Statistical Round Session: R code - 2023 KoreAnesthesia

KJA Statistical Round Members

2023-11-11

Table of Contents

# 1 SESSION 1: R SOFTWARE (PREPARE FOR A LONG JOURNEY)

## 1.1 Session 1-1: R software, RStudio

## 1.2 Session 1-2: Understand R: CSV file format, Packages, List, data frame, object

### 1.2.1 Data Preparation

Set working directory: The directory where the sample\_data exists.

setwd(r"(Your\WorkingDirectory\Here)")

Import CSV format data.

#Read CSV file  
mydata = read.csv("sample\_data\_23workshop4.csv", header = TRUE, fileEncoding = "UTF-8-BOM", fill = TRUE)

### 1.2.2 Installing a package

R package의 종류  
1. Base Packages: R software 설치시 자동으로 설치되는 package들. R 시작시 함께 load되기 때문에 포함된 명령어를 그대로 사용하면 된다. (예: base, datasets)  
2. Recommended Packages: R software 설치시 자동으로 설치되나, R 시작시 load되지 않기 때문에 사용하려면R로 불러오기를 해야 한다. (예: MASS)  
3. Other Packages: 필요에 따라 특정 Package를 설치한 뒤, 불러와서 사용한다. (예: dplyr, ggplot2)

Package의 설치 (대소문자 주의)

install.packages("PackageName")

Package를 load하는 방법

library("PackageName")

예를 들어, 간단한 power analysis를 가능하게 해주는 “pwr” package를 설치해보자.

install.packages("pwr")

“pwr” package를 불러와 보자

library(pwr)

### 1.2.3 Vector, data frame, List

R이 자료를 처리하기 위해 저장하는 구조는 여러가지가 있다. 보통 Excel을 통해 자료를 저장하여 통계 프로그램에서 사용하는 것이 익숙하기 때문에, 이런 구조가 생소하겠지만, R은 이런 구조들을 사용하여 결과를 출력하여 저장해 두기 때문에, 특정 결과만 추출하여 다른 통계에 이용하기 위해서는 알아 둘 필요가 있다. 하나의 값에서 부터 vector, data frame, list등 R에서 다루는 자료를 object라고 한다.

1. vector  
   한가지 변수 형태들이 나열되어 저장된 형태이다.

name <- c("a", "b", "c", "d")  
  
value <- c(1, 2, 3, 4)  
  
  
name

## [1] "a" "b" "c" "d"

value

## [1] 1 2 3 4

1. data frame  
   Excel에서 자료를 저장하는 형태라고 이해하면 쉽다.  
   행과 열로 구성되어 있으며, 1행이나 1열로 이루어진 자료도 구성이 가능하다.  
   Excel과의 차이는 행이름 (row name)이나 열 이름 (column name)을 따로 가지고 있다는 점이다. (Excel은 보통 1행에 열 이름 - 흔히 변수이름 - 을 저장하고, 1열에는 자료 번호를 저장한다)  
   일반적으로 자료를 불러와서 통계분석을 시행할 때는 data frame형태의 자료를 이용하게 된다.

myexample <- data.frame(name, value)  
  
myexample

## name value  
## 1 a 1  
## 2 b 2  
## 3 c 3  
## 4 d 4

1. List  
   List는 여러개의 vector, data frame을 모두 포함할 수 있으며, 여기에 설명되지 않은 다른 형태의 자료 구조도 모두 포함할 수 있다. 대부분의 통계 결과는 List 형태로 출력이 되며, 이를 지정하여 특정 값 (예를 들면, 평균이나 P값)을 추출 해 낼 수 있다.

mylist <- list(session = "statistical round session",   
 person = 30,  
 seat = c(10, 20, 30))  
mylist

## $session  
## [1] "statistical round session"  
##   
## $person  
## [1] 30  
##   
## $seat  
## [1] 10 20 30

mylist$session

## [1] "statistical round session"

mylist$person

## [1] 30

mylist$seat

## [1] 10 20 30

mylist$seat[2]

## [1] 20

## 1.3 Session 1-3: R commend structure

R의 통계분석은 직접 필요한 명령어를 이용하여 coding을 해야 한다다.  
Coding에 필요한 모든 문법을 외워서 시행하는 것이 가장 이상적이겠지만, 명령어 각각이 다른 형식의 coding을 요구하는 경우가 흔하기 때문에, 기존에 구성되어 있는 명령어 예제를 적용하려는 자료에 맞추어 변형만 하여도 대부분의 통계 분석이 가능하다.  
이정도의 coding을 편집하기 위해서는 기본적으로 대부분의 명령어가 어떤 형태로 이루어져 있는지 이해하는 것이 필요하다.

1. Assignment  
   변수에 값을 할당하는 방법이다. 기호 = 혹은, <-, ->를 사용한다.
2. Function call  
   함수를 호출하여 작업을 수행하고 object에 할당하는 방법이다.

result <- mean(c(5, 8, 12))  
  
result

## [1] 8.333333

1. Conditional statements  
   Excel에서 if명령어를 사용하는 것과 유사하다.

if (x > 0){  
 print("Postive")  
} else if (x == 0){  
 print("zero")  
} else {  
 print("negative")  
}

1. Loop  
   반복하여 작업이 필요할 때 사용하는 방법이다.  
   for 혹은 while과 같은 반복문을 사용할 수 있다.

for(i in 1:5){  
 print(i)  
}

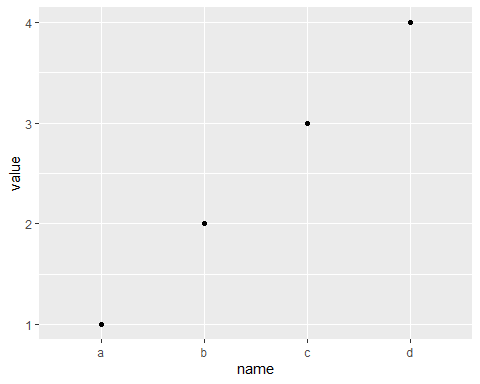
## [1] 1  
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5

1. Data structure  
   앞서 살펴본 방법들로 자료를 저장할 수 있다.
2. Package usage  
   R에서 기본적으로 제공되지 않는 기능은 Package를 설치하여 확장 할 수 있다. 이들은 package 설치 후 해당 package를 loading해야 사용이 가능하다.  
   특정 명령어를 이용하여 통계분석을 시행할 때는, 어떤 방식으로 계산을 해야 하는지 지정을 해줘야 하며, 이것들은 argument를 통해 가능하다. 다만 package마다 argument를 설정하는 방식이 조금씩 다르기 때문에, 기존에 완성된 code를 복사하여 자신의 상황에 맞도록 수정하는 것이 편리하다.  
   예를 들어, 그래프를 그리기 위해 자주 사용되는 package는 ggplot2라는 것이 있으며며, 이를 이용하여 산점도를 그린다면 다음과 같은 명령어 형식을 사용할 수 있다.

#intalling "ggplot2"  
if (!requireNamespace("ggplot2", quietly = TRUE)) {  
 install.packages("ggplot2")  
}

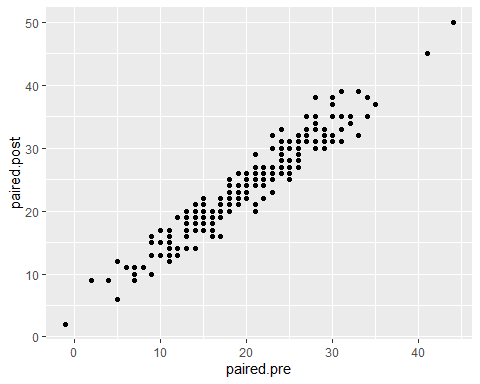
이제 ggplot2를 load하여 간단한 그래프를 그려보자.

library(ggplot2)  
  
ggplot(data = myexample, aes(x = name, y = value)) + geom\_point()



얼핏 보면 생소하고 어려운 듯 하나, 자세히 살펴보면 x에 x축에 해당되는 변수명을, y에 y축에 해당하는 변수명을 넣어주면 다른 자료들의 scatter plot을 그릴수 있다. sample csv file에는 여러개의 변수들이 있으며, 이들 중 “vas1”과 “vas2”사이의 산점도 (scatter plot)을 그린다면 다음과 같은 명령어를 사용하면 된다.

ggplot(data = mydata, aes(x = paired.pre, y = paired.post)) + geom\_point()



ggplot2를 포함하여 R에서 통계분석을 할때, 많은 도움이 되는 몇가지 package들을 소개한다.

1. ggplot2  
   앞서 소개한 바와 같이 그래프를 작성할때 매우 유용한 package이다.
2. dplyr  
   R은 한번에 한가지씩 처리를 하는 방식으로 coding이 된다. 예를 들어 자료에서 특정 조건에 맞는 경우들을 골라내서, 평균을 계산하고 새로운 data로 저장하는 작업을 R에서 수행하려면, 조건에 맞게 추출한 자료를 저장하고, 그 자료를 불러와서 평균을 계산해서 그 값들을 저장하고, 저장된 값을 data형태로 다시 저장을 해야 한다. 이런 일련의 과정을 연속적으로 수행하게 해주는 package가 dplyr이다.
3. tidyverse  
   tidyverse는 dplyr과 ggplot2를 포함하고 tidyr, readr, purrr, tibble, stringr, forcats와 같은 여러개의 package를 모아서, 통계분석 작업을 더 효율적이고 일관된 방식으로 수행할 수 있도록도와준다. 다만, 이와 같은 package들을 모두 load해서 사용하게 되면 일부 저성능 컴퓨터에서는 처리 속도가 지연될 수 있기 때문에, 필요한 package를 load하거나 잠시 불러와서 사용하는 방법도 고려해야 한다.

Package를이용하는 두가지 방법  
첫번째는 library()를 이용하여 컴퓨터의 메모리에 전체 package를 상주시킨 뒤 필요한 명령어를 사용하는 방법이다. 이방법은 한번 package를 load하면 언제든지 포함된 명령어를 사용할 수 있어 편리하지만, 저사양 컴퓨터나 아주 복잡한통계 분석을 시행할 때는 컴퓨터의 메모리를 차지하기 때문에 불리할 수 있다.  
두번째는 package::command형태로 해당 명령어를 실행할때만 package를 지정하여 시행하는 방법이다. coding시 입력을 매번 해야 하는 번거로움이 있지만, 메모리를 아낄 수 있는 장점이 있다. 전체 분석에 자주 사용하는 경우가 아니라면 추천할 만 하다.

# 2 SESSION 2: DEPARTURE TO CODING FOR A STATISTICAL ANALYSIS

## 2.1 Session 2-1: Basic concepts of statistical analysis

## 2.2 Session 2-2: Descriptive statistics, Normality test, Equal variance test, Basic data explorations

Session 1에서 불러온 mydata의 구조를 살펴보자.  
mydata를 입력하면 불러온 자료의 일부를 보여준다.

mydata

자료의 일부분만 살펴보기 위해선 다음과 같은 방법을 사용할 수 있다.  
결과 출력의 길이가 길기 때문에, 결과는 생략한다.

# Show the first 6 rows   
head(mydata)  
  
# Show the first 3 rows  
head(mydata, n=3)  
  
# Show the last 6 rows  
tail(mydata)  
  
# Show 4 - 6 rows of the data  
mydata[4:6,]  
  
# Show 3~5 columns of the data  
mydata[,3:5]

mydata의 구조를 확인하는 방법은 다음과 같다.  
자료가 몇개의 행과 열로 이루어져 있는지 확인하는 방법.

# dimension  
dim(mydata)

## [1] 216 12

변수의 개수를 확인하는 방법.

# No of columns (No of variables)  
ncol(mydata)

## [1] 12

# No of variables (length of the first row)  
length(mydata)

## [1] 12

입력된 개수 (표본수)를 확인하는 방법.

# No of rows (No of cases: excluding row of variable names)  
nrow(mydata)

## [1] 216

# length of a specific variable - case.no vs. mw.group  
length(mydata$case.no)

## [1] 216

변수들의 특성을 확인하는 방법.  
(int: integer, chr: character, num: number)

# the characters of variables  
str(mydata)

## 'data.frame': 216 obs. of 12 variables:  
## $ case.no : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ group : int 1 2 1 1 1 2 1 2 2 2 ...  
## $ t\_test.value: int 70 48 77 64 80 57 77 47 58 52 ...  
## $ paired.pre : int 15 10 30 23 2 19 30 25 11 29 ...  
## $ paired.post : int 20 17 37 25 9 22 35 28 12 33 ...  
## $ chisqr : int 0 0 0 1 1 1 0 0 0 1 ...  
## $ medication : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ observed : int 387 505 612 584 454 530 627 437 501 601 ...  
## $ obs1 : int 132 106 130 132 119 142 100 170 89 118 ...  
## $ obs2 : int 148 121 145 147 134 157 114 185 104 135 ...  
## $ obs3 : int 163 136 159 163 149 172 128 202 117 149 ...  
## $ obs4 : int 178 151 174 178 166 186 144 218 132 165 ...

변수의 descriptive statistics와 missing value개수를 확인하는 방법.

# Summarized values   
summary(mydata)

## case.no group t\_test.value paired.pre paired.post   
## Min. : 1.00 Min. :1.0 Min. :35.00 Min. :-1.00 Min. : 2.00   
## 1st Qu.: 54.75 1st Qu.:1.0 1st Qu.:56.00 1st Qu.:15.00 1st Qu.:19.00   
## Median :108.50 Median :1.5 Median :63.00 Median :20.00 Median :24.00   
## Mean :108.50 Mean :1.5 Mean :64.31 Mean :20.25 Mean :24.12   
## 3rd Qu.:162.25 3rd Qu.:2.0 3rd Qu.:71.00 3rd Qu.:25.25 3rd Qu.:30.00   
## Max. :216.00 Max. :2.0 Max. :95.00 Max. :44.00 Max. :50.00   
## chisqr medication observed obs1 obs2   
## Min. :0.0000 Min. :1 Min. :375.0 Min. : 17.0 Min. : 31.0   
## 1st Qu.:0.0000 1st Qu.:1 1st Qu.:498.0 1st Qu.: 83.0 1st Qu.: 99.0   
## Median :0.0000 Median :2 Median :536.0 Median :103.0 Median :118.0   
## Mean :0.4815 Mean :2 Mean :543.3 Mean :102.7 Mean :117.7   
## 3rd Qu.:1.0000 3rd Qu.:3 3rd Qu.:589.0 3rd Qu.:123.2 3rd Qu.:138.5   
## Max. :1.0000 Max. :3 Max. :725.0 Max. :197.0 Max. :211.0   
## obs3 obs4   
## Min. : 45.0 Min. : 61.0   
## 1st Qu.:113.8 1st Qu.:129.0   
## Median :133.5 Median :147.0   
## Mean :132.7 Mean :147.7   
## 3rd Qu.:154.0 3rd Qu.:168.2   
## Max. :226.0 Max. :243.0

변수명을 확인하는 방법

# list of variables  
names(mydata)

## [1] "case.no" "group" "t\_test.value" "paired.pre" "paired.post"   
## [6] "chisqr" "medication" "observed" "obs1" "obs2"   
## [11] "obs3" "obs4"

# Metadata of the imported data  
attributes(mydata)

## $names  
## [1] "case.no" "group" "t\_test.value" "paired.pre" "paired.post"   
## [6] "chisqr" "medication" "observed" "obs1" "obs2"   
## [11] "obs3" "obs4"   
##   
## $class  
## [1] "data.frame"  
##   
## $row.names  
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18  
## [19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36  
## [37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54  
## [55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72  
## [73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90  
## [91] 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108  
## [109] 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126  
## [127] 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144  
## [145] 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162  
## [163] 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180  
## [181] 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198  
## [199] 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216

### 2.2.1 Descriptive statistics and distribution

자료의 형태를 확인하는 방법은, str을 사용하는 것도 가능하지만, skim이라는 package를 사용하면 여러가지의 정보를 한번에 할 수 있다. skim은 기본적으로 dplyr을 이용하기 때문에 두개의 package가 모두 설치되어 있어야 한다. 다음은 dplyr과 skimr이 설치되어 있지 않은 경우 설치를 진행해주는 code이다.

if (!requireNamespace("dplyr", quietly = TRUE)) {  
 install.packages("dplyr")  
}  
  
if (!requireNamespace("skimr", quietly = TRUE)) {  
 install.packages("skimr")  
}

skimr에 포함되어 있는 명령어 skim을 이용하면 다음과 같은 summary statistics를 얻을수 있다.

skimr::skim(mydata)

Data summary

|  |  |
| --- | --- |
| Name | mydata |
| Number of rows | 216 |
| Number of columns | 12 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| numeric | 12 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| case.no | 0 | 1 | 108.50 | 62.50 | 1 | 54.75 | 108.5 | 162.25 | 216 | ▇▇▇▇▇ |
| group | 0 | 1 | 1.50 | 0.50 | 1 | 1.00 | 1.5 | 2.00 | 2 | ▇▁▁▁▇ |
| t\_test.value | 0 | 1 | 64.31 | 11.31 | 35 | 56.00 | 63.0 | 71.00 | 95 | ▁▆▇▃▁ |
| paired.pre | 0 | 1 | 20.25 | 7.35 | -1 | 15.00 | 20.0 | 25.25 | 44 | ▁▅▇▃▁ |
| paired.post | 0 | 1 | 24.12 | 7.53 | 2 | 19.00 | 24.0 | 30.00 | 50 | ▁▆▇▃▁ |
| chisqr | 0 | 1 | 0.48 | 0.50 | 0 | 0.00 | 0.0 | 1.00 | 1 | ▇▁▁▁▇ |
| medication | 0 | 1 | 2.00 | 0.82 | 1 | 1.00 | 2.0 | 3.00 | 3 | ▇▁▇▁▇ |
| observed | 0 | 1 | 543.32 | 67.04 | 375 | 498.00 | 536.0 | 589.00 | 725 | ▂▆▇▅▁ |
| obs1 | 0 | 1 | 102.71 | 30.23 | 17 | 83.00 | 103.0 | 123.25 | 197 | ▁▆▇▅▁ |
| obs2 | 0 | 1 | 117.69 | 30.28 | 31 | 99.00 | 118.0 | 138.50 | 211 | ▁▆▇▅▁ |
| obs3 | 0 | 1 | 132.72 | 30.39 | 45 | 113.75 | 133.5 | 154.00 | 226 | ▁▆▇▅▁ |
| obs4 | 0 | 1 | 147.73 | 30.48 | 61 | 129.00 | 147.0 | 168.25 | 243 | ▁▅▇▃▁ |

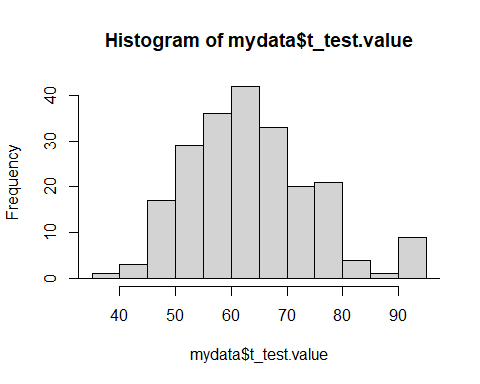
군 별로 나누어져 있는 자료의 descriptive statistics는 psych라는 package의 desribeBy명령어를 사용하거나 dplyr의 filter 명령어로 구할 수 있다. 다음은 psych package를 이용한 방법이다.

if (!requireNamespace("psych", quietly = TRUE)) {  
 install.packages("psych")  
}  
  
psych::describeBy(mydata$t\_test.value, group = mydata$group )

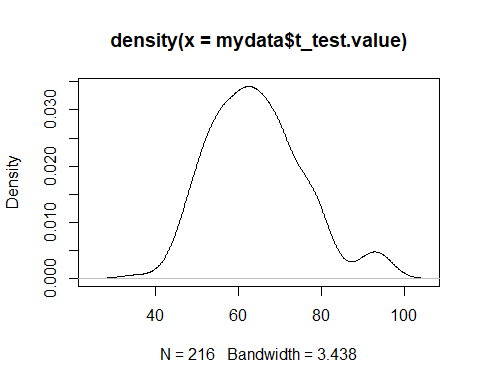
##   
## Descriptive statistics by group   
## group: 1  
## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 108 70.2 11.28 70 69.77 10.38 43 95 52 0.18 -0.01 1.09  
## ------------------------------------------------------------   
## group: 2  
## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 108 58.43 7.75 58 58.2 8.9 35 80 45 0.16 -0.01 0.75

Histogram을 확인하는 방법은 다음과 같다.

hist(mydata$t\_test.value)  
lines(density(mydata$t\_test.value))

 위 그래프에서 density가 잘 나오지 않는다. density plot만 확인하려면 다음의 코드를 이용한다.

plot(density(mydata$t\_test.value))



### 2.2.2 Normality test

R에서 기본으로 제공하는 정규성 검정 방법은 Shapiro-Wilk test (shapiro.test()), Kolmogrorov-Smirnov test (ks.test()), Anderson-Darling test (ad.test()), Lilliefors test (lillie.test()), Jarque-Bera test(jarque.bera.test())가 있다.  
다음의 code를 이용하면 숫자형변수들의 정규성 검정을 한번에 실행 할 수 있다.

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

mydata %>% # "%>%"는 dplyr의 연속적인 계산을 명령하는 기호,Pipe operator.  
 dplyr::select(t\_test.value,  
 paired.pre, # 숫자형변수만 선택하는 dplyr을 이용한 작업이다.   
 paired.post, # 알아보기 쉽게 줄바꿈을 하면,   
 observed  
 ) %>%  
 apply(2, shapiro.test) -> normality.result   
   
   
normality.result

## $t\_test.value  
##   
## Shapiro-Wilk normality test  
##   
## data: newX[, i]  
## W = 0.97434, p-value = 0.0005661  
##   
##   
## $paired.pre  
##   
## Shapiro-Wilk normality test  
##   
## data: newX[, i]  
## W = 0.99374, p-value = 0.5016  
##   
##   
## $paired.post  
##   
## Shapiro-Wilk normality test  
##   
## data: newX[, i]  
## W = 0.99329, p-value = 0.4379  
##   
##   
## $observed  
##   
## Shapiro-Wilk normality test  
##   
## data: newX[, i]  
## W = 0.99266, p-value = 0.3575

위의 code는apply라는 명령어를 사용하였다.  
apply의 argument는 다음과 같다.  
apply(data, margin, function)  
data: pipeline operator를 이용하였기 때문에 data는 지정하지 않아도 이전 단계의 자료를 사용한다.  
margin: 1은 행단위의 자료를, 2는 열단위로 자료를 이용한다고 지정한다.  
function: 시행할 명령어를 넣어준다.

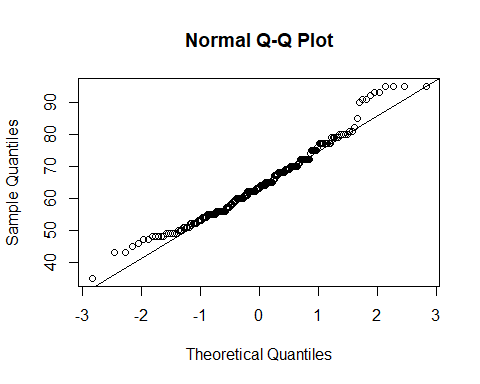
rstatix package에서 제공하는 shapiro\_test()는 기본 명령어인 shapiro.test를 이용하여 dplyr의 pipeline operator를 쉽게 사용할 수 있게 해주며, grouped data를 이용 할 수 있다.

if (!requireNamespace("rstatix", quietly = TRUE)) {  
 install.packages("rstatix")  
}  
  
mydata %>%  
 group\_by(group) %>%  
 rstatix::shapiro\_test(t\_test.value)

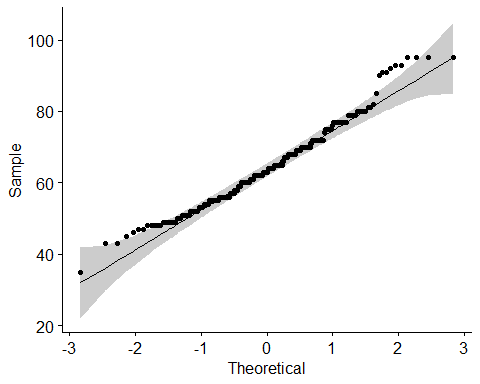
## # A tibble: 2 × 4  
## group variable statistic p  
## <int> <chr> <dbl> <dbl>  
## 1 1 t\_test.value 0.977 0.0586  
## 2 2 t\_test.value 0.987 0.391

R에서 QQ plot은 기본 함수로 그릴 수 있다.

qqnorm(mydata$t\_test.value)  
qqline(mydata$t\_test.value)

 ggpubr package에서 제공하는 ggqqplot를 이용하면 해석이 조금 더 용이하다.

if (!requireNamespace("ggpubr", quietly = TRUE)) {  
 install.packages("ggpubr")  
}  
ggpubr::ggqqplot(mydata$t\_test.value)



### 2.2.3 Equal variance test (Homogeneity of variance test)

R에서 기본 제공하는 equal variance test는 Bartlett test와 Fligner test이며, SPSS등에서 사용되고 있는 더 robust한 방법인 Leven’s test는 “car” package를 설치하거나, 기본으로 설치되는 DescTools를 이용하여여 사용할 수 있다.

# For these test, grouping variable should not be a quantitative variable.  
mydata$group <- as.factor(mydata$group)  
  
# Bartlett test  
bartlett.test(t\_test.value ~ group, data = mydata)

##   
## Bartlett test of homogeneity of variances  
##   
## data: t\_test.value by group  
## Bartlett's K-squared = 14.629, df = 1, p-value = 0.0001309

# Fligner test  
fligner.test(t\_test.value ~ group, data = mydata)

##   
## Fligner-Killeen test of homogeneity of variances  
##   
## data: t\_test.value by group  
## Fligner-Killeen:med chi-squared = 7.4018, df = 1, p-value = 0.006516

# Levene's test  
if (!requireNamespace("car", quietly = TRUE)) {  
 install.packages("car")  
}  
  
car::leveneTest(t\_test.value ~ group, data = mydata)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)   
## group 1 8.8181 0.003323 \*\*  
## 214   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

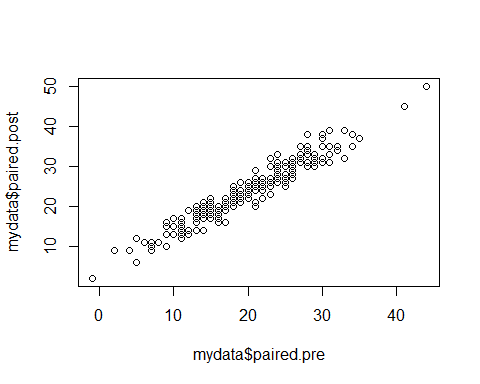
DescTools::LeveneTest(t\_test.value ~ group, data = mydata)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)   
## group 1 8.8181 0.003323 \*\*  
## 214   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

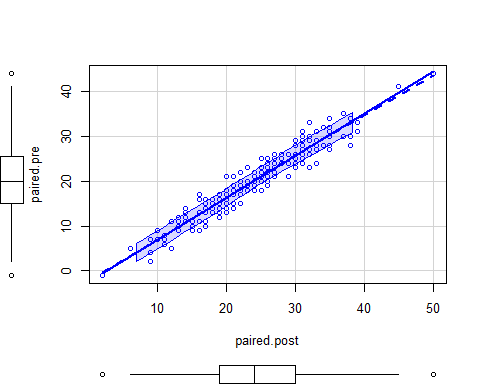
### 2.2.4 Crude data exploration

변수들간의 관계에 대한 정보는 이후 통계 분석 과정에 많은 도움을 준다. 변수들간의 관계를 파악하기 쉬운 방법은 산점도(scatter plot)을 이용하거나 correlation analysis를 이용한다.  
R package들 중 이런 정보를 얻기 쉽게 해주는 명령어를 제공하는 몇가지를 소개한다.

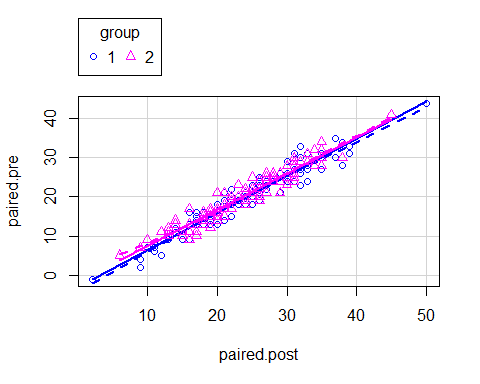
# Basic scatter plot between two variables  
plot(mydata$paired.pre, mydata$paired.post)



# advanced scatter plot   
car::scatterplot(paired.pre ~ paired.post, data = mydata)



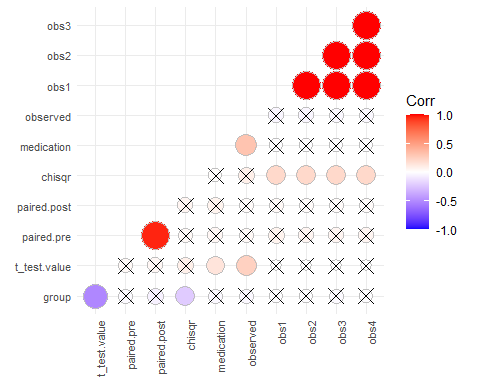
# Scatter plot by groups  
car::scatterplot(paired.pre ~ paired.post|group, data = mydata)



변수들 간의 관계를 알아보기 위해 correlation analysis를 이용할 수 있지만, 전체 자료에서의 관계를 한눈에 파악하기 위해 correlation plot을 이용하는 방법을소개한다.

correlation plot

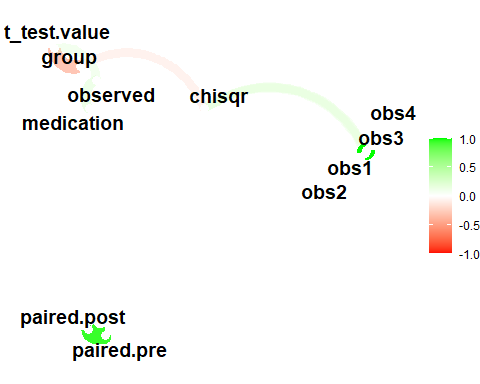
mydata$group <- as.numeric(mydata$group)  
  
ggcorrplot::ggcorrplot(cor(mydata[,2:12]),  
 p.mat = ggcorrplot::cor\_pmat(mydata[,2:12]),  
 type = "lower",  
 method = "circle",  
 tl.cex = 8,  
 tl.srt = 90)



if (!requireNamespace("tidyverse", quietly = TRUE)) {  
 install.packages("tidyverse")  
}  
if (!requireNamespace("corrr", quietly = TRUE)) {  
 install.packages("corrr")  
}  
if (!requireNamespace("PerformanceAnalytics", quietly = TRUE)) {  
 install.packages("PerformanceAnalytics")  
}  
  
library(tidyverse)  
library(corrr)  
library(PerformanceAnalytics)

mydata[,2:12] %>%  
 correlate(method = "pearson") %>%  
 network\_plot(min\_cor = 0.2,  
 colors = c("red", "white", "green"),  
 repel = TRUE,  
 curved = TRUE)

## Correlation computed with  
## • Method: 'pearson'  
## • Missing treated using: 'pairwise.complete.obs'



## 2.3 Session 2-3: T-test (independent, paired), chi-square test, one-way ANOVA

### 2.3.1 Independnet T-test

T-test를 위한 자료는 t\_test.group과 t\_test.value이다.

1. Normality test

mydata %>%  
 filter(group == 1) %>%  
 select(t\_test.value) %>%  
 apply(2, shapiro.test)

## $t\_test.value  
##   
## Shapiro-Wilk normality test  
##   
## data: newX[, i]  
## W = 0.97705, p-value = 0.05857

mydata %>%  
 filter(group == 2) %>%  
 select(t\_test.value) %>%  
 apply(2, shapiro.test)

## $t\_test.value  
##   
## Shapiro-Wilk normality test  
##   
## data: newX[, i]  
## W = 0.98714, p-value = 0.3907

1. Homogeneity of variance test

mydata$group <- as.factor(mydata$group)  
  
car::leveneTest(t\_test.value ~ group, data = mydata)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)   
## group 1 8.8181 0.003323 \*\*  
## 214   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

1. t-test

등분산성이 확립되었을 때

t.test(t\_test.value ~ group,  
 data = mydata,  
 alternative = "two.sided",  
 paired = FALSE,  
 var.equal = TRUE  
 ) -> t.test.res  
  
t.test.res

##   
## Two Sample t-test  
##   
## data: t\_test.value by group  
## t = 8.9451, df = 214, p-value < 2.2e-16  
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0  
## 95 percent confidence interval:  
## 9.182461 14.373095  
## sample estimates:  
## mean in group 1 mean in group 2   
## 70.20370 58.42593

# Effect size: Cohen's d  
if (!requireNamespace("effectsize", quietly = TRUE)) {  
 install.packages("effectsize")  
}  
  
effectsize::cohens\_d(t\_test.value ~ group,   
 data = mydata) -> cohend  
  
cohend

## Cohen's d | 95% CI  
## ------------------------  
## 1.22 | [0.93, 1.51]  
##   
## - Estimated using pooled SD.

등분산성이 확립되지 않았을 때

t.test(t\_test.value ~ group,  
 data = mydata,  
 alternative = "two.sided",  
 paired = FALSE,  
 var.equal = FALSE  
 ) -> t.test.ueq.res  
  
t.test.ueq.res

##   
## Welch Two Sample t-test  
##   
## data: t\_test.value by group  
## t = 8.9451, df = 189.66, p-value = 3.349e-16  
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0  
## 95 percent confidence interval:  
## 9.180565 14.374990  
## sample estimates:  
## mean in group 1 mean in group 2   
## 70.20370 58.42593

# Effect size: Cohen's d  
effectsize::cohens\_d(t\_test.value ~ group,   
 data = mydata,  
 pooled\_sd = FALSE # Default value is "TRUE"  
 ) -> cohend.ueq  
  
cohend.ueq

## Cohen's d | 95% CI  
## ------------------------  
## 1.22 | [0.92, 1.51]  
##   
## - Estimated using un-pooled SD.

### 2.3.2 Paired T-test

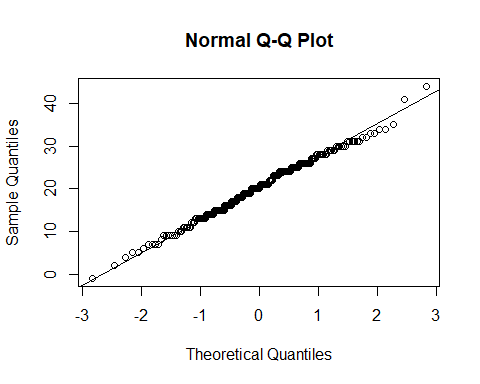
Paired t-test data: pair\_t.pre pair\_t.post

1. normality test

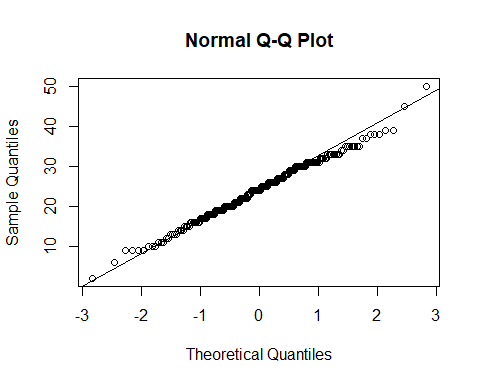
mydata %>%  
 select(paired.pre, paired.post) %>%  
 apply(2, shapiro.test)

## $paired.pre  
##   
## Shapiro-Wilk normality test  
##   
## data: newX[, i]  
## W = 0.99374, p-value = 0.5016  
##   
##   
## $paired.post  
##   
## Shapiro-Wilk normality test  
##   
## data: newX[, i]  
## W = 0.99329, p-value = 0.4379

qqnorm(mydata$paired.pre)  
qqline(mydata$paired.pre)



qqnorm(mydata$paired.post)  
qqline(mydata$paired.post)

 2. paired t-test

t.test(mydata$paired.pre, mydata$paired.post,  
 alternative = "two.sided",  
 paired = TRUE  
 ) -> t.test.pair.res  
  
t.test.pair.res

##   
## Paired t-test  
##   
## data: mydata$paired.pre and mydata$paired.post  
## t = -26.901, df = 215, p-value < 2.2e-16  
## alternative hypothesis: true mean difference is not equal to 0  
## 95 percent confidence interval:  
## -4.158928 -3.591072  
## sample estimates:  
## mean difference   
## -3.875

# Effect size  
effectsize::cohens\_d(mydata$paired.pre, mydata$paired.post,  
 paired = TRUE) -> cohend.pair  
  
cohend.pair

## Cohen's d | 95% CI  
## --------------------------  
## -1.83 | [-2.37, -1.61]

# the following also works  
effectsize::cohens\_d(Pair(paired.pre, paired.post) ~ 1, data = mydata)

### 2.3.3 Chi-square test

Chi-square test를 위한 자료: group, chisqr

Table 만드는 방법 1

chi.tbl <- xtabs(~group + chisqr, mydata)  
addmargins(chi.tbl)

## chisqr  
## group 0 1 Sum  
## 1 44 64 108  
## 2 68 40 108  
## Sum 112 104 216

Table 만드는 방법 2

if (!requireNamespace("gmodels", quietly = TRUE)) {  
 install.packages("gmodels")  
}

## Registered S3 method overwritten by 'gdata':  
## method from   
## reorder.factor DescTools

gmodels::CrossTable(mydata$group, mydata$chisqr)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | Chi-square contribution |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 216   
##   
##   
## | mydata$chisqr   
## mydata$group | 0 | 1 | Row Total |   
## -------------|-----------|-----------|-----------|  
## 1 | 44 | 64 | 108 |   
## | 2.571 | 2.769 | |   
## | 0.407 | 0.593 | 0.500 |   
## | 0.393 | 0.615 | |   
## | 0.204 | 0.296 | |   
## -------------|-----------|-----------|-----------|  
## 2 | 68 | 40 | 108 |   
## | 2.571 | 2.769 | |   
## | 0.630 | 0.370 | 0.500 |   
## | 0.607 | 0.385 | |   
## | 0.315 | 0.185 | |   
## -------------|-----------|-----------|-----------|  
## Column Total | 112 | 104 | 216 |   
## | 0.519 | 0.481 | |   
## -------------|-----------|-----------|-----------|  
##   
##

Chi-square test (기본 package)

chisq.test(mydata$group, mydata$chisqr,  
 correct = TRUE) -> chisq.res  
  
chisq.res

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: mydata$group and mydata$chisqr  
## X-squared = 9.8098, df = 1, p-value = 0.001736

# effect size: Cramér's V  
effectsize::cramers\_v(mydata$group, mydata$chisqr) -> cramerv  
  
cramerv

## Cramer's V (adj.) | 95% CI  
## --------------------------------  
## 0.21 | [0.09, 1.00]  
##   
## - One-sided CIs: upper bound fixed at [1.00].

### 2.3.4 ANOVA

Normality test

mydata %>%  
 group\_by(medication) %>%  
 rstatix::shapiro\_test(observed)

## # A tibble: 3 × 4  
## medication variable statistic p  
## <int> <chr> <dbl> <dbl>  
## 1 1 observed 0.993 0.956  
## 2 2 observed 0.983 0.449  
## 3 3 observed 0.988 0.748

One-way ANOVA test

mydata$medication <- as.factor(mydata$medication)  
  
  
aov(observed ~ medication, data = mydata) -> one.anova.res  
  
summary(one.anova.res)

## Df Sum Sq Mean Sq F value Pr(>F)   
## medication 2 94450 47225 11.54 1.75e-05 \*\*\*  
## Residuals 213 871927 4094   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# effect size: eta squared  
effectsize::eta\_squared(one.anova.res) -> one.anova.eta

## For one-way between subjects designs, partial eta squared is equivalent  
## to eta squared. Returning eta squared.

one.anova.eta

## # Effect Size for ANOVA  
##   
## Parameter | Eta2 | 95% CI  
## --------------------------------  
## medication | 0.10 | [0.04, 1.00]  
##   
## - One-sided CIs: upper bound fixed at [1.00].

# effet size: Cohen's f  
effectsize::cohens\_f(one.anova.res) -> one.anova.f

## For one-way between subjects designs, partial eta squared is equivalent  
## to eta squared. Returning eta squared.

one.anova.f

## # Effect Size for ANOVA  
##   
## Parameter | Cohen's f | 95% CI  
## ------------------------------------  
## medication | 0.33 | [0.20, Inf]  
##   
## - One-sided CIs: upper bound fixed at [Inf].

Two-way ANOVA test

mydata$group <- as.factor(mydata$group)  
  
aov(observed ~ medication + group + medication\*group, data = mydata) -> two.anova.res  
  
summary(two.anova.res)

## Df Sum Sq Mean Sq F value Pr(>F)   
## medication 2 94450 47225 11.604 1.66e-05 \*\*\*  
## group 1 271 271 0.066 0.797   
## medication:group 2 17007 8503 2.089 0.126   
## Residuals 210 854650 4070   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# effect size: eta squared (Type I & III)  
 # Type I  
effectsize::eta\_squared(two.anova.res) ->two.anova.eta.i  
  
two.anova.eta.i

## # Effect Size for ANOVA (Type I)  
##   
## Parameter | Eta2 (partial) | 95% CI  
## ------------------------------------------------  
## medication | 0.10 | [0.04, 1.00]  
## group | 3.16e-04 | [0.00, 1.00]  
## medication:group | 0.02 | [0.00, 1.00]  
##   
## - One-sided CIs: upper bound fixed at [1.00].

effectsize::cohens\_f(two.anova.res) -> two.anova.f.i  
  
two.anova.f.i

## # Effect Size for ANOVA (Type I)  
##   
## Parameter | Cohen's f (partial) | 95% CI  
## ----------------------------------------------------  
## medication | 0.33 | [0.21, Inf]  
## group | 0.02 | [0.00, Inf]  
## medication:group | 0.14 | [0.00, Inf]  
##   
## - One-sided CIs: upper bound fixed at [Inf].

# Type III  
two.anova.res.iii <- car::Anova(two.anova.res, type = 3) # Two-way ANOVA, type III result  
  
effectsize::eta\_squared(two.anova.res.iii) -> two.anova.eta.iii

## Type 3 ANOVAs only give sensible and informative results when covariates  
## are mean-centered and factors are coded with orthogonal contrasts (such  
## as those produced by `contr.sum`, `contr.poly`, or `contr.helmert`, but  
## \*not\* by the default `contr.treatment`).

two.anova.eta.iii

## # Effect Size for ANOVA (Type III)  
##   
## Parameter | Eta2 (partial) | 95% CI  
## ------------------------------------------------  
## medication | 0.10 | [0.04, 1.00]  
## group | 8.56e-03 | [0.00, 1.00]  
## medication:group | 0.02 | [0.00, 1.00]  
##   
## - One-sided CIs: upper bound fixed at [1.00].

effectsize::cohens\_f(two.anova.res.iii)

## Type 3 ANOVAs only give sensible and informative results when covariates  
## are mean-centered and factors are coded with orthogonal contrasts (such  
## as those produced by `contr.sum`, `contr.poly`, or `contr.helmert`, but  
## \*not\* by the default `contr.treatment`).

## # Effect Size for ANOVA (Type III)  
##   
## Parameter | Cohen's f (partial) | 95% CI  
## ----------------------------------------------------  
## medication | 0.33 | [0.21, Inf]  
## group | 0.09 | [0.00, Inf]  
## medication:group | 0.14 | [0.00, Inf]  
##   
## - One-sided CIs: upper bound fixed at [Inf].

effectsize::cohen\_f값은 실제 수기 계산값과 조금 다르다. 아마도, eta squared에서 계산한 값 같다.  
수기 계산은 다음의 식이다.

eta\_square와 f의 관계는 다음과 같다.

여기서 주의할 점은 two-way ANOVA table의 ss를 계산할때는 평균값의 차이 제곱합에 표본수를 곱해야 한다는 점이다. (reference: How to Perform a Two-Way ANOVA by Hand - Statology.pdf)  
Eta\_squared도 SPSS에서와 G\*Power에서 계산한게 다르듯이 패키지 마다 다를 수 있다. (reference: G\*Power manual.pdf, P27)

### 2.3.5 Repeated measures ANOVA

R의 aov 명령어를 이용하여 RM ANOVA를 시행하려면, 자료의 형태가 long format이어야 가능하다. 다음의 code는 long format으로 자료의 구조를 바꾸어 준다.

tidyr::gather(mydata,   
 ws.factor, # Variable of within-subject factor  
 observed, # observed value at each measurement point  
 obs1:obs4,  
 factor\_key = TRUE  
 ) -> mydata.long

변경된 자료의 구조를 살펴보면, 마지막에 within-subject factor (repeated measures factor)와 observed라는 변수가 생성된 것을 볼 수 있다. Within-subject factor는 obs1, obs2, obs3, obs4와 같이 측정시점의 정보가 담겨져 있고, observed는 해당 측정 시점에서의 측정값이 담겨있다. 즉, 첫번째 case의자료가 다음과 같이 4개의 자료로 바뀐다.

# 1st case wide type data  
mydata[1,]

## case.no group t\_test.value paired.pre paired.post chisqr medication observed  
## 1 1 1 70 15 20 0 1 387  
## obs1 obs2 obs3 obs4  
## 1 132 148 163 178

# 1st case long type data  
mydata.long[c(1,217, 433, 649),]

## case.no group t\_test.value paired.pre paired.post chisqr medication  
## 1 1 1 70 15 20 0 1  
## 217 1 1 70 15 20 0 1  
## 433 1 1 70 15 20 0 1  
## 649 1 1 70 15 20 0 1  
## ws.factor observed  
## 1 obs1 132  
## 217 obs2 148  
## 433 obs3 163  
## 649 obs4 178

RM ANOVA를 시행하기 위한 기본 가정 (basic assumptions)을 확인한다.  
1. Outliers check

mydata.long %>%  
 select(ws.factor, observed) %>%  
 group\_by(ws.factor) %>%  
 rstatix::identify\_outliers(observed) -> outliers  
  
summary(outliers)

## ws.factor observed is.outlier is.extreme   
## obs1:3 Min. : 17.0 Mode:logical Mode :logical   
## obs2:3 1st Qu.: 57.0 TRUE:12 FALSE:12   
## obs3:3 Median :198.5   
## obs4:3 Mean :155.1   
## 3rd Qu.:217.8   
## Max. :243.0

as.data.frame(outliers) %>%  
 filter(is.extreme == "TRUE")

## [1] ws.factor observed is.outlier is.extreme  
## <0 rows> (or 0-length row.names)

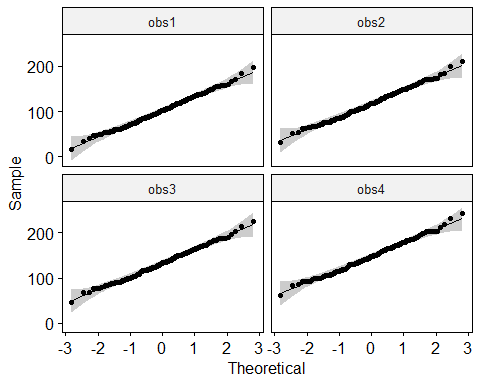
“is.extreme”에서 TRUE의 개수가 0개이므로 extreme outliers가 없음을 확인할 수 있다. 만약 extreme outlier가 있다면, 자료의 값이 잘못 입력되었는지 확인해보고, winsorization이 필요하다면, winsorization후 sensitivity analysis까지 계획을 해야 한다.

1. Normality test

mydata.long %>%  
 group\_by(ws.factor) %>%  
 rstatix::shapiro\_test(observed)

## # A tibble: 4 × 4  
## ws.factor variable statistic p  
## <fct> <chr> <dbl> <dbl>  
## 1 obs1 observed 0.996 0.894  
## 2 obs2 observed 0.996 0.863  
## 3 obs3 observed 0.996 0.880  
## 4 obs4 observed 0.996 0.821

ggpubr::ggqqplot(mydata.long, "observed", facet.by = "ws.factor")



1. Sphericity assumption & ANOVA test  
   rstatix의 anova\_test를 이용하면, Sphericity assumption의 과정과 결과에 따라 Greenhose-Geisser correction이나 Hyunh-Feldt correction등 보정의 작업을 한번에 수행해준다.

*one way RM ANOVA (ANOVA with one repeated measure factor)*

library(rstatix)

##   
## Attaching package: 'rstatix'

## The following object is masked from 'package:stats':  
##   
## filter

anova\_test(observed ~ ws.factor + Error(case.no/ws.factor),  
 data = mydata.long,  
 #effect.size = "ges",  
 #effect.size = "pes",  
 #type = 1  
 #type = 3  
 detail = TRUE) -> one.rmanova.res  
  
one.rmanova.res

## ANOVA Table (type III tests)  
##   
## $ANOVA  
## Effect DFn DFd SSn SSd F p p<.05 ges  
## 1 (Intercept) 1 215 13546289.6 791502.638 3679.649 3.23e-137 \* 0.945  
## 2 ws.factor 3 645 243285.2 557.543 93815.788 0.00e+00 \* 0.235  
##   
## $`Mauchly's Test for Sphericity`  
## Effect W p p<.05  
## 1 ws.factor 0.458 3.65e-34 \*  
##   
## $`Sphericity Corrections`  
## Effect GGe DF[GG] p[GG] p[GG]<.05 HFe DF[HF] p[HF] p[HF]<.05  
## 1 ws.factor 0.666 2, 429.61 0 \* 0.672 2.02, 433.64 0 \*

# ANOVA table (Type III SS) and Generalized Eta-squared measure (ges)  
get\_anova\_table(one.rmanova.res)

## ANOVA Table (type III tests)  
##   
## Effect DFn DFd SSn SSd F p p<.05 ges  
## 1 (Intercept) 1 215.00 13546289.6 791502.638 3679.649 3.23e-137 \* 0.945  
## 2 ws.factor 2 429.61 243285.2 557.543 93815.788 0.00e+00 \* 0.235

one.rmanova.res$`Mauchly's Test for Sphericity`

## Effect W p p<.05  
## 1 ws.factor 0.458 3.65e-34 \*

출력되는 ANOVA table에서..  
DFn: Degrees of Freedom in the numerator (i.e. DF effect)  
DFd: Degrees of Freedom in the denominator (i.e., DF error)  
SSn Sum of Squares in the numerator (i.e., SS effect)  
SSd Sum of Squares in the denominator (i.e.,SS error)  
F F-value  
p p-value (probability of the data given the null hypothesis)  
p<.05 Highlights p-values less than the traditional alpha level of .05  
ges Generalized Eta-Squared measure of effect size  
GGe Greenhouse-Geisser epsilon  
p[GGe] p-value after correction using Greenhouse-Geisser epsilon  
p[GGe]<.05 Highlights p-values (after correction using Greenhouse-Geisser epsilon) less than the traditional alpha level of .05  
HFe Huynh-Feldt epsilon  
p[HFe] p-value after correction using Huynh-Feldt epsilon  
p[HFe]<.05 Highlights p-values (after correction using Huynh-Feldt epsilon) less than the traditional alpha level of .05  
W Mauchly’s W statistic

*More about Mauchly’s Sphericity test* Mauchly’s Test of Sphericity (모쉬리의 구형성검정)은 ’모상관행렬은 단위행렬이다’라는 귀무가설을 검정하는 과정이다. 이것은 개체-내 효과검정을 시행하기 전에 오차항의 분산-공분산 행렬이 항등행렬에 비례한다는 가설에 대해서 타당성을 검토하고 Greenhouse-Geisser, Huynh-Feldt Epsilon을 계산해서 개체-내 효과검정의 F검정을 보정하는데 사용하게 된다. Epsilon값이 1보다 작아질수록 구형성의 가설은 적절하지 않음을 나타내고 이후 보정의 정도가 크게 된다. Epsilon의 하한값은 1/(m-1)으로 계산하며 (m= 반복측정회수), 3번 반복 측정된 경우 1/(3-1) = 0.5, 4번 반복 측정하면 1/(4-1) = 0.33이 된다. (이래서 5번 미만으로 해야 한다고 하나?) Epsilon이 0.40보다 작게 계산된다면, GG나 HF correction을 하더라도 보정된 P값에 sphericity를 violation한 영향이 미치게 된다. Correction은 F값을 계산하는 df에 대한 보정이 가해지며, epsilon이 0.75보다 크면 HF correction을, 0.75보다 작거나 sphericity에 대한 정보가 없을 때에는 GG correction을 시행한다.

rstatix::anova\_test는 sphericity검정을 시행해 주기 때문에 편리하고, partial eta square를 계산 (effect.size = “pes) 또는 generalized eta square (effect.size =”ges”)를 Type I SS와 Type III SS에서 계산도 가능하다. 다만 이것들을 한번에 계산할 수 없기 때문에 같은 code를 변형하여 여러번 시행하여야 한다.  
R의 basic 명령어인 “aov”와 “afex::aov\_car”를 이용해서 계산하는 방법도 아래 소개되어 있다.

# RM ANOVA with "aov" for Type I SS  
aov(observed ~ ws.factor + Error(case.no/ws.factor),  
 data = mydata.long) -> one.rmanova.res1  
  
 # Effect size: partial eta squared (Type I SS)  
 effectsize::eta\_squared(one.rmanova.res1,  
 partial = TRUE)

## Warning: Some CIs could not be estimated due to non-finite F, df, or df\_error  
## values.

## # Effect Size for ANOVA (Type I)  
##   
## Group | Parameter | Eta2 (partial) | 95% CI  
## -------------------------------------------------------------  
## case.no:ws.factor | ws.factor | 1.00 | [ , 1.00]  
## Within | ws.factor | 0.07 | [0.04, 1.00]  
##   
## - One-sided CIs: upper bound fixed at [1.00].

# Effect size: Generalized eta squared (Type I SS)  
 effectsize::eta\_squared(one.rmanova.res1,  
 generalized = TRUE)

## Warning: Some CIs could not be estimated due to non-finite F, df, or df\_error  
## values.

## # Effect Size for ANOVA (Type I)  
##   
## Group | Parameter | Eta2 (generalized) | 95% CI  
## -----------------------------------------------------------------  
## case.no:ws.factor | ws.factor | 0.19 | [ , 1.00]  
## Within | ws.factor | 0.07 | [0.04, 1.00]  
##   
## - Observed variables: All  
## - One-sided CIs: upper bound fixed at [1.00].

# Effect size: Cohen's f, partial (Type I SS)  
 effectsize::cohens\_f(one.rmanova.res1) -> one.rmanova.f1

## Warning: Some CIs could not be estimated due to non-finite F, df, or df\_error  
## values.

one.rmanova.f1

## # Effect Size for ANOVA (Type I)  
##   
## Group | Parameter | Cohen's f (partial) | 95% CI  
## -----------------------------------------------------------------  
## case.no:ws.factor | ws.factor | Inf | [ , Inf]  
## Within | ws.factor | 0.28 | [0.21, Inf]  
##   
## - One-sided CIs: upper bound fixed at [Inf].

# RM ANOVA with "afex::aov\_car" for Type III SS  
if (!requireNamespace("afex", quietly = TRUE)) {  
 install.packages("afex")  
}  
   
afex::aov\_car(observed ~ ws.factor + Error(case.no/ws.factor),  
 data = mydata.long) -> one.rmanova.res3  
  
 # Effect size: parital eta-squared (Type III SS)  
 effectsize::eta\_squared(one.rmanova.res3,  
 partial = TRUE)

## # Effect Size for ANOVA (Type III)  
##   
## Parameter | Eta2 (partial) | 95% CI  
## -----------------------------------------  
## ws.factor | 1.00 | [1.00, 1.00]  
##   
## - One-sided CIs: upper bound fixed at [1.00].

# Effect size: Generalized eta squared (Type III SS)  
 effectsize::eta\_squared(one.rmanova.res3,  
 generalized = TRUE)

## # Effect Size for ANOVA (Type III)  
##   
## Parameter | Eta2 (generalized) | 95% CI  
## ---------------------------------------------  
## ws.factor | 0.23 | [0.19, 1.00]  
##   
## - Observed variables: All  
## - One-sided CIs: upper bound fixed at [1.00].

# Effect size: Cohen's f, partial (Type III SS)  
 effectsize::cohens\_f(one.rmanova.res3)

## # Effect Size for ANOVA (Type III)  
##   
## Parameter | Cohen's f (partial) | 95% CI  
## ----------------------------------------------  
## ws.factor | 20.89 | [19.93, Inf]  
##   
## - One-sided CIs: upper bound fixed at [Inf].

*two way RM ANOVA (ANOVA with one repeated measure factor and one between-subject factor)*

anova\_test(observed ~ medication + ws.factor + medication\*ws.factor + Error(case.no/ws.factor),  
 data = mydata.long,  
 #effect.size = "pes",  
 #effect.size = "ges",  
 #type = 1,  
 #type = 3,  
 detailed = TRUE) -> two.rmanova.result  
  
get\_anova\_table(two.rmanova.result)

## ANOVA Table (type II tests)  
##   
## Effect DFn DFd SSn SSd F p  
## 1 (Intercept) 1.00 213.0 13546289.612 783582.677 3682.266 2.14e-136  
## 2 medication 2.00 213.0 7919.961 783582.677 1.076 3.43e-01  
## 3 ws.factor 2.00 425.3 243285.207 556.253 93158.518 0.00e+00  
## 4 medication:ws.factor 3.99 425.3 1.289 556.253 0.247 9.11e-01  
## p<.05 ges  
## 1 \* 9.45e-01  
## 2 1.00e-02  
## 3 \* 2.37e-01  
## 4 1.64e-06

two.rmanova.result$`Mauchly's Test for Sphericity`

## Effect W p p<.05  
## 1 ws.factor 0.458 6.89e-34 \*  
## 2 medication:ws.factor 0.458 6.89e-34 \*

# RM ANOVA with "aov" for Type I SS  
aov(observed ~ medication + ws.factor + medication\*ws.factor + Error(case.no/ws.factor),  
 data = mydata.long) -> two.rmanova.res1  
  
 # Effect size: partial eta squared (Type I SS)  
 effectsize::eta\_squared(two.rmanova.res1,  
 partial = TRUE)

## Warning: Some CIs could not be estimated due to non-finite F, df, or df\_error  
## values.

## # Effect Size for ANOVA (Type I)  
##   
## Group | Parameter | Eta2 (partial) | 95% CI  
## ------------------------------------------------------------------------  
## case.no | medication | 1.00 | [ , 1.00]  
## case.no:ws.factor | ws.factor | 1.00 | [ , 1.00]  
## Within | medication | 0.01 | [0.00, 1.00]  
## Within | ws.factor | 0.07 | [0.04, 1.00]  
## Within | medication:ws.factor | 2.66e-06 | [0.00, 1.00]  
##   
## - One-sided CIs: upper bound fixed at [1.00].

# Effect size: Generalized eta squared (Type I SS)  
 effectsize::eta\_squared(two.rmanova.res1,  
 generalized = TRUE)

## Warning: Some CIs could not be estimated due to non-finite F, df, or df\_error  
## values.

## # Effect Size for ANOVA (Type I)  
##   
## Group | Parameter | Eta2 (generalized) | 95% CI  
## ----------------------------------------------------------------------------  
## case.no | medication | 4.27e-05 | [ , 1.00]  
## case.no:ws.factor | ws.factor | 0.19 | [ , 1.00]  
## Within | medication | 0.01 | [0.00, 1.00]  
## Within | ws.factor | 0.07 | [0.04, 1.00]  
## Within | medication:ws.factor | 2.66e-06 | [0.00, 1.00]  
##   
## - Observed variables: All  
## - One-sided CIs: upper bound fixed at [1.00].

# Effect size: Cohen's f, partial (Type I SS)  
 effectsize::cohens\_f(two.rmanova.res1) -> two.rmanova.f1

## Warning: Some CIs could not be estimated due to non-finite F, df, or df\_error  
## values.

two.rmanova.f1

## # Effect Size for ANOVA (Type I)  
##   
## Group | Parameter | Cohen's f (partial) | 95% CI  
## ----------------------------------------------------------------------------  
## case.no | medication | Inf | [ , Inf]  
## case.no:ws.factor | ws.factor | Inf | [ , Inf]  
## Within | medication | 0.10 | [0.03, Inf]  
## Within | ws.factor | 0.28 | [0.21, Inf]  
## Within | medication:ws.factor | 1.63e-03 | [0.00, Inf]  
##   
## - One-sided CIs: upper bound fixed at [Inf].

# RM ANOVA with "afex::aov\_car" for Type III SS  
  
afex::aov\_car(observed ~ medication + ws.factor + medication\*ws.factor + Error(case.no/ws.factor),  
 data = mydata.long) -> two.rmanova.res3

## Contrasts set to contr.sum for the following variables: medication

# Effect size: parital eta-squared (Type III SS)  
 effectsize::eta\_squared(two.rmanova.res3,  
 partial = TRUE)

## # Effect Size for ANOVA (Type III)  
##   
## Parameter | Eta2 (partial) | 95% CI  
## ----------------------------------------------------  
## medication | 0.01 | [0.00, 1.00]  
## ws.factor | 1.00 | [1.00, 1.00]  
## medication:ws.factor | 2.31e-03 | [0.00, 1.00]  
##   
## - One-sided CIs: upper bound fixed at [1.00].

# Effect size: Generalized eta squared (Type III SS)  
 effectsize::eta\_squared(two.rmanova.res3,  
 generalized = TRUE)

## # Effect Size for ANOVA (Type III)  
##   
## Parameter | Eta2 (generalized) | 95% CI  
## --------------------------------------------------------  
## medication | 1.00e-02 | [0.00, 1.00]  
## ws.factor | 0.24 | [0.19, 1.00]  
## medication:ws.factor | 1.64e-06 | [0.00, 1.00]  
##   
## - Observed variables: All  
## - One-sided CIs: upper bound fixed at [1.00].

# Effect size: Cohen's f, partial (Type III SS)  
 effectsize::cohens\_f(two.rmanova.res3) -> two.rmanova.f3  
   
 two.rmanova.f3

## # Effect Size for ANOVA (Type III)  
##   
## Parameter | Cohen's f (partial) | 95% CI  
## ---------------------------------------------------------  
## medication | 0.10 | [ 0.00, Inf]  
## ws.factor | 20.91 | [19.95, Inf]  
## medication:ws.factor | 0.05 | [ 0.00, Inf]  
##   
## - One-sided CIs: upper bound fixed at [Inf].

# 3 SESSION 3: BACK TO YOUR HOME – PREPARE FOR THE NEXT JOURNEY

## 3.1 Session 3-1: Basic concepts of power analysis

## 3.2 Session 3-2: Sample size calculation: R vs. G\*Power

R에서 power analysis를 위해 필요한 package는 다음과 같다.  
pwr  
WebPower (for ANOVA series)

if (!requireNamespace("pwr", quietly = TRUE)) {  
 install.packages("pwr")  
}  
  
if (!requireNamespace("WebPower", quietly = TRUE)) {  
 install.packages("WebPower")  
}

### 3.2.1 T-test

1. 등분산성이 확립되었을 때  
   이전 session 2-3에서 t-test 이후 effect size Cohen’s d를 저장해 두었다.

# Effect size  
cohend

## Cohen's d | 95% CI  
## ------------------------  
## 1.22 | [0.93, 1.51]  
##   
## - Estimated using pooled SD.

pwr::pwr.t.test(d = cohend[1,1],  
 sig.level = 0.05,  
 power = 0.80,  
 type = "two.sample",  
 alternative = "greater") -> t.test.n  
  
t.test.n

##   
## Two-sample t test power calculation   
##   
## n = 9.104912  
## d = 1.217271  
## sig.level = 0.05  
## power = 0.8  
## alternative = greater  
##   
## NOTE: n is number in \*each\* group

# Considering drop-out rate of 10%  
t.test.n.do <- ceiling(t.test.n$n)/0.9  
  
ceiling(t.test.n.do)

## [1] 12

1. 등분산성이 확립되지 않을 때

cohend.ueq

## Cohen's d | 95% CI  
## ------------------------  
## 1.22 | [0.92, 1.51]  
##   
## - Estimated using un-pooled SD.

poolsd <- effectsize::sd\_pooled(t\_test.value ~ group, data = mydata)  
poolsd

## [1] 9.675561

pwr::pwr.t.test(d = cohend.ueq[1,1],  
 sig.level = 0.05,  
 power = 0.80,  
 type = "two.sample",  
 alternative = "greater") -> t.test.ueq.n  
t.test.ueq.n

##   
## Two-sample t test power calculation   
##   
## n = 9.104912  
## d = 1.217271  
## sig.level = 0.05  
## power = 0.8  
## alternative = greater  
##   
## NOTE: n is number in \*each\* group

# Considering drop-out rate of 10%  
t.test.ueq.n.do <- ceiling(t.test.ueq.n$n)/0.9  
  
ceiling(t.test.ueq.n.do)

## [1] 12

### 3.2.2 Paired T-test

pwr::pwr.t.test(d = cohend.pair[1,1],  
 sig.level = 0.05,  
 power = 0.8,  
 type = "paired") -> t.test.pair.n  
  
t.test.pair.n

##   
## Paired t test power calculation   
##   
## n = 4.586183  
## d = 1.830362  
## sig.level = 0.05  
## power = 0.8  
## alternative = two.sided  
##   
## NOTE: n is number of \*pairs\*

# Considering drop-out rate of 10%  
t.test.pair.n.do <- ceiling(t.test.pair.n$n)/0.9  
  
ceiling(t.test.pair.n.do)

## [1] 6

### 3.2.3 Chi-square test

chisq.res$parameter[[1]]

## [1] 1

pwr::pwr.chisq.test(w = cramerv[1,1],  
 df = chisq.res$parameter[[1]],  
 sig.level = 0.05,  
 power = 0.8) -> chisq.n  
  
chisq.n

##   
## Chi squared power calculation   
##   
## w = 0.2121526  
## N = 174.3853  
## df = 1  
## sig.level = 0.05  
## power = 0.8  
##   
## NOTE: N is the number of observations

# Considering drop-out rate of 10%  
chisq.n.do <- ceiling(chisq.n$N)/0.9  
  
ceiling(chisq.n.do)

## [1] 195

### 3.2.4 ANOVA

1. One-way ANOVA

이전 ANOVA test에서 group의 수는 다음을 통해 3군임을 확인할 수 있다.

table(mydata$medication)

##   
## 1 2 3   
## 72 72 72

One-way ANOVA의 표본수 계산에는 Cohen’s f를 사용한다.

one.anova.f[1,2] -> f  
  
  
pwr::pwr.anova.test(k = 3, # number of groups  
 f = f,  
 sig.level = 0.05,  
 power = 0.8) -> one.anova.n  
one.anova.n

##   
## Balanced one-way analysis of variance power calculation   
##   
## k = 3  
## n = 30.6689  
## f = 0.329125  
## sig.level = 0.05  
## power = 0.8  
##   
## NOTE: n is number in each group

# Considering drop-out rate of 10%  
one.anova.n.do <- ceiling(one.anova.n$n)/0.9  
  
ceiling(one.anova.n.do)

## [1] 35

각 군당 35명, 전체 105명의 환자수가 필요하다.

1. Two-way ANOVA  
   이전 session에서 시행한 two-way ANOVA는

* 3가지 medications (예: 3가지 항구토제)
* 2개 groups(예: 남/여, young age/old age)  
  를 대상으로 검정한 것이다.  
  이와 같은 상황에서 표본수 계산은 “WebPower::wp.kanova”를 이용하여 계산 할 수 있다. 계산을 위해 필요한 정보는 다음과 같다.

1. Numerator degrees of freedom = df(medication) df(group) = (3-1) (2-1) = 2
2. effect size f
3. Number of groups: No of medications No of groups = 3 2 = 6
4. alpha = 0.05 & power = 0.8

WebPower::wp.kanova(ndf = 2,  
 f = two.anova.f.i[3,2],  
 ng = 6,  
 alpha = 0.05,  
 power = 0.8) -> two.anova.n  
  
two.anova.n

## Multiple way ANOVA analysis  
##   
## n ndf ddf f ng alpha power  
## 487.1993 2 481.1993 0.1410649 6 0.05 0.8  
##   
## NOTE: Sample size is the total sample size  
## URL: http://psychstat.org/kanova

# Considering drop-out rate of 10%  
two.anova.n.do <- ceiling(two.anova.n$n)/0.9  
  
ceiling(two.anova.n.do)

## [1] 543

전체 543명의 환자수가 필요하다.

### 3.2.5 RM ANOVA

1. RM ANOVA with one repeated factor (within-subject factor)

RM ANOVA with one repeated factor의 표본수 계산은 WebPower::wp.rmanova를 이용할 수 있다.  
wp.rmanova를 이용하기 위한 정보들은 다음과 같다.  
a. number of groups (ng): 1 b. number of measurements (nm) = 4 (obs1 ~ obs4) c. effect size: Cohen’s f d. nonsphericity correction coefficient (nscor) = 0.672 e. alpha & Power f. type = 0, 1, 2: The value “0” is for between-effect; “1” is for within-effect; and “2” is for interaction effect.

effect size는 within-subject factor에 대한 값을 이용한다 (between-subject에 대한 변수가 이 분석에는 포함되지 않는다).

f = one.rmanova.f1[2,3]

Nonsphericity correction coefficient (epsilon)  
이전 실행된 결과에서 Sphericity corrections부분을 살펴보면

one.rmanova.res$`Sphericity Corrections`

## Effect GGe DF[GG] p[GG] p[GG]<.05 HFe DF[HF] p[HF] p[HF]<.05  
## 1 ws.factor 0.666 2, 429.61 0 \* 0.672 2.02, 433.64 0 \*

HFe (Huynh-Feldt epsilon)이 0.672이다.

Type는 하나의 within-subject factor (repeated measures factor)만 있는 분석이기 때문에 1 (for within-effect)를 선택하면 된다.

WebPower::wp.rmanova(ng = 1,  
 nm = 4,  
 f = one.rmanova.f1[2,3],  
 nscor = one.rmanova.res$`Sphericity Corrections`[1,6], # = 0.672  
 alpha = 0.05, power = 0.80,  
 type = 1) -> one.rmanova.n  
one.rmanova.n

## Repeated-measures ANOVA analysis  
##   
## n f ng nm nscor alpha power  
## 190.5745 0.2756987 1 4 0.672 0.05 0.8  
##   
## NOTE: Power analysis for within-effect test  
## URL: http://psychstat.org/rmanova

# Considering drop-out rate of 10%  
one.rmanova.n.do <- ceiling(one.rmanova.n$n)/0.9  
  
ceiling(one.rmanova.n.do)

## [1] 213

1. two way RM ANOVA (ANOVA with one repeated measure factor and one between-subject factor)

반복측정 변수 한개와 군을 나누는 변수 한개인 경우이다.  
표본수 계산은 WebPower::wp.rmanova를 사용한다.

1. number of groups (ng): 3

table(mydata.long$medication)

##   
## 1 2 3   
## 288 288 288

1. number of measurements (nm) = 4 (obs1 ~ obs4)

table(mydata.long$ws.factor)

##   
## obs1 obs2 obs3 obs4   
## 216 216 216 216

1. effect size: Cohen’s f  
   이전 통계 결과는 medication에 따른 결과만 유의하게 차이 있는 것으로 나왔으며, interaction은 없다고 결론지어졌다. 그러나, 이 연습에서는 interaction이 있다고 가정하고, 표본수계산을 시행해보자. 따라서, f값은 medication:ws.factor에 해당하는 것을 사용한다.

two.rmanova.f1

## # Effect Size for ANOVA (Type I)  
##   
## Group | Parameter | Cohen's f (partial) | 95% CI  
## ----------------------------------------------------------------------------  
## case.no | medication | Inf | [ , Inf]  
## case.no:ws.factor | ws.factor | Inf | [ , Inf]  
## Within | medication | 0.10 | [0.03, Inf]  
## Within | ws.factor | 0.28 | [0.21, Inf]  
## Within | medication:ws.factor | 1.63e-03 | [0.00, Inf]  
##   
## - One-sided CIs: upper bound fixed at [Inf].

two.rmanova.f1[5,3] -> f

1. nonsphericity correction coefficient (nscor) = 0.672

two.rmanova.result$`Sphericity Corrections`

## Effect GGe DF[GG] p[GG] p[GG]<.05 HFe DF[HF]  
## 1 ws.factor 0.666 2, 425.3 0.000 \* 0.672 2.02, 429.32  
## 2 medication:ws.factor 0.666 3.99, 425.3 0.911 0.672 4.03, 429.32  
## p[HF] p[HF]<.05  
## 1 0.000 \*  
## 2 0.913

1. alpha & Power
2. type = 0, 1, 2: The value “0” is for between-effect; “1” is for within-effect; and “2” is for interaction effect

WebPower::wp.rmanova(ng = 3,  
 nm = 4,  
 f = two.rmanova.f1[5,3],  
 nscor = two.rmanova.result$`Sphericity Corrections`[1,6], # = 0.672  
 alpha = 0.05, power = 0.80,  
 type = 2) -> two.rmanova.n  
two.rmanova.n

## Repeated-measures ANOVA analysis  
##   
## n f ng nm nscor alpha power  
## 6696831 0.001630609 3 4 0.672 0.05 0.8  
##   
## NOTE: Power analysis for interaction-effect test  
## URL: http://psychstat.org/rmanova

# Considering drop-out rate of 10%  
two.rmanova.n.do <- ceiling(two.rmanova.n$n)/0.9  
  
ceiling(two.rmanova.n.do)

## [1] 7440925

-End of File-