUI({

selectInput("columns", "Choose Variables", col\_list())

})

tot.dat <- eventReactive(input$button1, {

variable <- input$columns

narzio <- narzio\_df() %>% select("Type", "Subject", "Time", variable)

control <- control\_df() %>% select("Type", "Subject", "Time", variable)

total\_data <- rbind(narzio, control)

colnames(total\_data) <- c("Type", "Subject", "Time", "Variable")

total\_data <- as.data.frame(total\_data)

#standard <- function(x) (x-min(x))/(max(x)-min(x))

#total\_data <- total\_data %>% group\_by\_("Subject") %>% mutate\_(Variable=interp(~standard(x), x=as.name("Variable")))

return(total\_data)

})

plot1 <- eventReactive(input$button1,{

alpha <- as.numeric(as.character(input$alpha))

p <- tot.dat() %>% group\_by\_("Type", "Time") %>%

summarise\_(aver=interp(~mean(x), x=as.name("Variable")),

lower=interp(~mean(x)-qt(1-alpha/2, 14)\*sd(x), x=as.name("Variable")),

upper=interp(~mean(x)+qt(1-alpha/2, 14)\*sd(x), x=as.name("Variable"))) %>%

ungroup() %>% ggplot() + geom\_line(aes(x=Time, y=aver, color=Type)) + theme\_classic() +

scale\_color\_manual(values=c("red", "blue")) + xlab("Time") + ylab(input$columns) + ggtitle("Difference between Narzio and Control")

if(input$ci=="Yes"){

p <- p+geom\_ribbon(aes(x=Time, ymin=lower, ymax=upper, fill=Type), alpha=0.2) + scale\_fill\_manual(values=c("red", "blue"))

}

return(p)

})

output$plot1 <- renderPlot({plot1()})

output$plot2 <- renderPlot({

tot.dat() %>% mutate(Subject=factor(Subject)) %>% ggplot(aes(x=Time, y=Variable, color=Subject)) +

geom\_line() + facet\_grid(~Type) + theme\_classic() + xlab("Time") + ylab(input$columns) + ggtitle(paste(input$columns, "Level by Subject"))

})

output$plot3 <- renderPlot({

tot.dat() %>% mutate(Subject=factor(Subject)) %>% ggplot(aes(x=Time, y=Variable, color=Type)) + geom\_line() +

facet\_wrap(~Subject, scale='free') + theme\_classic() + scale\_fill\_manual(values=c("red", "blue")) +

xlab("Time") + ylab(input$columns) + ggtitle(paste(input$columns, "Level for Each Subject"))

})

test.dat <- eventReactive(input$button1, {

variable <- input$columns

narzio <- narzio\_df() %>% select("Type", "Subject", "Time", variable)

control <- control\_df() %>% select("Type", "Subject", "Time", variable)

total\_data <- rbind(narzio, control)

colnames(total\_data) <- c("Type", "Subject", "Time", "Variable")

total\_data <- as.data.frame(total\_data)

f <- function(x) {sum(x)-0.5\*(x[1]+x[length(x)])}

f2 <- function(x) {x[2]-x[1]}

test\_dat <- total\_data %>% group\_by\_("Subject", "Type") %>%

summarise\_(underarea = interp(~f(x), x=as.name("Variable"))) %>% ungroup() %>%

group\_by\_("Subject") %>% arrange(Subject, Type) %>% summarise\_(area2 = interp(~f2(x), x=as.name("underarea")))

return(test\_dat)

})

output$text1 <- renderPrint({

variable <- test.dat() %>% ungroup() %>% select(area2) %>% unlist

print(t.test(variable))

})}

shinyApp(ui = ui, server = server)