논문작성을 위한 R통계분석: Statistical analysis of papers WithR

Park Joonghee (Yonsei University coginitive Eng. lab)

2023-12-16

# 연구방법과 통계분석

**저자의 말**

이 책에서 다루고자 하는 것은 실제로 논문을 집필하거나 혹은 통계분석을 하고자 하는 경우에 실질적으로 필요한 부분과 집중해야할 부분을 담았다. 연구자로서 직접 분석을 하고 자신의 연구를 정밀하게 담고 싶은 마음은 정말 간절할 것이다. R은 무료로 이 모든 것을 가능하게 해주며, 배우는 것도 그리 어려운 것은 아니다. 그러나, R을 이용하여 통계분석을 하고자 하는 경우, 너무 기초에 매달리다보니 실제로 필요한 것을 익히는데 너무 올래걸리는 것이 현실이다. 연구자들에게 통계분석만을 목적으로 사용한다면 너무 어려운 것은 아님에도 불구하고 코드(syntax)방식이라는 이유로 벽에 부딪히는 경우가 많다. 그러나, 연구를 재현하는 관점에서 접근해보면 이러한 코드형태로 사용하는 방식은 매우 편리한 방법임에는 그 누구도 부인할 수 없다.  
본 책에서는 논문 통계분석의 절차에 따라서 사용하는 R함수와 라이브러리를 소개하여 분석하는 다양한 방법을 소개하면서 R을 익힐 수 있게 하고 있다. 함수를 일일이 기억하는 것은 매우 어려운 일이지만 상황에 맞추어서 최적의 함수를 사용하는 것은 경험으로서 기억이 잘 나고 찾아볼 수 있으므로 그러한 방식으로 분석방법을 소개하려고 한다. 분석후에는 간단하게라도 가능한 APA양식에 맞추어서 보고하는 방식을 소개하고 있다.

이 책에서는 필자가 개발한 R패키지도 사용하고 있다. jjstat라는 패키지를 이용하여 여러가지 분석의 방법과 요약기술등을 편리하게 하는 방법들을 제시하고 있다. 대체로 분석하는 것에 대한 함수는 잘 발달해있지만, 이를 정리하여 논문에 넣기위한 함수나 정리방법은 좀 미흡한 편이다. 따라서, 필자는 분석을 편리하게 하는 방법과 그리고, 결과물을 만들기 위해서 도입해야 하는 함수들을 모아서 jjstat라는 패키지로 담았다.

# 1. 데이터에 대한 이해

데이터를 이해한다는 것을 자신이 분석할 실태를 잘 이해하고 있다는 의미가 된다. 데이터를 이해한다는 것은 자신의 데이터의 열에 대한 것, 즉 변수가 무엇인지 정확히 이해하는 것이다. 그 변수의 형식과 내용에 대하여 명확히 이해하는 것이 중요하다. 보통은 행렬(matrix), 리스트(list), 배열(array), 데이터프레임(data.frame), 티블(tibble)데이터를 자주 사용한다. 이러한 형식과 변수의 형식, 그리고 변수를 리코딩하여 원하는 분석에 적용하는 방식을 잘 이해해야 한다. 이러한 과정을 배우는 것은 처음에 R을 배울 때 어렵게 만드는 경향이 있다.

## 1. 논문을 위한 기본 개념

### 1.1. 연구주제의 선정

연구 주제 선정의 중요성  
논문 작성의 첫 걸음이자 성공적인 연구의 핵심 요소는 바로 연구 주제의 선정이다. 잘 선택된 주제는 연구의 방향성을 제시하고, 명확한 목표 설정을 가능하게 하며, 학문적 기여도를 높여주게 된다. 반대로, 부적절한 주제는 연구의 어려움, 시간 낭비, 학문적 가치 감소로 이어질 수 있다.

**연구 주제 선정을 위한 5가지 전략**  
1) 관심 분야 및 전문성 고려:  
자신의 관심 분야와 전문 지식을 기반으로 주제를 선정하면 연구에 대한 열정과 몰입도를 높일 수 있다. 이미 갖춘 지식을 바탕으로 연구를 시작하면 학습 곡선을 단축하고 연구 효율성을 높일 수 있다.

1. 최신 동향 및 학문적 격차 파악:  
   최신 학문적 트렌드와 연구 동향을 파악하여 미래 지향적인 주제를 발굴하고, 학문적 격차를 채울 수 있는 연구를 진행하는것이 좋다. 학술 논문, 학회 발표, 전문가 의견 등을 참고하여 최신 연구 동향을 파악하는 것도 매우 중요하다.
2. 연구 가능성 및 실현 가능성 평가:  
   연구 주제의 연구 가능성을 평가하여 성공적인 연구 수행 가능성을 높일 수 있는 방법을 찾아본다. 연구 자료, 방법론, 윤리적 문제 등을 고려하여 연구 가능성을 평가한다. 이때 중요한 것은, 실현 가능성을 위해 연구 시간, 예산, 인력, 장비 등 현실적인 제약 조건을 고려하는 것이 중요하다.
3. 학문적 기여도 및 사회적 영향력 검토:  
   연구 주제의 학문적 기여도와 사회적 영향력을 검토하여 연구의 가치를 높여야 한다. 연구 결과가 학문 분야에 어떤 기여를 할 수 있는지, 사회에 어떤 영향을 미칠 수 있는지 고려한다.
4. 명확한 연구 질문 및 가설 설정:  
   연구 주제를 바탕으로 명확한 연구 질문과 가설을 설정하여 연구 방향성을 명확하게 제시한다. 연구 질문은 연구의 핵심을 요약하고, 가설은 연구 결과를 예측하는 명제이므로 철저한 선행연구와 연구자의 깊은 고민을 통해서 가설에 대한 예측을 하는 것이 중요하다.

# 2. 기초통계분석

## 2.1. 연구에 관한 이해

### 연구가설과 귀무가설

연구 가설은 연구 결과를 예측하는 명제이다. 연구가설은 연구 질문을 바탕으로 설정하며, 연구의 방향성을 제시하고 검증 가능성을 제공하므로, 명확하고 구체적인 가설 설정은 연구의 성공에 중요한 역할을 하게된다.

효과적인 연구 가설 설정 방법을 알아보자.

먼저, 명확하고 구체적인 표현을 해야 한다. 연구 질문에 대한 직접적인 답변을 제공하고, 체계적인 선행연구를 통하여 측정 가능한 변수를 포함해야 한다. 둘째, 검증이 가능해야 한다. 경험적 또는 논리적 방법으로 검증 가능한 연구여야 한다. 셋째, 관련성이 있어야 한다. 연구가설은 선행 연구와 이론적 근거를 바탕으로 설정해야 한다. 특히 조심해야 할 것은 연구자의 개인 경험에 의존해서는 안된다는 것이다. 대체로 선행연구가 부족한 경우 자신의 경험으로 모든 가설을 설정하는 경우가 있는데 이는 피해야 한다. 넷째, 단순해야 한다. 즉 이해하기 쉽고 불필요한 복잡성을 피해야 한다는 것이다. 다섯째, 파괴력이 있어야 한다. 기존 이론이나 가설에 대한 새로운 도전을 제시하는 것이 바람직하다.

연구가설의 예를 들어보자.  
연구 질문: 온라인 학습 참여도가 학습 성취도에 미치는 영향은 무엇인가?  
연구 가설: 온라인 학습 참여도가 높을수록 학습 성취도가 높아질 것이다.

**대립가설 (Alternative Hypothesis)** :

연구자가 검증하고자 하는 가설이다. 즉 연구 결과에 대한 예상이나 가정을 나타낸다. 보통 “H1”로 표기하며, 예를 들어 “A 치료법은 B 치료법보다 더 효과적이다”와 같은 가설이 연구가설을 말한다.

**귀무가설 (Null Hypothesis)**:

연구자가 연구가설을 검증하기 위해 반박하려는 가설이다. 통계적 검정에서 기본적으로 설정되는 가설이다. 보통 “H0”로 표기하며, 예를 들어 “A 치료법과 B 치료법은 효과적인 차이가 없다”와 같은 가설이 귀무가설이다. 연구가설과 귀무가설은 통계적 검정에서 중요한 역할을 하며, 연구 결과를 평가하고 결론을 도출하는데 사용된다.

**연구 가설의 유형**:  
연구가설의 유형으로는 다음과 같은 것들이 있다.  
- 서술적 가설: 두 변수 간의 관계를 단순히 서술하는 가설  
- 관계적 가설: 두 변수 간의 방향 또는 강도를 명시하는 가설  
- 원인적 가설: 한 변수가 다른 변수에 영향을 미치는 관계를 명시하는 가설

### 제 1종오류와 2종 오류

1종 오류와 2종 오류는 통계적 검정에서 중요한 개념이다.

**1종 오류 (Type I Error)**:  
- 의미: 귀무가설이 실제로 참인데도, 이에 불구하고 귀무가설을 기각하는 오류이다. 예를 들어보면, 의사가 환자에게 병이 없다고 진단했는데, 실제로는 병이 없는 경우이거나, 신약의 효과가 없는데 있다고 잘못 판단한 경우 등이다.

**2종 오류 (Type II Error)**:  
- 의미: 귀무가설이 실제로 거짓인데도, 이에 불구하고 귀무가설을 기각하지 못하는 오류이다. 예를 들면, 의사가 환자에게 병이 있다고 진단했는데, 실제로는 병이 있는 경우이다. 혹은 신약의 효과가 있는데 없다고 잘못 판단한 경우이다.  
이 두 오류는 통계적 검정에서 발생할 수 있으며, 연구 결과를 평가하고 결론을 도출하는데 중요한 역할을 한다. 1종 오류는 실제 음성인 것을 양성으로 판정하는 경우이고, 2종 오류는 실제 양성인 것을 음성으로 판정하는 경우이다.

### 연구모형의 설정

연구 모형은 연구 변수 간의 관계를 시각적으로 표현한 도식으로, 연구 가설을 검증하기 위한 구조를 제시하며, 연구 변수 간의 상호 작용을 이해하는 데 도움을 준다.

연구 모형 설정 방법은 다음과 같다. (1) 연구 가설을 기반으로 연구 가설을 검증하기 위한 변수와 그 관계를 명확하게 설정해야 한다. (2) 변수 선정시에는 연구 질문과 가설에 부합하는 주요 변수를 선정하고, 변수 간의 논리적 관계를 설정한다. (3) 변수 간의 관계를 화살표로 표현하여 나타낸다. 화살표 방향은 영향 관계를 나타낸다. (4) 모형 유형의 설정을 한다. 연구 목적에 따라 다양한 모형 유형을 선택할 수 있다. 예를 들어, 경로 분석, 구조 방정식 모델링, 회귀 분석 등이 있다.

### 연구 가설 및 연구 모형 설정 시 고려해야 할 사항

1. 연구의 목적과 범위:  
   연구 목적과 범위를 명확하게 설정해야 연구 가설과 연구 모형을 효과적으로 설정할 수 있다.
2. 선행 연구:  
   선행 연구 결과를 참고하여 연구 가설과 연구 모형을 설정하는 것이 바람직하다.
3. 통계적 검증 가능성:  
   연구 가설과 연구 모형은 통계적 방법으로 검증 가능해야 한다.
4. 윤리적 문제:  
   연구 가설과 연구 모형 설정 과정에서 윤리적 문제를 고려해야 합니다.

### 유의수준과 신뢰수준, 유의확률

1. 유의수준 (α):  
   귀무가설이 사실인데도 불구하고 기각할 확률을 의미한다. 일반적으로 0.05 또는 0.01로 설정한다.
2. 신뢰수준 (1-α):  
   귀무가설이 사실인 경우 기각하지 않을 확률을 의미한다. 즉, 연구 결과가 신뢰할 수 있는 정도를 나타낸다.
3. 유의확률 (p-value):  
   표본 데이터에서 관찰된 결과가 우연히 발생할 확률을 의미한다. 유의확률이 유의수준보다 작으면 귀무가설을 기각하고 대립가설을 채택한다.

* 통계적 의사결정

## 2.2. 데이터의 처리와 이해

- 변수변환 처리 recode - 역코딩 방법 데이터를 역코딩 하는 방법은 단순하다.

# 3. 기본 통계분석

## 3.1. ㅣ기술통계(Descriptive statistics)

기본적인 분석은 가장 먼저 해야할 분석을 말한다.  
대푯값과 산포도를 중심으로 분석하게 된다. 대표값에는 평균, 중앙값, 최빈값을 분석하게 되는데 평균은 mean(), 중앙값은 median(), 최빈값은 table()을 사용하여 분석한다.

### 3.1.1. 빈도분석

먼저 R에 기본으로 탑재된 mtcars 데이터를 이용하여 빈도분석을 해보자.  
데이터의 구조는 다음과 같다.  
이 데이터는 1974년 미국 Motor Trend US 잡지에서 추출된 자료로 32개의 자동차회사의 1973년~1974년 모델에 대한 자동차의 10가지의 디자인 및 성능에 대한 연료 소비량(mpg)를 나타낸 자료이다.

A data frame with 32 observations on 11 (numeric) variables.  
[, 1] mpg Miles/(US) gallon  
[, 2] cyl Number of cylinders  
[, 3] disp Displacement (cu.in.)  
[, 4] hp Gross horsepower  
[, 5] drat Rear axle ratio  
[, 6] wt Weight (1000 lbs)  
[, 7] qsec 1/4 mile time  
[, 8] vs Engine (0 = V-shaped, 1 = straight)  
[, 9] am Transmission (0 = automatic, 1 = manual)  
[,10] gear Number of forward gears  
[,11] carb Number of carburetors  
  
데이터의 구조를 보는 좋은 방법중 하나가 str()함수를 사용하는 것이다.

library(tidyverse) # dplyr: %>% (pipe)  
data(mtcars)  
mtcars %>% str() # same as str(mtcars)

## 'data.frame': 32 obs. of 11 variables:  
## $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...  
## $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...  
## $ disp: num 160 160 108 258 360 ...  
## $ hp : num 110 110 93 110 175 105 245 62 95 123 ...  
## $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...  
## $ wt : num 2.62 2.88 2.32 3.21 3.44 ...  
## $ qsec: num 16.5 17 18.6 19.4 17 ...  
## $ vs : num 0 0 1 1 0 1 0 1 1 1 ...  
## $ am : num 1 1 1 0 0 0 0 0 0 0 ...  
## $ gear: num 4 4 4 3 3 3 3 4 4 4 ...  
## $ carb: num 4 4 1 1 2 1 4 2 2 4 ...

#### 3.1.1.1. table()

논문의 분석을 시작할 때, 항상 설문이나 실험참가자의 빈도분석을 통해서 제시하게 되어있는데, 이때 사용하는 것이 table함수이다.  
빈도분석은 보통 table()함수를 활용하는 경우가 많다. 빈도분석은 입력된 데이터들이 도수분포표(frequency table)상에서 어떠한 분포적 특성을 가지고 있는지 파악하는 데 사용된다.

mtcars의 am변수는 Transmission을 나타낸다 (0 = automatic, 1 = manual).  
따라서, 먼저 변수를 factor처리를 먼저해서 빈도를 분석해보자.

# mtcars$am  
data(mtcars)  
mtcars$am <- factor(mtcars$am,  
 level = c(0, 1),  
 labels = c("automatic", "manual" ))  
# frequency analysis  
table(mtcars$am)

##   
## automatic manual   
## 19 13

분석된 결과는 다음과 같이 설명한다.  
빈도분석결과, 자동차의 트랜스미션(transmission)에 대하여 오토매틱은 19개, 수동은 13개로 나타났다.  
이번에는 엔진(engine)에 대한 빈도분석을 해보자.

mtcars$vs <- factor(mtcars$vs, level=c(0, 1),  
 labels = c("V-shaped", "straight" ) )  
table(mtcars$vs)

##   
## V-shaped straight   
## 18 14

트랜스미션(transmission)별 엔진(Enigne)의 빈도분석을 해보자. 이는 table(변수1, 변수2)를 넣으면 분석이 된다. 행렬(matrix)의 형태로 빈도를 볼 수 있다.

table(mtcars$am, mtcars$vs)

##   
## V-shaped straight  
## automatic 12 7  
## manual 6 7

그런데 이러한 방법이 아닌 목록으로 정리하여 보고 싶다면 데이터프레임(data.frame)으로 변경하면 목록형태로 나타나서 논문표에서 넣기 더 편할 수 있다.

data.frame(table(mtcars$am, mtcars$vs))

## Var1 Var2 Freq  
## 1 automatic V-shaped 12  
## 2 manual V-shaped 6  
## 3 automatic straight 7  
## 4 manual straight 7

#### 3.1.1.2. count(), n()

R을 사용하는 경우 다양한 라이브러리(library)를 사용할 수 있기 때문에 분석하는 방법이 한가지로 고정되어 있는 것은 아니다. 사용하는 라이브러리를 이용하여 다양한 분석을 할 수 있다. 먼저. dplyr패키지를 이용하여 나타내는 방법도 매우 유용하다. dplyr::count()는 매우 편리한 함수이다. 사용방법은 count(데이터, 변수)로 입력하여 사용한다.

count(mtcars, am)

## am n  
## 1 automatic 19  
## 2 manual 13

# mtcars %>% count(am)

count(mtcars, am, vs)

## am vs n  
## 1 automatic V-shaped 12  
## 2 automatic straight 7  
## 3 manual V-shaped 6  
## 4 manual straight 7

좀 불편할 수 있으나, 실제 활용도가 높은 것중 하나가 n()이다. 빈도분석하는 과정에서 수시로 사용회는 함수인데, summarise()와 함께 사용하는 유용한 함수이다.  
summarise()와 n()을 이용하여 다음과 같이 나타낼 수 있다.

mtcars %>% group\_by(am)%>% summarise(N=n())

## # A tibble: 2 × 2  
## am N  
## <fct> <int>  
## 1 automatic 19  
## 2 manual 13

#### forcats::fct\_count()

이번에는 forcats라는 라이브러리의 fct\_count()이다. 필자의 경우도 매우 자주 사용하고 애용하는 라이브러리함수이다. 그 이유는 여러가지로 변형이 가능하기 때문이다. 그러나, fct\_count()는 vector형식으로 한가지 변수만 분석이 가능하다. 출력은 tibble데이터로 출력되어 table()와는 다른 형식으로 출력된다. fct\_count(데이터)를 입력하면 되고, 옵션으로는 sort를 이용한 정렬기능과 prop라는 비율을 나타내는 옵션이 있다.  
먼저 일반적인 빈도분석이다.

#frequency analysis : default option   
library(forcats)  
forcats::fct\_count(mtcars$am)

## # A tibble: 2 × 2  
## f n  
## <fct> <int>  
## 1 automatic 19  
## 2 manual 13

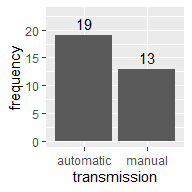
이번에는 각 빈도에 비율을 포함시켜보자.

# add proportion   
forcats::fct\_count(mtcars$am, prop=TRUE)

## # A tibble: 2 × 3  
## f n p  
## <fct> <int> <dbl>  
## 1 automatic 19 0.594  
## 2 manual 13 0.406

* 시각화

fct\_count(mtcars$am) %>%   
 ggplot(aes(x=f, y=n))+  
 geom\_bar(stat="Identity")+  
 geom\_text(aes(label=n), vjust=-.5)+  
 labs(x="transmission", y="frequency")+  
 ylim(0, 23)



#### 빈도의 수가 0인 경우 출력하는 방법

빈도수가 0인 것을 출력해야 하는 경우는 다양한 변수별로 빈도를 분석해서 데이터를 병합할 때 컬럼의 개수가 맞지 않으면 병합하기 어렵기 때문이다. 리커트 척도로 5개의 척도에 대하여 빈도를 분석한 경우, 0이라고 출력하지 않으면 어떤 것은 3개, 4개, 5개로 달라지기 때문에 병합할 수 없게 된다. 따라서, 빈도가 0으로 나와도 출력을 해주어야 하는 경우가 발생한다.

y <- c(0, 0, 0, 3, 4, 4, 4, 4)  
z <- c(0, 0, 1, 3, 4, 5, 5, 5)  
table(y)

## y  
## 0 3 4   
## 3 1 4

0에서 5까지의 변수중에서 개수를 센 상태인데, 2의 경우는 0개이다. 그런데 이 경우 2와 5가 0이라고 나타내고 싶은 경우는 factor로 처리하면서 2를 세도록 해야 한다.

table(factor(y, levels = 0:5))

##   
## 0 1 2 3 4 5   
## 3 0 0 1 4 0

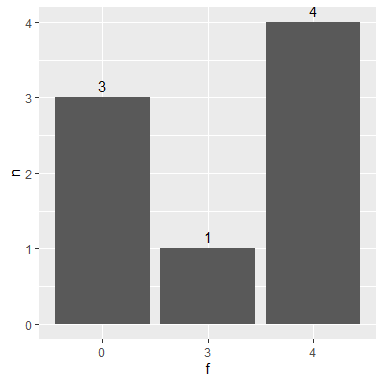
이것은 fct\_count()에서도 동일하다. 다만, fct\_count함수는 벡터에서만 사용한다.

y <- c(0, 0, 0, 3, 4, 4, 4, 4)  
#2가 0개에서 표시가 안된 경우   
fct\_count(factor(y))

## # A tibble: 3 × 2  
## f n  
## <fct> <int>  
## 1 0 3  
## 2 3 1  
## 3 4 4

2를 0개라고 표시한 경우이다.  
이를 시각화해보면 막대그래프가 4개가 그려지면서 실제 빈도수가 나타난다.

y <- c(0, 0, 0, 3, 4, 4, 4, 4)  
fct\_count(factor(y))%>%   
 ggplot(aes(x=f, y=n))+  
 geom\_bar(stat="Identity")+  
 geom\_text(aes(label=n), vjust=-.5)

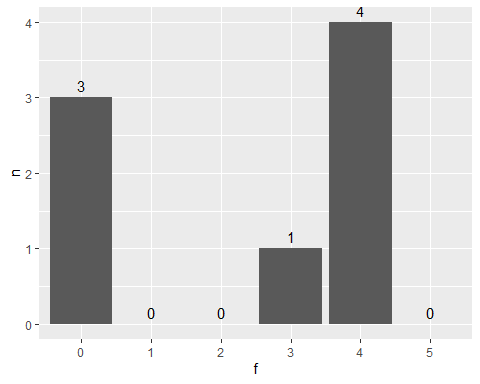


#2를 0개라고 표시한 경우   
fct\_count(factor(y, level=0:5))

## # A tibble: 6 × 2  
## f n  
## <fct> <int>  
## 1 0 3  
## 2 1 0  
## 3 2 0  
## 4 3 1  
## 5 4 4  
## 6 5 0

이것을 다시 시각화해보자.

fct\_count(factor(y, level=0:5)) %>%   
 ggplot(aes(x=f, y=n))+  
 geom\_bar(stat="Identity")+  
 geom\_text(aes(label=n), vjust=-.5)



* 사용자 함수만들어서 빈도분석하기

### 3.1.2 기술통계분석

#### 3.1.2.1 대푯값

대푯값에는 평균, 중앙값, 최빈값등이 있다. 평균을 구할 때에는 mean(), 중앙값은 median(), 최빈값을 구할 때에는 table()을 사용하는 것이 일반적이다.

### 3.1.3. 정규성 검정

## 3.2. 척도분석

이 부분은 처음 통계를 공부하거나 R을 사용하여 분석을 하는 경우 쉽지 않은 부분이다. 따라서, 이 부분을 과감히 넘어가는 것도 가능하다.

### 3.2.1. 신뢰도 분석

reliability는

### 3.2.2. 요인 분석

* 탐색적 요인 분석
* 확인적 요인 분석 지표 타당도 복합 신뢰도 수렴 타당성 판별 타당성

## 3.3. t-test

### 3.3.1 일표본 t검정(onesample t.test)

하나의 표본을 이용하여 모집단의 특정한 값과 같은지 추론하는 방법이다. 만약에 음료수를 사먹는데, 평균용량이 1L라고 할 때, 진짜 평균적으로 1L인지 샘플을 추출하여 평균을 구하여 일표본 t검정을 수행하면 된다.

검정에 필요한 t값은 다음과 같이 계산한다.

는 평균, n:샘플사이즈, s:샘플의 표준편차

결과를 해석할 때에는 p값을 기준하여 설명하면 된다. $ p < .05$이면 귀무가설을 기각하고 대립가설을 받아들인다. 즉, 표본평균은 이론적 평균과 유의하게 다르다는 결론을 내릴 수 있다.

### 3.3.2 독립표본 t검정

* 두변수간 평균 비교
* 두 집단간 평균 비교

#### 3.3.2.1. 등분산성 검정

영가설: 등분산을 이루고 있다. p >.05.  
대립가설: 등분산을 이루지 않는다 p< .05.

등분산인 경우와 등분산이 아닌 경우에 사용하는 t값은 다르다.

로 사용한다.이때 공통분산(pooled variance)는

이다.

만약에 등분산이 아니라면

으로 사용한다. 자유도가 많이 복잡한 상태로 변경되어 사용된다.

##### var.test()

var.test는 F검정이라고도 한다.  
F Test to Compare Two Variances  
Performs an F test to compare the variances of two samples from normal populations.

var.test(mpg~vs,mtcars)

##   
## F test to compare two variances  
##   
## data: mpg by vs  
## F = 0.51515, num df = 17, denom df = 13, p-value = 0.1997  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.1714935 1.4353527  
## sample estimates:  
## ratio of variances   
## 0.5151485

F검정결과, 현재의 vs데이터에 대한 등분산 검정은 p값이 0.05보다 크므로, 귀무가설을 기각할 수 없다. 즉, 등분산을 가정할 수 있다.

##### bartlett.test

bartlett.test는 모수적 등분산 검정 방법이다. 비모수적 방법으로는fliger.test()를 사용한다.  
Bartlett 검정은 두 개 이상의 모집단의 분산이 동일한지 검정하는 모수적 검정 방법이다. 일반적으로 ANOVA 등 분산 분석을 수행하기 전에 가정하는 등분산성을 검증하는 데 사용된다. Bartlett검정의 일반적인 가정은 각 모집단은 정규 분포를 따른다는 것과 각 모집단의 변수는 서로 독립적이라는 사실이다.

Bartlett 검정의 검정 통계량은 다음과 같이 계산된다.

: 전체 표본 크기  
: i번째 그룹의 표본 크기  
: 전체 표본의 분산  
: i번째 그룹의 분산

Bartlett 검정의 가설은 다음과 같다.  
- 귀무가설 (): 모든 모집단의 분산은 동일합니다. - 대립가설 (): 모든 모집단의 분산은 동일하지 않습니다.

Bartlett 검정의 p-값은 검정 통계량 T가 분포에서 자유도를 가지고 발생할 확률이다. p 값이 유의수준 ()보다 작으면 귀무가설을 기각하고 대립가설을 채택한다.  
표본 크기가 작거나 모집단의 분포가 정규 분포를 따르지 않는 경우 Bartlett 검정의 결과가 정확하지 않을 수 있다.

bartlett.test(mpg~vs,mtcars)

##   
## Bartlett test of homogeneity of variances  
##   
## data: mpg by vs  
## Bartlett's K-squared = 1.5849, df = 1, p-value = 0.2081

bartlett.test결과, p값이 0.05보다 커서 귀무가설을 기각할 수 없다. 따라서, 두 변수는 등분산이라고 할 수 있다.

##### leveneTest()

car 패키지에는 Computes Levene’s test for homogeneity of variance across groups라고 설명되어 있다. 르벤테스트는 SPSS에서 사용하고 있어서 유명한 등분산 검정이다. Levene 검정은 두 개 이상의 모집단의 분산이 동일한지 검정하는 비모수적 검정 방법이다. ANOVA 등 분산 분석을 수행하기 전에 가정하는 등분산성을 검증하는 데 사용된다. Bartlett 검정과 달리 모집단의 분포에 대한 가정을 하지 않아 더 넓은 범위의 상황에서 사용할 수 있다.

Levene 검정의 검정 통계량은 다음과 같이 계산됩니다.

: 전체 표본 크기 : 그룹 수 : i번째 그룹의 표본 크기 : i번째 그룹의 평균 : i번째 그룹의 j번째 데이터 : 전체 표본의 평균

Levene의 가설  
귀무가설 (): 모든 모집단의 분산은 동일합니다.  
대립가설 (): 모든 모집단의 분산은 동일하지 않습니다.

Levene 검정의 p-값은 검정 통계량 W가 F 분포에서 , 자유도를 가지고 발생할 확률이다. p값이 유의수준 ()보다 작으면 귀무가설을 기각하고 대립가설을 채택한다.

mtcars에서 vs에 따른 mpg의 등분산 검정

car::leveneTest(mpg~vs,mtcars)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)  
## group 1 1.5922 0.2167  
## 30

#### 비모수검정Fligner-Killeen 검정 (Fligner-Killeen test)

Fligner-Killeen 검정은 두 개 이상의 모집단의 분산이 동일한지 검정하는 비모수적 검정 방법이다. Levene 검정과 유사하지만, 순위 통계를 사용하여 검정 통계량을 계산하기 때문에 더 정확한 결과를 제공할 수 있다. 특히, 표본 크기가 작거나 모집단의 분포가 정규 분포를 따르지 않는 경우 유용한 방법이다. 비모수 검정을 사용하는 경우 chi-square(카이제곱)으로 통계치가 나타난다.

Fligner-Killeen 검정의 검정 통계량은 다음과 같이 계산된다.

: 전체 표본 크기 : 그룹 수 : i번째 그룹의 표본 크기 : i번째 그룹의 j번째 데이터의 순위

귀무가설 (): 모든 모집단의 분산은 동일합니다.  
대립가설 (): 모든 모집단의 분산은 동일하지 않습니다.

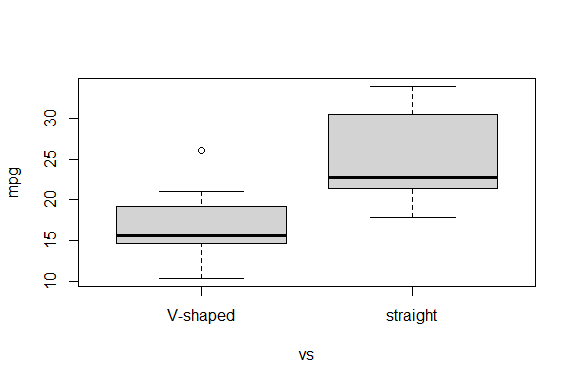
Fligner-Killeen 검정의 p-값은 검정 통계량 F가 χ^2 분포에서 자유도를 가지고 발생할 확률이다. p값이 유의수준 ()보다 작으면 귀무가설을 기각하고 대립가설을 채택한다.  
Fligner-Killeen 검정은 Levene 검정과 유사하지만, 순위 통계를 사용하기 때문에 더 정확한 결과를 제공할 수 있다. 표본 크기가 작거나 모집단의 분포가 정규 분포를 따르지 않는 경우 Fligner-Killeen 검정을 사용하는 것이 좋다.

fligner.test(mpg ~ vs,data= mtcars)

##   
## Fligner-Killeen test of homogeneity of variances  
##   
## data: mpg by vs  
## Fligner-Killeen:med chi-squared = 1.1765, df = 1, p-value = 0.2781

두변수의 차이를 시각화해보자. Straigth가 V-shaped보다 높은 값을 나타내고 있다.

plot(mpg ~ vs, data= mtcars)



##### t검정

등분산 검정(동질성 검정)결과, vs의 두 개의 레벨은 등분산이므로 var.equal = TRUE로 해서 분석을 한다.

t.test(mpg ~ vs, data= mtcars, var.equal = TRUE)

##   
## Two Sample t-test  
##   
## data: mpg by vs  
## t = -4.8644, df = 30, p-value = 3.416e-05  
## alternative hypothesis: true difference in means between group V-shaped and group straight is not equal to 0  
## 95 percent confidence interval:  
## -11.274221 -4.606732  
## sample estimates:  
## mean in group V-shaped mean in group straight   
## 16.61667 24.55714

보고서 작성

t.test(mpg ~ vs, data= mtcars, var.equal = TRUE) %>%   
 report::report()

## Effect sizes were labelled following Cohen's (1988) recommendations.  
##   
## The Two Sample t-test testing the difference of mpg by vs (mean in group  
## V-shaped = 16.62, mean in group straight = 24.56) suggests that the effect is  
## negative, statistically significant, and large (difference = -7.94, 95% CI  
## [-11.27, -4.61], t(30) = -4.86, p < .001; Cohen's d = -1.78, 95% CI [-2.61,  
## -0.92])

##### 등분산이 아닌 경우

등분산이 아닌 경우에는 var.equal = FALSE로 하게 되는데 이 경우에는 Two sample t-test가 아닌 Whelch Two t-test를 수행한다. 자유도가 바뀌어서 나타나게 된다.

t.test(mpg ~ vs, data= mtcars, var.equal = FALSE)

##   
## Welch Two Sample t-test  
##   
## data: mpg by vs  
## t = -4.6671, df = 22.716, p-value = 0.0001098  
## alternative hypothesis: true difference in means between group V-shaped and group straight is not equal to 0  
## 95 percent confidence interval:  
## -11.462508 -4.418445  
## sample estimates:  
## mean in group V-shaped mean in group straight   
## 16.61667 24.55714

다음은 report패키지를 이용하여 보고서를 작성한 경우를 나타낸 것이다.

t.test(mpg ~ vs, data= mtcars, var.equal = FALSE) %>%   
 report::report()

## Effect sizes were labelled following Cohen's (1988) recommendations.  
##   
## The Welch Two Sample t-test testing the difference of mpg by vs (mean in group  
## V-shaped = 16.62, mean in group straight = 24.56) suggests that the effect is  
## negative, statistically significant, and large (difference = -7.94, 95% CI  
## [-11.46, -4.42], t(22.72) = -4.67, p < .001; Cohen's d = -1.96, 95% CI [-2.94,  
## -0.95])

ex) 오렌지 주스와 비타민을 복용시킨 2개의 그룹에서 기니피그의 이빨길이는 통계적으로 유의미한 차이가 있는가?

A data frame with 60 observations on 3 variables.  
[,1] len numeric Tooth length  
[,2] supp factor Supplement type (VC or OJ).  
[,3] dose numeric Dose in milligrams/day

# Two-samples unpaired test  
data("ToothGrowth")  
ToothGrowth %>%str()

## 'data.frame': 60 obs. of 3 variables:  
## $ len : num 4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7 ...  
## $ supp: Factor w/ 2 levels "OJ","VC": 2 2 2 2 2 2 2 2 2 2 ...  
## $ dose: num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...

**1. 등분산 검정**

var.test(len ~ supp, ToothGrowth)

##   
## F test to compare two variances  
##   
## data: len by supp  
## F = 0.6386, num df = 29, denom df = 29, p-value = 0.2331  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.3039488 1.3416857  
## sample estimates:  
## ratio of variances   
## 0.6385951

귀무가설:H0:등분산이다.  
F검정결과 두 변수는 등분산을 충족하였다, *F*(29) = 0.64, *p* = 0.23.

t.test(len ~ supp, data = ToothGrowth, var.equal=TRUE)

##   
## Two Sample t-test  
##   
## data: len by supp  
## t = 1.9153, df = 58, p-value = 0.06039  
## alternative hypothesis: true difference in means between group OJ and group VC is not equal to 0  
## 95 percent confidence interval:  
## -0.1670064 7.5670064  
## sample estimates:  
## mean in group OJ mean in group VC   
## 20.66333 16.96333

**APA보고**  
기니피그의 supplement(오랜지주스와 비타민C)이 차이에 따른 이빨길이의 차이에 대하여 독립표본t검정을 시행한 결과 유의하지않았다, *t*(58) = 1.92, *p* =.06.

**report package**

t.test(len ~ supp, data = ToothGrowth, var.equal=TRUE) %>%   
 report::report()

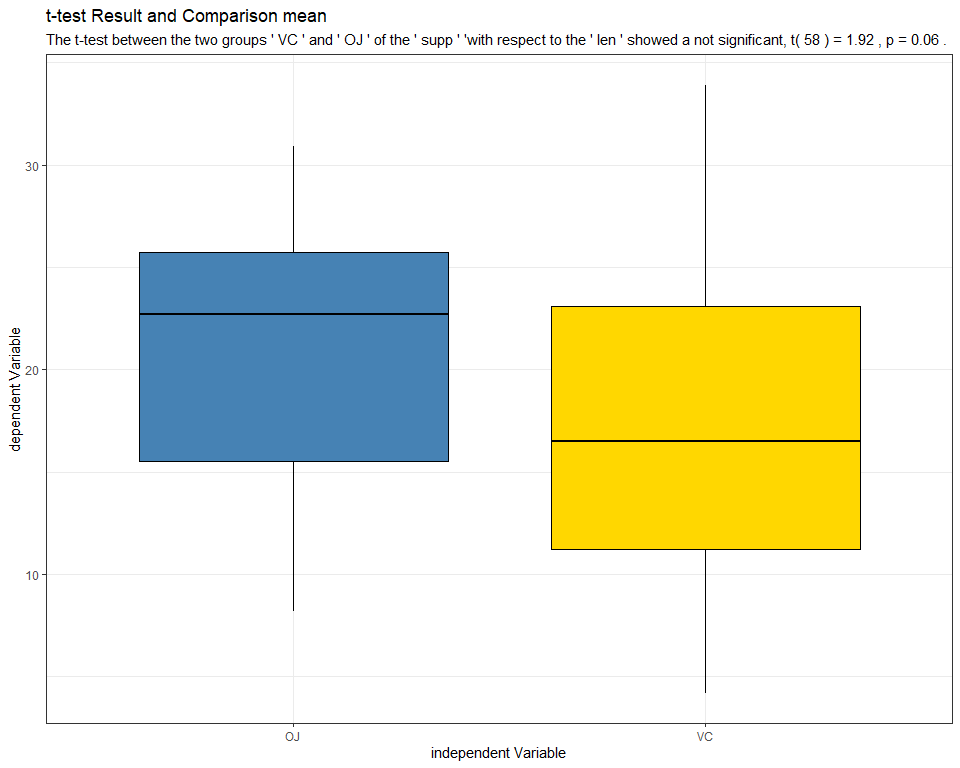
## Effect sizes were labelled following Cohen's (1988) recommendations.  
##   
## The Two Sample t-test testing the difference of len by supp (mean in group OJ =  
## 20.66, mean in group VC = 16.96) suggests that the effect is positive,  
## statistically not significant, and medium (difference = 3.70, 95% CI [-0.17,  
## 7.57], t(58) = 1.92, p = 0.060; Cohen's d = 0.50, 95% CI [-0.02, 1.02])

#### t.test 요약보고 함수

이번에는 필자가 개발한 jjstat package에 있는 t\_test\_report()함수를 이용하여 독립표본 t검정을 진행해보자. 이 함수는 먼저 var.test를 통하여 분산의 동질성 검정을 시행한다. 그리고 var.test의 결과에 대한 결과보고를 해준다. 이후에는 t.test()를 수행한 후에 t.test의 결과에 대한 보고를 해준다. 그리고 평균을 비교하는 것을 시각화하기 위해서 boxplot을 그려서 보여주는 함수이다.  
위에서 배운 개념을 모두 활용하는 것들을 모아서 보여주는 함수이다. 분석을 하면서 여러가지를 편리하게 만들어주는데 중요한 것은 t.test에 대한 깊은 이해를 한 후에 나타나는 결과는 매우 편리하게 만들어주지만, 정확한 이해없이 사용하면 그저 복잡한 결과만 출력되는 것이 된다. 따라서, 명확하게 인식하고 사용하는 것이 중요하다.

t\_test\_report(data = ToothGrowth, iv= "supp", dv="len", translate = FALSE)

## $var\_test\_result\_tibble  
## # A tibble: 2 × 4  
## Var df f\_value p\_value  
## <fct> <int> <dbl> <dbl>  
## 1 VC 29 0.639 0.233  
## 2 OJ 29 0.639 0.233  
##   
## $var\_test\_report  
## [1] "The t-test results between the two groups 'VC' and 'OJ' of the independent variable 'supp' with respect to the dependent variable 'len' showed anot statistically significant. F(29,29) = 0.64, p = 0.23. This indicates that the variances of the two groups are equal."  
##   
## $t\_test\_result\_tibble  
## # A tibble: 1 × 9  
## DV IV IV.1 IV.2 df t.value p.value sig test\_type   
## <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <chr> <chr>   
## 1 len supp VC(20.66) OJ(16.96) 58 1.92 0.0604 ns student's t.test  
##   
## $t\_test\_report  
## [1] "The t-test results between the two groups 'VC' and 'OJ' of the independent variable ' supp ''with respect to the dependent variable 'len' showed anot statistically significant. t(58) = 1.92,p = 0.06."  
##   
## $descriptive  
## # A tibble: 2 × 2  
## supp len[,"n"] [,"mean"] [,"sd"]  
## <fct> <dbl> <dbl> <dbl>  
## 1 OJ 30 20.7 6.61  
## 2 VC 30 17.0 8.27  
##   
## $boxplot



### 3.3.3. 대응표본 t검정

대응표본 t검정은 주로 실험에서 많이 쓰이는데, 어떤 처치에 대한 효과를 알아보고자 할 때 사용하는 경우가 많다.

예를 들면, 아래의 데이터는 시험데이터이다.  
학생이 중간고사를 본 후에 학원을 다녔다. 이후에 기말고사를 보았다. 학원을 다닌 후에 성적의 차이가 유의하게 나타났는지 검정해보고자 하였다.

대응표본 t-test는 분산의 동질성 검정을 하지 않는다. 같은 그룹에서 표집을 진행했기 때문이다.

표준오차는

으로 계산한다. t의 임계값을 구하여 유의성을 검정하거나 p값을 구하여 통계적 유의성을 검정한다.

ex) 시험성적은 학원을 다닌후에 유의하게 향상되었을까?

| mid | final |
| --- | --- |
| 16 | 19 |
| 20 | 22 |
| 21 | 24 |
| 22 | 24 |
| 23 | 25 |
| 22 | 25 |
| 27 | 26 |
| 25 | 26 |
| 27 | 28 |
| 28 | 32 |

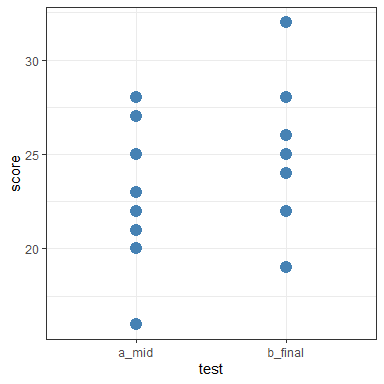
이 데이터를 롱포맷으로 만들어서 분석해보자.

exam\_long = data.frame(  
 test = c(rep("a\_mid", 10), rep("b\_final", 10)),  
 score = c(16, 20, 21, 22, 23, 22, 27, 25, 27, 28,  
 19, 22, 24, 24, 25, 25, 26, 26, 28, 32)  
)  
exam\_long

## test score  
## 1 a\_mid 16  
## 2 a\_mid 20  
## 3 a\_mid 21  
## 4 a\_mid 22  
## 5 a\_mid 23  
## 6 a\_mid 22  
## 7 a\_mid 27  
## 8 a\_mid 25  
## 9 a\_mid 27  
## 10 a\_mid 28  
## 11 b\_final 19  
## 12 b\_final 22  
## 13 b\_final 24  
## 14 b\_final 24  
## 15 b\_final 25  
## 16 b\_final 25  
## 17 b\_final 26  
## 18 b\_final 26  
## 19 b\_final 28  
## 20 b\_final 32

이를 시각화하여 그림으로 나타내보자. 각 성적의 변화를 보자.

exam\_long %>%   
 ggplot(aes(x=test, y= score))+  
 geom\_point(size=4, color= "steelblue")+  
 theme\_bw()



t.test(score ~ test, data= exam\_long, paired=TRUE)

##   
## Paired t-test  
##   
## data: score by test  
## t = -4.4721, df = 9, p-value = 0.00155  
## alternative hypothesis: true mean difference is not equal to 0  
## 95 percent confidence interval:  
## -3.0116674 -0.9883326  
## sample estimates:  
## mean difference   
## -2

# exam\_long

paired t.test결과, 학원을 다닌후에 성적을 유의하게 올랐다고 할 수 있다, t(9) =-4.47, p = .002.

데이터가 롱포맷이면 수식을 이용하여 분석한다. 종속변수 ~ 독립변수로 입력하면된다.

library(report)  
t.test(score ~ test, data= exam\_long, paired=TRUE) %>%   
 report()

## Effect sizes were labelled following Cohen's (1988) recommendations.  
##   
## The Paired t-test testing the difference of score by test (mean difference =  
## -2.00) suggests that the effect is negative, statistically significant, and  
## large (difference = -2.00, 95% CI [-3.01, -0.99], t(9) = -4.47, p = 0.002;  
## Cohen's d = -1.49, 95% CI [-2.41, -0.53])

##### 데이터가 와이드 포맷인 경우

데이터가 와이드포맷일 때에는 변수명을 선택하여 분석하면 된다.

exam = cbind.data.frame(  
 mid = c(16, 20, 21, 22, 23, 22, 27, 25, 27, 28),  
 final = c(19, 22, 24, 24, 25, 25, 26, 26, 28, 32)  
)  
exam

## mid final  
## 1 16 19  
## 2 20 22  
## 3 21 24  
## 4 22 24  
## 5 23 25  
## 6 22 25  
## 7 27 26  
## 8 25 26  
## 9 27 28  
## 10 28 32

t.test(변수1, 변수2, paired=TRUE)의 형식으로 분석하면 된다.

t.test(exam$mid, exam$final, paired= TRUE)

##   
## Paired t-test  
##   
## data: exam$mid and exam$final  
## t = -4.4721, df = 9, p-value = 0.00155  
## alternative hypothesis: true mean difference is not equal to 0  
## 95 percent confidence interval:  
## -3.0116674 -0.9883326  
## sample estimates:  
## mean difference   
## -2

paired t.test결과, 학원을 다닌후에 성적을 유의하게 올랐다고 할 수 있다, t(9) =-4.47, p = .002.

## 3.4. oneway ANOVA

### 3.4.1. oneway ANOVA

일원배치 분산분석을 실행하면서 분석의 기법를 이해보도록 하자.

**데이터**  
데이터를 잘 이해하는 것은 분석할 때 매우 유리하다. 그룹은 총 4개로 하여 성적을 측정하였다.

# 01 oneway anova ------------------------------------------------------------  
fuel = data.frame(  
 trt = c(1L,1L,1L,1L,1L,1L,2L,  
 2L,2L,2L,2L,2L,3L,3L,3L,3L,3L,3L,4L,4L,4L,  
 4L,4L,4L),  
 y = c(64L,72L,68L,77L,56L,95L,  
 78L,91L,97L,82L,85L,77L,75L,93L,78L,71L,63L,  
 76L,55L,66L,49L,64L,70L,68L)  
 )  
  
fuel |> head()

## trt y  
## 1 1 64  
## 2 1 72  
## 3 1 68  
## 4 1 77  
## 5 1 56  
## 6 1 95

#### 데이터의 구조보기

fuel$trt <- as.factor(fuel$trt) #변수의 factor처리   
fuel %>% str()

## 'data.frame': 24 obs. of 2 variables:  
## $ trt: Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 2 2 2 2 ...  
## $ y : int 64 72 68 77 56 95 78 91 97 82 ...

데이터를 wide format으로 변경하여 보자. 일반적으로 프로그램에서 보는 형태이다.

# wideData  
fuel %>%   
 mutate(rowid= rep(1:6,4)) %>%  
 pivot\_wider(values\_from = y, names\_from = trt)

## # A tibble: 6 × 5  
## rowid `1` `2` `3` `4`  
## <int> <int> <int> <int> <int>  
## 1 1 64 78 75 55  
## 2 2 72 91 93 66  
## 3 3 68 97 78 49  
## 4 4 77 82 71 64  
## 5 5 56 85 63 70  
## 6 6 95 77 76 68

#### 기술통계 분석

기술통계를 분석하는 여러가지 방법이 있지만, 분석을 하다보면 가장 빠르고 편리하게 사용하는 방법이어서 소개한다.

#statistical describe   
fuel %>%  
 group\_by(trt) %>%   
 summarise(M=mean(y), SD=sd(y),N=n(),  
 Median=median(y), Max=max(y),Min=min(y),  
 skewness=SKEW(y),  
 kurtosis=KURT(y),  
 SE=sd(y)/sqrt(n()),  
 lowerCI = mean(y)-1.96\*SE,  
 upperCI = mean(y)+1.96\*SE  
 ) %>% kable("pandoc", digits = 2, caption="기술통계분석")

기술통계분석

| trt | M | SD | N | Median | Max | Min | skewness | kurtosis | SE | lowerCI | upperCI |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 72 | 13.34 | 6 | 70.0 | 95 | 56 | 0.53 | -1.16 | 5.45 | 61.32 | 82.68 |
| 2 | 85 | 7.77 | 6 | 83.5 | 97 | 77 | 0.38 | -1.69 | 3.17 | 78.78 | 91.22 |
| 3 | 76 | 9.88 | 6 | 75.5 | 93 | 63 | 0.45 | -1.03 | 4.03 | 68.09 | 83.91 |
| 4 | 62 | 8.22 | 6 | 65.0 | 70 | 49 | -0.52 | -1.66 | 3.36 | 55.42 | 68.58 |

참고로 패키지를 사용하면 다음과 같은 방법을 사용하기도 한다.

#the other packages psych   
jjstat::mysummaryBy( y ~ trt, data= fuel)

## # A tibble: 4 × 8  
## var Mean SD N Min Max Skew Kurt  
## <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 72 13.3 6 56 95 0.535 -1.16  
## 2 2 85 7.77 6 77 97 0.377 -1.69  
## 3 3 76 9.88 6 63 93 0.449 -1.03  
## 4 4 62 8.22 6 49 70 -0.522 -1.66

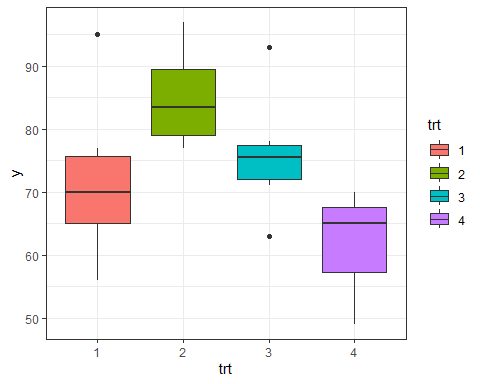
##same method   
# fuel |> mysummaryBy(formula = y~ trt)  
  
#다른방법   
# psych::describeBy(fuel ~ trt)

#### 평균비교 시각화

평균을 비교할 때, 가장 눈에 쉽게 비교해주는 방법은 시각화하는 방법이다.

\*\* A boxplot is a standardized way of displaying the dataset based on the five-number summary: the minimum, the maximum, the sample median, and the first and third quartiles. Minimum (Q0 or 0th percentile): the lowest data point in the data set excluding any outliers(from WIKI).

# boxplot visualization   
fuel %>%   
 ggplot(aes(x=trt, y=y, fill=trt))+  
 geom\_boxplot()+  
 theme\_bw()



### 3.4.1. ANOVA process

ANOVA를 하는 방법은 매우 간단하다. 이렇게 간단해도 될 정도인가?라 고 생각이 들정도로 사용방법은 너무나 간단하다. 따라서, 코드를 명확히 익히고 분석하는 것이 필요하고, 무엇보다 분산분석표를 이해하는 것이 매우 중요하다.

aov(y~ trt, data=fuel) %>% summary()

## Df Sum Sq Mean Sq F value Pr(>F)   
## trt 3 1636 545.5 5.406 0.00688 \*\*  
## Residuals 20 2018 100.9   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

H0: 4개 그룹의 평균은 같다.  
H1: not H0  
일원분산 분석결과, trt의 4가지 그룹은 통계적으로 유의한 차이가 나타났다, F(3, )= 5.41, p =.01.

<**참고**>  
lm()함수와 anova()함수를 이용하여 분석하는 방법도 있다. lm()은 회귀분석(regression)을 수행하는 함수이다. 그러나 독립변수가 명목형 변수인 경우에는 일반화 선형 모형에 근거하면 aov()와 같음 함수가 된다. 이 부분은 별도의 부록에 설명해두었다.

lm(y~ trt, data=fuel) %>% anova()

## Analysis of Variance Table  
##   
## Response: y  
## Df Sum Sq Mean Sq F value Pr(>F)   
## trt 3 1636.5 545.5 5.4063 0.006876 \*\*  
## Residuals 20 2018.0 100.9   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

### 3.4.1. Post Hoc(사후분석석)

ANOVA가 통계적으로 유의하였으므로 이제는 어떤 그룹간 차이가 발생했지는 확인해봐야 한다. 이러한 과정을 사후분석(POST HOC)이라고 한다.

#post HOC  
aov(y~ trt, data=fuel) %>% TukeyHSD()

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = y ~ trt, data = fuel)  
##   
## $trt  
## diff lwr upr p adj  
## 2-1 13 -3.232221 29.232221 0.1461929  
## 3-1 4 -12.232221 20.232221 0.8998057  
## 4-1 -10 -26.232221 6.232221 0.3378150  
## 3-2 -9 -25.232221 7.232221 0.4270717  
## 4-2 -23 -39.232221 -6.767779 0.0039064  
## 4-3 -14 -30.232221 2.232221 0.1065573

이 결과를 좀더 tidy하게 정리해보면 다음과 같다.  
분석을 정리한 후에 유의성을 표시하였다.

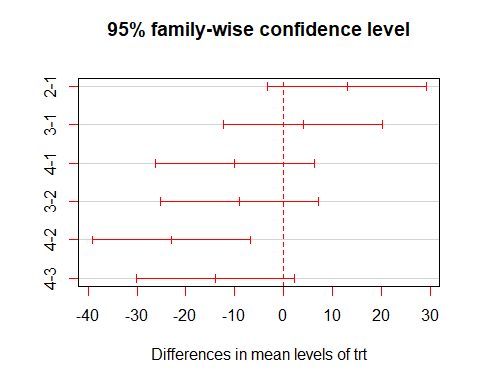
#broom packages의 tidy  
aov(y~ trt, data=fuel) %>% TukeyHSD() %>% broom::tidy() %>%   
 dplyr::select(2,4,5,6,7) %>%   
 rename(p.value=adj.p.value) %>%  
 p\_mark\_sig() %>%   
 tibble()

## # A tibble: 6 × 6  
## contrast estimate conf.low conf.high p.value sig   
## <chr> <dbl> <dbl> <dbl> <dbl> <chr>  
## 1 2-1 13.0 -3.23 29.2 0.146 ns   
## 2 3-1 4.00 -12.2 20.2 0.900 ns   
## 3 4-1 -10.0 -26.2 6.23 0.338 ns   
## 4 3-2 -9 -25.2 7.23 0.427 ns   
## 5 4-2 -23 -39.2 -6.77 0.00391 \*\*   
## 6 4-3 -14 -30.2 2.23 0.107 ns

그룹 4와 2가 통계적으로 유의한 차이가 나타났다, diff = -23, 95%CI[-39.23,-6.76], *p* = .003.

보통은 이렇게 정리하지 않고 시각화 하여 유의성을 보기도 한다. 이때 plot()만 추가하면 된다.  
신뢰구간을 이용하여 나타낸 것이므로 0을 신뢰구간이 0을 포함하지 않는 경우에 유의하다고 볼 수 있다.

#plot   
aov(y~ trt, data=fuel) %>% TukeyHSD() %>% plot(col="red")



### 3.4.2. Planned Contrast

사후분석을 위해서는 보통 emmeans를 사용하는 경우가 많다. 가장 일반적이라고 할 수 있다. 물론 방법은 다양하지만 기억하기 쉽고 사용하기가 쉽다. 먼저 marginal mean을 구한다. 그후에 대비계수를 만들고, 대비검정을 수행하면 된다(대비검정이나 계획비교를 혼용하여 사용하는 중인데, 같은 의미로 보면 된다).

#marginal mean   
library(emmeans)  
  
aov(y~ trt, data=fuel) %>% emmeans("trt")

## trt emmean SE df lower.CL upper.CL  
## 1 72 4.1 20 63.4 80.6  
## 2 85 4.1 20 76.4 93.6  
## 3 76 4.1 20 67.4 84.6  
## 4 62 4.1 20 53.4 70.6  
##   
## Confidence level used: 0.95

# lm(y~ trt, data=fuel) %>% emmeans("trt") #same method

계획해서 비교하고 싶은 것은 그룹2 vs 그룹1, 3, 4의 합이다. 그리고, 그룹1+그룹1 vs 그룹3+그룹4를 비교해보자.  
그런데, TukeyHSD를 함께 보여주기 위해서 contrast계수를 모두 설정했다. 대비계수를 설정할 때 중요한 것은 수직대비(Othogonal cantrast)가 되어야 한다.

contrast\_coef = list(  
 "2-1"= c(-1, 1, 0, 0),  
 "3-1"= c(-1, 0, 1, 0),  
 "4-1"= c(-1, 0, 0, 1),  
 "3-2"= c( 0,-1, 1, 0),  
 "4-2"= c( 0,-1, 0, 1),  
 "4-3"= c( 0, 0,-1, 1),  
 "m1-m2+m3+m4"= c(1,-3, 1,1), # 검증하고자 하는 가설 1  
 "m1+m2-m3-m4"= c(1,1,-1,-1) # 검증하고자 하는 가설 2   
)  
contrast\_coef

## $`2-1`  
## [1] -1 1 0 0  
##   
## $`3-1`  
## [1] -1 0 1 0  
##   
## $`4-1`  
## [1] -1 0 0 1  
##   
## $`3-2`  
## [1] 0 -1 1 0  
##   
## $`4-2`  
## [1] 0 -1 0 1  
##   
## $`4-3`  
## [1] 0 0 -1 1  
##   
## $`m1-m2+m3+m4`  
## [1] 1 -3 1 1  
##   
## $`m1+m2-m3-m4`  
## [1] 1 1 -1 -1

#planned contrast analysis   
aov(y~ trt, data=fuel) %>% emmeans("trt") %>%   
 contrast(contrast\_coef,  
 adjust="bonferroni") %>% p\_mark\_sig()

## # A tibble: 8 × 7  
## contrast estimate SE df t.ratio p.value sig   
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>  
## 1 2-1 13.0 5.80 20 2.24 0.292 ns   
## 2 3-1 4.00 5.80 20 0.690 1 ns   
## 3 4-1 -10.0 5.80 20 -1.72 0.801 ns   
## 4 3-2 -9.00 5.80 20 -1.55 1 ns   
## 5 4-2 -23 5.80 20 -3.97 0.00610 \*\*   
## 6 4-3 -14 5.80 20 -2.41 0.204 ns   
## 7 m1-m2+m3+m4 -45.0 14.2 20 -3.17 0.0387 \*   
## 8 m1+m2-m3-m4 19 8.20 20 2.32 0.250 ns

4-2에 관한 차이가 유의했고, 가설1은 통계적으로 유의하였다.

그리고 p값에 대한 adjust가 적용되지 않으면 유의한 것들이 더 생겨나게 된다. 그렇다면 채택되지 말아야 할 귀무가설이 채택되는지 확인해보자. p값에 대한 보정을 하지 않고 “none”로 한 경우이다. 4-3도 유의하게 나타났고, 가설2도 통계적으로 유의하게 나타났다. bonferroni 교정의 경우는 유의수준을 총 검정의 수(N)로 나누어서 개별 검정을 보정를 하는 방법이다.

#planned contrast analysis   
aov(y~ trt, data=fuel) %>% emmeans("trt") %>%   
 contrast(contrast\_coef,  
 adjust="none") %>% p\_mark\_sig()

## # A tibble: 8 × 7  
## contrast estimate SE df t.ratio p.value sig   
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>  
## 1 2-1 13.0 5.80 20 2.24 0.0365 \*   
## 2 3-1 4.00 5.80 20 0.690 0.498 ns   
## 3 4-1 -10.0 5.80 20 -1.72 0.100 ns   
## 4 3-2 -9.00 5.80 20 -1.55 0.136 ns   
## 5 4-2 -23 5.80 20 -3.97 0.000762 \*\*\*   
## 6 4-3 -14 5.80 20 -2.41 0.0255 \*   
## 7 m1-m2+m3+m4 -45.0 14.2 20 -3.17 0.00484 \*\*   
## 8 m1+m2-m3-m4 19 8.20 20 2.32 0.0313 \*

#### 95%신뢰구간 산정과 시각화

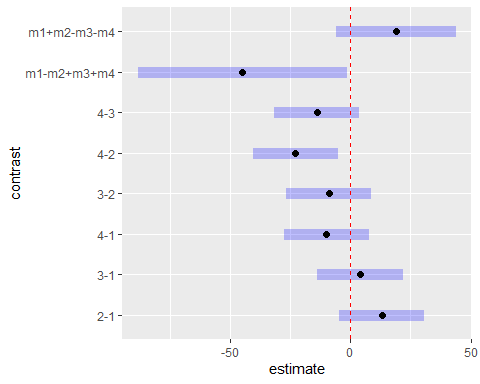
신뢰구간과 유의성을 표시해준다.

#confidence interval   
aov(y~ trt, data=fuel) %>% emmeans("trt") %>%   
 contrast(contrast\_coef,  
 adjust="bonferroni", ) %>% confint()

## contrast estimate SE df lower.CL upper.CL  
## 2-1 13 5.8 20 -4.72 30.72  
## 3-1 4 5.8 20 -13.72 21.72  
## 4-1 -10 5.8 20 -27.72 7.72  
## 3-2 -9 5.8 20 -26.72 8.72  
## 4-2 -23 5.8 20 -40.72 -5.28  
## 4-3 -14 5.8 20 -31.72 3.72  
## m1-m2+m3+m4 -45 14.2 20 -88.40 -1.60  
## m1+m2-m3-m4 19 8.2 20 -6.06 44.06  
##   
## Confidence level used: 0.95   
## Conf-level adjustment: bonferroni method for 8 estimates

#### 신뢰구간의 유의성

#plotting   
aov(y~ trt, data=fuel) %>% emmeans("trt") %>%   
 contrast(contrast\_coef,  
 adjust="bonferroni", ) %>%   
 plot()+  
 geom\_vline(xintercept = 0, col= "red", linetype=2)



#### 계획비교를 TukeyHSD처럼 하고 싶을 때, pairs()

TukeyHSD()는 쌍으로 두개의 변수끼리 차를 이용하여 비교하는 것이다. 이러한 방법을 emmeans에서도 사용할 수 있는데 이때 사용하는 함수가 pairs()이다.

# similarity of TukeyHSD  
aov(y~ trt, data=fuel) %>% emmeans("trt") %>%   
 pairs(simple = "trt") %>% p\_mark\_sig()

## # A tibble: 6 × 7  
## contrast estimate SE df t.ratio p.value sig   
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>  
## 1 trt1 - trt2 -13.0 5.80 20 -2.24 0.146 ns   
## 2 trt1 - trt3 -4.00 5.80 20 -0.690 0.900 ns   
## 3 trt1 - trt4 10.0 5.80 20 1.72 0.338 ns   
## 4 trt2 - trt3 9.00 5.80 20 1.55 0.427 ns   
## 5 trt2 - trt4 23 5.80 20 3.97 0.00391 \*\*   
## 6 trt3 - trt4 14 5.80 20 2.41 0.107 ns

이 방법을 이용하여 쉽게 TukeyHSD와 같은 비교를 할 수 있다. TukeyHSD()가 p값을 adjust가 되지 않고 자동 적용이 된다. 이런 경우 pairs()를 이용하면 p값을 조절해서 확인해볼 수 있다.

# similarity of TukeyHSD  
aov(y~ trt, data=fuel) %>% emmeans("trt") %>%   
 pairs(simple = "trt", adjust="none") %>% p\_mark\_sig() %>% tibble()

## # A tibble: 6 × 7  
## contrast estimate SE df t.ratio p.value sig   
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>  
## 1 trt1 - trt2 -13.0 5.80 20 -2.24 0.0365 \*   
## 2 trt1 - trt3 -4.00 5.80 20 -0.690 0.498 ns   
## 3 trt1 - trt4 10.0 5.80 20 1.72 0.100 ns   
## 4 trt2 - trt3 9.00 5.80 20 1.55 0.136 ns   
## 5 trt2 - trt4 23 5.80 20 3.97 0.000762 \*\*\*   
## 6 trt3 - trt4 14 5.80 20 2.41 0.0255 \*

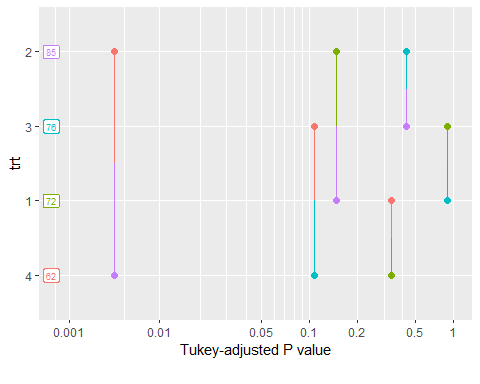
**사후분석을 행렬로 표현** 사후분석을 행렬 형태로 분석하는 방법은 누군가에게는 익숙할 수 있다. 그 이유는 예전 사후분석 패키지의 방식이 행렬을 통해서 나타내는 경우가 종종 있었기 때문이다.  
행렬의 하단에는 diff값을 나타내고, 상단에는 유의성을 나타내는 p값이 있다.

aov(y~ trt, data=fuel) %>% emmeans("trt") %>% pwpm()

## 1 2 3 4  
## 1 [72] 0.1462 0.8998 0.3378  
## 2 -13 [85] 0.4271 0.0039  
## 3 -4 9 [76] 0.1066  
## 4 10 23 14 [62]  
##   
## Row and column labels: trt  
## Upper triangle: P values adjust = "tukey"  
## Diagonal: [Estimates] (emmean)   
## Lower triangle: Comparisons (estimate) earlier vs. later

**시각화**  
그림을 보면 다음과 같다. x축에는 p값을 나타냈고, y축에는 각 그룹의 평균과 연결된 부분을 나타냈다. 각 선 그래프는 p값의 위치가 어디에서 나타나는지 구체적으로 나타내고 있어서 이해하기 쉽다는 장점이 있다.

aov(y~ trt, data=fuel) %>% emmeans("trt") %>% pwpp()



###3.4.3. multicomp::glht()를 이용한 계획비교

이번에는 다른 방법을 더 소개한다. glht(general linear hypothesis test)를 사용하여 계획비교를 하는 방법을 알아보고자 한다. 계획비교를 하는 방법을 먼저 one way ANOVA를 설명하면서 emmeans를 이용하는 방법을 설명하였다. 이번에는 좀 다른 방법을 설명하고자 한다. 전체적인 맥락은 비슷하기 때문에 어려움은 없을 것으로 예상된다.

사용할 데이터는 기존과 동일하다.

fuel = data.frame(  
 trt = c(1L,1L,1L,1L,1L,1L,2L,  
 2L,2L,2L,2L,2L,3L,3L,3L,3L,3L,3L,4L,4L,4L,  
 4L,4L,4L),  
 y = c(64L,72L,68L,77L,56L,95L,  
 78L,91L,97L,82L,85L,77L,75L,93L,78L,71L,63L,  
 76L,55L,66L,49L,64L,70L,68L)  
)  
# factor processing of variable  
fuel$trt <- as.factor(fuel$trt)  
fuel %>% str()

## 'data.frame': 24 obs. of 2 variables:  
## $ trt: Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 2 2 2 2 ...  
## $ y : int 64 72 68 77 56 95 78 91 97 82 ...

분산분석의 실행  
분산분석을 통해서 값을 정확히 알아보자

library(multcomp) #gjht  
aov(y~ trt, data=fuel) %>% summary()

## Df Sum Sq Mean Sq F value Pr(>F)   
## trt 3 1636 545.5 5.406 0.00688 \*\*  
## Residuals 20 2018 100.9   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**대비계수의 설정**  
여기에서는 대비계수를 설정할 때 행렬로 만든다. rbind()이용하여 결합해둔다. (참고로 emmeans에서는 list()데이터로 만들었었다)

# define contrast matrix directly  
contrast\_coef2 = rbind(  
 "2-1"= c(-1, 1, 0, 0),  
 "3-1"= c(-1, 0, 1, 0),  
 "4-1"= c(-1, 0, 0, 1),  
 "3-2"= c( 0,-1, 1, 0),  
 "4-2"= c( 0,-1, 0, 1),  
 "4-3"= c( 0, 0,-1, 1),  
 "m1-m2+m3+m4"= c(1,-3, 1,1),  
 "m1+m2-m3-m4"= c(1,1,-1,-1)  
) %>% `colnames<-`(c("m1","m2","m3","m4"))  
  
contrast\_coef2

## m1 m2 m3 m4  
## 2-1 -1 1 0 0  
## 3-1 -1 0 1 0  
## 4-1 -1 0 0 1  
## 3-2 0 -1 1 0  
## 4-2 0 -1 0 1  
## 4-3 0 0 -1 1  
## m1-m2+m3+m4 1 -3 1 1  
## m1+m2-m3-m4 1 1 -1 -1

<*참고설명*> colnames<-(c(“m1”,“m2”,“m3”,“m4”))부분은 컬럼명을 설정하는 방법이다. %>%를 사용할 때 이 방법이 생각보다 더 유용한 방법이 된다. 이 함수는 R의 base함수중 하나이다.

**계획비교(Contrast)**  
분산분석후 glht()함수를 실행한다. linfct옵션에 mcp()에 factor변수 = 대비계수를 설정해주는 방법으로 설정해주면 된다.

aov(y~ trt, data=fuel) %>%   
 glht(linfct = mcp(trt = contrast\_coef2)) %>% summary()

##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Multiple Comparisons of Means: User-defined Contrasts  
##   
##   
## Fit: aov(formula = y ~ trt, data = fuel)  
##   
## Linear Hypotheses:  
## Estimate Std. Error t value Pr(>|t|)   
## 2-1 == 0 13.000 5.799 2.242 0.15519   
## 3-1 == 0 4.000 5.799 0.690 0.90392   
## 4-1 == 0 -10.000 5.799 -1.724 0.35072   
## 3-2 == 0 -9.000 5.799 -1.552 0.44009   
## 4-2 == 0 -23.000 5.799 -3.966 0.00452 \*\*  
## 4-3 == 0 -14.000 5.799 -2.414 0.11392   
## m1-m2+m3+m4 == 0 -45.000 14.206 -3.168 0.02534 \*   
## m1+m2-m3-m4 == 0 19.000 8.202 2.317 0.13581   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## (Adjusted p values reported -- single-step method)

4-2의 차이가 유의하게 나타났으며, 가설1번(m1-m2+m3+m4 == 0 )이 유의하게 나타났다.

자유도를 구해보면, 20이 되었다.

# fuel의 개수 24, trt factor variable level은 4개(1,2,3,4)  
nrow(fuel)-length(levels(fuel$trt))

## [1] 20

이는 상당히 유용한 정보이다. 무슨 말이냐면, 두 개씩 비교하는 경우(t.test)의 경우는 자유도가 5일수 밖에 없다. 그러나 지금은 자유도가 높게 나타난다는 것은 표준오차의 공식에서 에서 n값이 커졌다는 의미가 된다. 이는 표준오차를 작아지게 만든다는 것을 알려주기 때문이다.

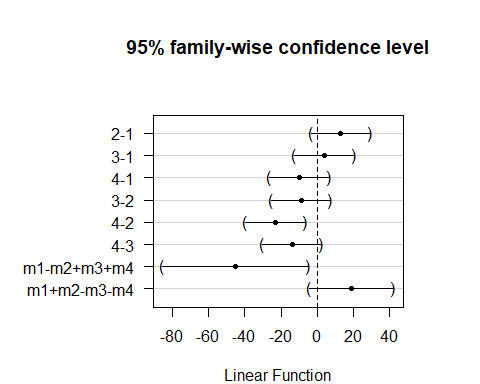
신뢰구간 을 나타내기

#신뢰구간   
aov(y~ trt, data=fuel) %>%   
 glht(linfct = mcp(trt = contrast\_coef2)) %>% confint()

##   
## Simultaneous Confidence Intervals  
##   
## Multiple Comparisons of Means: User-defined Contrasts  
##   
##   
## Fit: aov(formula = y ~ trt, data = fuel)  
##   
## Quantile = 2.8399  
## 95% family-wise confidence level  
##   
##   
## Linear Hypotheses:  
## Estimate lwr upr   
## 2-1 == 0 13.0000 -3.4700 29.4700  
## 3-1 == 0 4.0000 -12.4700 20.4700  
## 4-1 == 0 -10.0000 -26.4700 6.4700  
## 3-2 == 0 -9.0000 -25.4700 7.4700  
## 4-2 == 0 -23.0000 -39.4700 -6.5300  
## 4-3 == 0 -14.0000 -30.4700 2.4700  
## m1-m2+m3+m4 == 0 -45.0000 -85.3431 -4.6569  
## m1+m2-m3-m4 == 0 19.0000 -4.2921 42.2921

시각화

par(mar=c(4,8,6,4))  
aov(y~ trt, data=fuel) %>%   
 glht(linfct = mcp(trt = contrast\_coef2)) %>% plot()



**Tukey를 사용하는 방법**

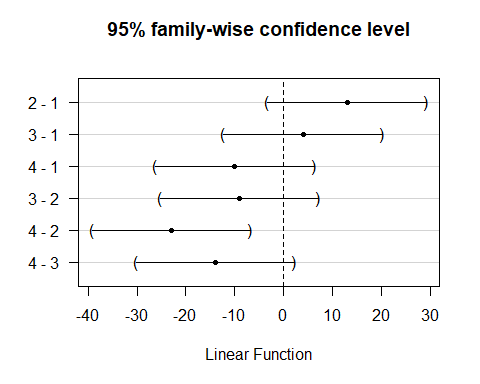
glht()에서는 Tukey를 사용하는 방법이 mcp()에서 명목변수인 trt에 “Tukey”를 설정하는 방법이다.

# Tukey   
aov(y~ trt, data=fuel) %>%   
 glht(linfct = mcp(trt = "Tukey")) %>% summary()

##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Multiple Comparisons of Means: Tukey Contrasts  
##   
##   
## Fit: aov(formula = y ~ trt, data = fuel)  
##   
## Linear Hypotheses:  
## Estimate Std. Error t value Pr(>|t|)   
## 2 - 1 == 0 13.000 5.799 2.242 0.14611   
## 3 - 1 == 0 4.000 5.799 0.690 0.89981   
## 4 - 1 == 0 -10.000 5.799 -1.724 0.33789   
## 3 - 2 == 0 -9.000 5.799 -1.552 0.42707   
## 4 - 2 == 0 -23.000 5.799 -3.966 0.00382 \*\*  
## 4 - 3 == 0 -14.000 5.799 -2.414 0.10672   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## (Adjusted p values reported -- single-step method)

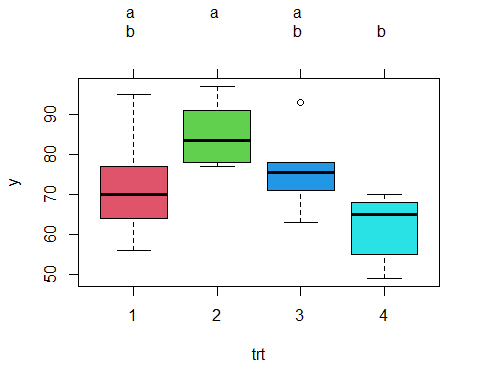
Tukey이용한 경우의 시각화

aov(y~ trt, data=fuel) %>%   
 glht(linfct = mcp(trt = "Tukey")) %>% plot()



Boxplot을 이용한 사후분석 결과 표시 이 방법도 매우 유용하며 의미가 있다.  
Boxplot에서 유의한 것이 무엇인지 보여주는 방식이다. cld()와 plot()을 이용하여 나타낸다.

#bocplot   
aov(y~ trt, data=fuel) %>%   
 glht(linfct = mcp(trt = "Tukey")) %>%   
 cld(level = 0.05) %>% plot(col=c(2:5))



여기에서는 유의성 표시를 a, b 등을 이용하여 나타낸 것이다. ab는 서로 구분되지 않는다는 의미이므로 2는 a로 나타났고, 4는 b로 나타났으므로 이 둘은 유의미한 차이가 난다는 의미이다. 아마도 전통적으로 많이 사용해온 방식이라 이것이 더 익숙할 수도 있다고 보여진다.  
###3.4.4. oneway ANOVA 수기계산

이번에는 Wiki에 나온 데이터를 가지고 일원배치 분산분석(oneway ANOVA)에 대한 수기계산 공식을 확인하려고 한다.

df = data.frame(a1 = c(6, 8, 4, 5, 3, 4),  
 a2 = c(8, 12, 9, 11, 6, 8),  
 a3 = c(13, 9 ,11, 8, 7, 12))  
df

## a1 a2 a3  
## 1 6 8 13  
## 2 8 12 9  
## 3 4 9 11  
## 4 5 11 8  
## 5 3 6 7  
## 6 4 8 12

주어진 데이터를 롱포맷으로 변형하자.

df\_long <- df %>% pivot\_longer(cols = c(a1,a2,a3)) %>% arrange(name)  
df\_long

## # A tibble: 18 × 2  
## name value  
## <chr> <dbl>  
## 1 a1 6  
## 2 a1 8  
## 3 a1 4  
## 4 a1 5  
## 5 a1 3  
## 6 a1 4  
## 7 a2 8  
## 8 a2 12  
## 9 a2 9  
## 10 a2 11  
## 11 a2 6  
## 12 a2 8  
## 13 a3 13  
## 14 a3 9  
## 15 a3 11  
## 16 a3 8  
## 17 a3 7  
## 18 a3 12

데이터에 추가하는 방식으로 계산하면 다음과 같이 할 수 있다.

anovaData = df\_long %>% mutate(  
 grp\_mean = ave(df\_long$value, df\_long$name), #그룹별 평균  
 grand\_mean = mean(df\_long$value), #전체 평균   
 n = ave(df\_long$value, df\_long$name, FUN = function(x) length(x)), #그룹별샘플수  
 Sb = sum(unique(n\*(grp\_mean - grand\_mean)^2)), #between sum of square  
 dofb = length(unique(df\_long$name)) -1, #degree of freedom between   
 MSb = Sb/dofb, # mean of square bewteen   
 Sw = sum((value - grp\_mean)^2), #winthin sum of square   
 dofw = length(unique(df\_long$name))\*(n-1), #degree of freedom within  
 MSw = Sw/dofw, #mean of square within   
 F.value = MSb/MSw,   
 p.value = 1-pf(F.value, dofb, dofw)  
 )  
anovaData[,c(1,2,4,6,8,11,12,13)]

## # A tibble: 18 × 8  
## name value grand\_mean Sb MSb MSw F.value p.value  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 a1 6 8 84 42 4.53 9.26 0.00240  
## 2 a1 8 8 84 42 4.53 9.26 0.00240  
## 3 a1 4 8 84 42 4.53 9.26 0.00240  
## 4 a1 5 8 84 42 4.53 9.26 0.00240  
## 5 a1 3 8 84 42 4.53 9.26 0.00240  
## 6 a1 4 8 84 42 4.53 9.26 0.00240  
## 7 a2 8 8 84 42 4.53 9.26 0.00240  
## 8 a2 12 8 84 42 4.53 9.26 0.00240  
## 9 a2 9 8 84 42 4.53 9.26 0.00240  
## 10 a2 11 8 84 42 4.53 9.26 0.00240  
## 11 a2 6 8 84 42 4.53 9.26 0.00240  
## 12 a2 8 8 84 42 4.53 9.26 0.00240  
## 13 a3 13 8 84 42 4.53 9.26 0.00240  
## 14 a3 9 8 84 42 4.53 9.26 0.00240  
## 15 a3 11 8 84 42 4.53 9.26 0.00240  
## 16 a3 8 8 84 42 4.53 9.26 0.00240  
## 17 a3 7 8 84 42 4.53 9.26 0.00240  
## 18 a3 12 8 84 42 4.53 9.26 0.00240

anovaTableData <-anovaData %>%   
 dplyr::select(6:13) %>% unique()  
anovaTableData

## # A tibble: 1 × 8  
## Sb dofb MSb Sw dofw MSw F.value p.value  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 84 2 42 68 15 4.53 9.26 0.00240

분산분석표

rbind.data.frame(  
anovaTableData[1,c(1,2,3,7,8)] %>%   
 `colnames<-`(c("Sum\_sq","Df", "Mean\_sq","F","p")),  
anovaTableData[1,c(4,5,6,7,8)] %>%  
 `colnames<-`(c("Sum\_sq","Df", "Mean\_sq","F","p"))) %>%  
 as.data.frame() %>% `row.names<-`(c("between","within")) %>% round(3)

## Sum\_sq Df Mean\_sq F p  
## between 84 2 42.000 9.265 0.002  
## within 68 15 4.533 9.265 0.002

이것을 표로 나타내면 다음과 같이 나타낼 수 있다.

|  | Sum\_sq | Df | Mean\_sq | F | p |
| --- | --- | --- | --- | --- | --- |
| between | 84 | 2 | 42.000 | 9.265 | 0.002 |
| within | 68 | 15 | 4.533 |  |  |

분산분석 결과, 각 그룹은 통계적으로 유의한 차이가 나타났다, F(2, 15 ) = 9.26, p = 0.002.

각각의 과정을 살펴보자.

각 그룹의 평균

#group mean   
grp\_mean = ave(df\_long$value, df\_long$name) %>%   
 unique()%>%   
 data.frame() %>%   
 `rownames<-`(c("a1\_mean","a2\_mean","a3\_mean")) %>%   
 `colnames<-`(c("mean.value"))  
grp\_mean

## mean.value  
## a1\_mean 5  
## a2\_mean 9  
## a3\_mean 10

전체평균

#ground\_mean  
ground\_mean= mean(df\_long$value)  
ground\_mean

## [1] 8

그룹별샘플수

n=df\_long %>% group\_by(name) %>% summarize(N=n())  
n

## # A tibble: 3 × 2  
## name N  
## <chr> <int>  
## 1 a1 6  
## 2 a2 6  
## 3 a3 6

sum of square between

n[,2]\*(grp\_mean - ground\_mean)^2

## N  
## 1 54  
## 2 6  
## 3 24

sum of square between

# sum of square between  
Sb= sum(n[,2]\*(grp\_mean - ground\_mean)^2)  
Sb

## [1] 84

degree of freedoom between

dofb = length(unique(df\_long$name)) -1 #degree of freedom between   
dofb

## [1] 2

sum of square within

# sum of squae within  
Sw = sum((df\_long$value - ave(df\_long$value, df\_long$name))^2)  
Sw

## [1] 68

degree of freedoom within

dofw = length(unique(df\_long$name))\*((n[1,2]) %>% unlist()-1) #degree of freedom within  
dofw<- dofw %>% as.vector()  
dofw

## [1] 15

mean sum of square

MSb = Sb/dofb %>% as.vector() # mean of square bewteen   
MSw = Sw/dofw %>% as.vector() #mean of square within   
F.value = MSb/MSw   
  
MSb

## [1] 42

MSw

## [1] 4.533333

F.value

## [1] 9.264706

통계적 유의성 검정 여기에서 임계치의 F값을 구해보자. qf()를 이용하여 구하게 된다.

qf(1-.05, dofb, dofw)

## [1] 3.68232

현재 나타난 F값은 9.26이고 임계치는 3.68이므로 F값(9.26) > 임계치 F값(3.68)이므로 통계적으로 유의하다. 다른 방법으로는 p값을 구하여 확인하는 방법이 있다.

p value

#p  
p.value = 1-pf(F.value, dofb, dofw)  
p.value

## [1] 0.002398777

p< 0.001이므로 귀무가설을 기각하고 대립가설을 선택하면 된다.

이렇게 분해해서 값을 구하면 생각보다 일이 좀 많다. 그래서 데이터내에서 계산을 마치는 것이 더 낫다는 것이다. 이렇게 계산하면서 일원배치 분산분석(Oneway ANOVA)에 대한 이해를 높였을 것으로 생각도니다.

#### aov()

aov(value ~ factor(name), df\_long) %>% summary() R의 oneway ANOVA 를 실행하는 함수는 aov()이다. 이를 이용하면 한줄로 간단하게 분산분석을 마무리할 수 있다.

aov(value ~ name, df\_long) %>% summary()

## Df Sum Sq Mean Sq F value Pr(>F)   
## name 2 84 42.00 9.265 0.0024 \*\*  
## Residuals 15 68 4.53   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**APA결과 보고**  
report 패키지를 이용하여 기본적 보고 형식을 이해할 수 있다.

#결과보고   
library(report)  
aov(value ~ factor(name), df\_long) %>% report()

## The ANOVA (formula: value ~ factor(name)) suggests that:  
##   
## - The main effect of factor(name) is statistically significant and large (F(2,  
## 15) = 9.26, p = 0.002; Eta2 = 0.55, 95% CI [0.21, 1.00])  
##   
## Effect sizes were labelled following Field's (2013) recommendations.

## 3.5. Twoway ANOVA

## 반복측정 분산 분석

## MANOVA(다변량 분석)

## ANCOVA 공분산 분석

## 3.5. Chi-square test

독립성 검정과 동질성 검정

## 3.6. Correlation

상관분석은 두 연속변수가 서로 관련이 있는지 검정하는 통계분석이다. 상관분석을 통해서 두 변수가 어느정도의 관계가 있는지 파악할 수 있다. 상관계수의 값은 -1 ~ 1사이의 값을 가ㅣ져 1에 가까울 수록 관련성이 높은 것으로 본다. 값이 양수이면 정비례, 음수이면 반비례 관계를 나타낸다.

### 3.6.1. 상관계수

상관계수는 cor()을 이용하여 상관계수를 찾는다.  
두 개의 변수를 지정하면 두 변수 사이의 관계가 나타난다.

cor(mtcars$mpg, mtcars$hp)

## [1] -0.7761684

변수를 3개로 하면 3개의 변수가 나타난다. R의 베이스에서 변수 선택방벙을 이용하여 나타내었다.

#R base를 이용한 상관계수   
mtcars[, c("mpg","hp","wt")] %>% cor()

## mpg hp wt  
## mpg 1.0000000 -0.7761684 -0.8676594  
## hp -0.7761684 1.0000000 0.6587479  
## wt -0.8676594 0.6587479 1.0000000

tidyverse이 방법을 사용하면 다음과 같다. 익숙해지면 이 방법이 더 편리한 것으로 생각 된다.

#tidyverse를 이용  
mtcars %>% dplyr::select(mpg, hp, wt) %>% cor()

## mpg hp wt  
## mpg 1.0000000 -0.7761684 -0.8676594  
## hp -0.7761684 1.0000000 0.6587479  
## wt -0.8676594 0.6587479 1.0000000

더 많은 변수로 구성된 경우를 상관행렬을 얻어보자.

#다변량으로 구성된 경우   
mtcars[,1:7] %>% cor() %>% round(3)

## mpg cyl disp hp drat wt qsec  
## mpg 1.000 -0.852 -0.848 -0.776 0.681 -0.868 0.419  
## cyl -0.852 1.000 0.902 0.832 -0.700 0.782 -0.591  
## disp -0.848 0.902 1.000 0.791 -0.710 0.888 -0.434  
## hp -0.776 0.832 0.791 1.000 -0.449 0.659 -0.708  
## drat 0.681 -0.700 -0.710 -0.449 1.000 -0.712 0.091  
## wt -0.868 0.782 0.888 0.659 -0.712 1.000 -0.175  
## qsec 0.419 -0.591 -0.434 -0.708 0.091 -0.175 1.000

많은 변수의 상관을 구할 수 있다. cor()함수는 상관계수만을 구해주는 것이며, 유의성 검정을 해주지는 않는다. 따라서, 유의성 검정은 별도로 해야 한다.

### 3.6.2. 상관계수의 유의성 검정

두 변수와의 상관관계는 통계적으로 유의한 것인지 알아보자.  
이때 사용하는 함수는 cor.test()함수이다.  
이 함수의 경우는 2개의 변수만을 입력해야 한다.

cor.test(변수1, 변수2)

이러한 형태로 입력하면 된다.

cor.test(mtcars$mpg, mtcars$hp)

##   
## Pearson's product-moment correlation  
##   
## data: mtcars$mpg and mtcars$hp  
## t = -6.7424, df = 30, p-value = 1.788e-07  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.8852686 -0.5860994  
## sample estimates:  
## cor   
## -0.7761684

mpg(연비)와 마력(hp)의 상관은 -0.78이고, 통게적으로 유의하다(p < .001).

The Pearson’s product-moment correlation between mtcarshp is negative, statistically significant, and very large (r = -0.78, 95% CI [-0.89, -0.59], t(30) = -6.74, p < .001).

이 결과는 report package를 이용하여 보고서를 작성한 것이다.

#report   
cor.test(mtcars$mpg, mtcars$hp) %>%   
 report::report()

## Effect sizes were labelled following Funder's (2019) recommendations.  
##   
## The Pearson's product-moment correlation between mtcars$mpg and mtcars$hp is  
## negative, statistically significant, and very large (r = -0.78, 95% CI [-0.89,  
## -0.59], t(30) = -6.74, p < .001)

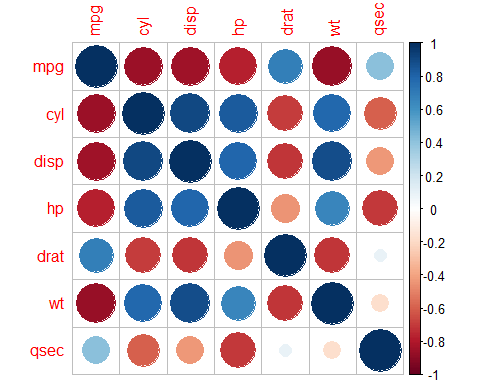
### 3.6.3. 상관계수의 시각화

보통 상관계수를 시각화하면 여러가지 면에서 빠른 파악이 된다. 간단하게 시각화하여 내용을 파악하는 방법과 응용하는 방법을 알아보자.

#### 3.6.3.1 히트맵을 그리는 corrplot

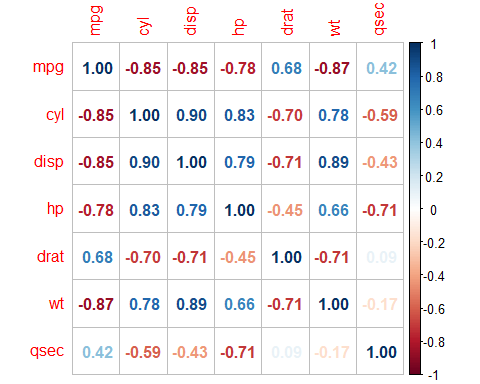
히트맵을 이욯하여 시각화하면 상관의 정도를 파악하는데 유리하다. corrplot패키지를 이용하여 시각화하면 화려한 모습을 나타나게 된다.

#heatmap   
library(corrplot)  
cor(mtcars[,1:7]) %>% corrplot()



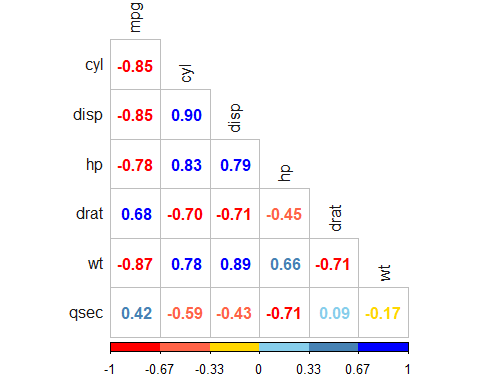
동그라미의 크기로 상관을 파악하고 색으로 판단이 된다. 그러나 정확한 수치(number)를 아는 것이 더 편리할 것이다.  
method = “number”로 하면 숫치로 나타나게 된다.

cor(mtcars[,1:7]) %>%  
 corrplot(method ="number")



이를 좀 더 다듬에서 행렬의 하위 부분에 나타내고, 대각선 부분의 1은 없앤 상태로 나타내 보자. 색은 음의 상관이면 “red”쪽으로 나타나고, 양의 상관은 “blue”쪽으로 나타나도록 설정하였다.

cor(mtcars[,1:7]) %>%   
 corrplot(method ="number", #숫자로 표시   
 type = "lower", #아래 행렬만 표현   
 tl.col = "gray10", #변수명 색 지정  
 diag = FALSE,  
 addCoef.col = "black",  
 col=c("red","tomato","gold","skyblue","steelblue","blue")  
 )



#### 3.6.3.2. rstatix패키지를 이용한 시각화(cor\_plot)

rstatix를 이용하는 것도 실은 corrplot을 이용한 것이나 마찬가지이다. 이를 기반으로 만들어진 것이라서 그런 것인데, rstatix패키지는는 실제 논문에서 적용하기 좋은 형태로 분석의 결과를 제시해주기 때문에 매우 의미있는 패키지이다.

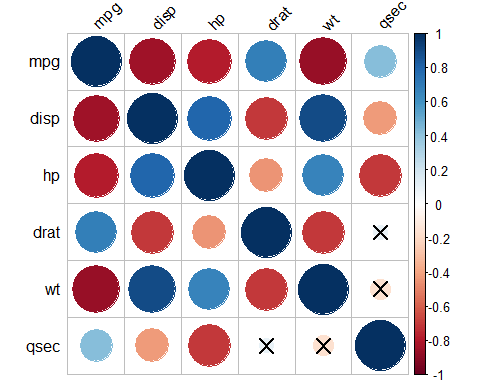
**cor\_mat()**  
먼저 데이터에서 cor\_mat()를 이용하여 상관행렬을 만들어내는 것이 중요하고 이를 이용하여 분석을 한다. 데이터를 저장하기 보다는 최대한 데이터에서 상관행렬을 만드는 것들을 보여주면서 설명하고 있다.

library(rstatix)  
mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat()

## # A tibble: 6 × 7  
## rowname mpg disp hp drat wt qsec  
## \* <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 mpg 1 -0.85 -0.78 0.68 -0.87 0.42   
## 2 disp -0.85 1 0.79 -0.71 0.89 -0.43   
## 3 hp -0.78 0.79 1 -0.45 0.66 -0.71   
## 4 drat 0.68 -0.71 -0.45 1 -0.71 0.091  
## 5 wt -0.87 0.89 0.66 -0.71 1 -0.17   
## 6 qsec 0.42 -0.43 -0.71 0.091 -0.17 1

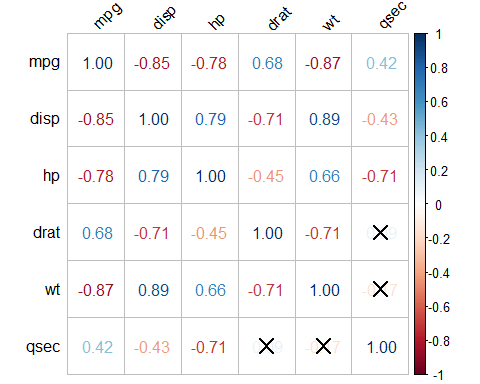
**(1)시각화**  
corrplot과의 차이점은 유의하지 않은 곳은 ’x’로 표시를 해주고 있다.  
사용은 cor\_plot()만 입력하면 되는데, corrplot과도 옵션이 동일하다.

#cor\_plot  
mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat() %>% cor\_plot()



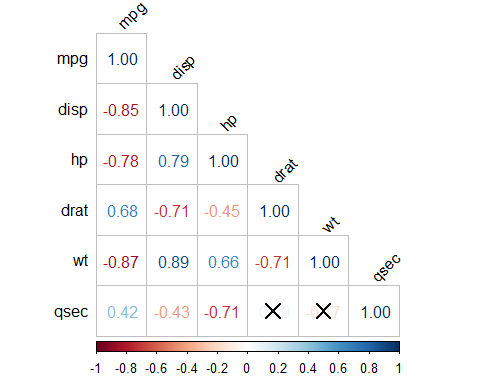
**(2)상관게수를 수치로 나타낸 것** 이번에는 상관계수를 수치로 나타낸 것이다. method = “number”를 설정하면된다. 또한 변수명등 심미적으로 더 보기좋게 나타나고 있다.

mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat()%>% cor\_plot(method = "number")



**(3)lower로 나타내기**  
rstatix::cor\_plot() 먼저 행렬을 lower부분만을 나타내려면 type =“lower”로 표시하면 된다.

mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat()%>%   
 cor\_plot(method = "number", type = "lower")



#diag =FALSE옵션은 수행되지 않음

다른 방법으로는 pull\_lower\_triangel()함수를 이용하여 실제로 행렬을 하위부분만 나타낸 후에 cor\_plot()을 적용하는 방법도 있다.

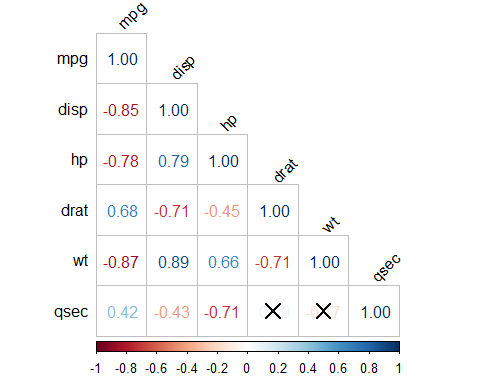
먼저 pull\_lower\_triangel()을 적용하면 다음과 같다. 상관행렬을 보기 좋게 만들어준다.

# pull lower triangle and visualize  
mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat() %>%  
 pull\_lower\_triangle()

## rowname mpg disp hp drat wt qsec  
## 1 mpg   
## 2 disp -0.85   
## 3 hp -0.78 0.79   
## 4 drat 0.68 -0.71 -0.45   
## 5 wt -0.87 0.89 0.66 -0.71   
## 6 qsec 0.42 -0.43 -0.71 0.091 -0.17

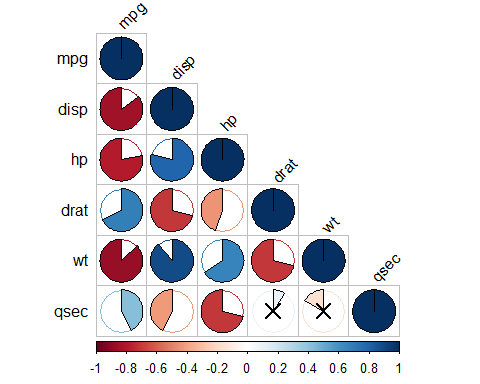
cor+plot()에서 diag=FALSE 옵션은 수행되지 않는다. 따라서, 행렬을 수정하여 상관행렬 히트맵을 그리는 방법이 가능하다. diagonal = TRUE 옵션을 켜서 위에 보여준 히트맵과 동일하게 설정하였다.

mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat() %>%  
 pull\_lower\_triangle(diagonal = T)%>% # lower행렬   
 cor\_plot(method="number") #시각화



**(4) pie를 이욯하여 나타내기**  
상관계수가 아닌 pie를 이용하여 나타낼 수도 있다. method =“pie”로 설정해주면 된다.

mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat() %>%  
 pull\_lower\_triangle(diagonal=T)%>%   
 cor\_plot(method="pie")



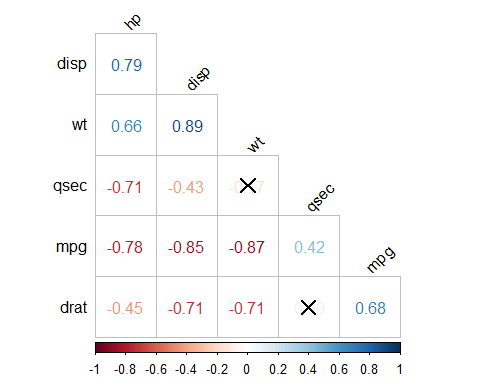
**(5)상관의 크기대로 정렬렬** 만약에 변수를 정렬해서 나타내고자 한다면 cor\_reorder()를 활용한다.  
상관계수가 큰 것부터 나타났다.  
특별한 경우 상관계수를 정렬해서 보여주어야할 때 필요할 것으로 생각이 든다.

mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat() %>%  
 cor\_reorder() %>%  
 pull\_lower\_triangle()

## rowname hp disp wt qsec mpg drat  
## 1 hp   
## 2 disp 0.79   
## 3 wt 0.66 0.89   
## 4 qsec -0.71 -0.43 -0.17   
## 5 mpg -0.78 -0.85 -0.87 0.42   
## 6 drat -0.45 -0.71 -0.71 0.091 0.68

이렇게 상관의 크기대로 정렬하여 그려보자.

mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat() %>%  
 cor\_reorder() %>%  
 pull\_lower\_triangle()%>% cor\_plot(method="number")



### 3.6.4. 상관계수의 데이터 처리와 활용

1. 먼저 상관행렬을 보자.

여기까지는 cor()을 이용하는 것이나 큰 차이가 없다.

mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%   
 rstatix::cor\_mat()

## # A tibble: 6 × 7  
## rowname mpg disp hp drat wt qsec  
## \* <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 mpg 1 -0.85 -0.78 0.68 -0.87 0.42   
## 2 disp -0.85 1 0.79 -0.71 0.89 -0.43   
## 3 hp -0.78 0.79 1 -0.45 0.66 -0.71   
## 4 drat 0.68 -0.71 -0.45 1 -0.71 0.091  
## 5 wt -0.87 0.89 0.66 -0.71 1 -0.17   
## 6 qsec 0.42 -0.43 -0.71 0.091 -0.17 1

1. 상관계수의 유의성 검정

상관계수의 유의성 검정을 위해 p값을 확인하여 검정해보자.  
각 위치에 적절하게 p값을 나타내고 있다.

# p value  
mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%   
 rstatix::cor\_mat() %>%   
 cor\_get\_pval()

## # A tibble: 6 × 7  
## rowname mpg disp hp drat wt qsec  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 mpg 0 9.38e-10 0.000000179 0.0000178 1.29e- 10 0.0171   
## 2 disp 9.38e-10 0 0.0000000714 0.00000528 1.22e- 11 0.0131   
## 3 hp 1.79e- 7 7.14e- 8 0 0.00999 4.15e- 5 0.00000577  
## 4 drat 1.78e- 5 5.28e- 6 0.00999 0 4.78e- 6 0.62   
## 5 wt 1.29e-10 1.22e-11 0.0000415 0.00000478 2.27e-236 0.339   
## 6 qsec 1.71e- 2 1.31e- 2 0.00000577 0.62 3.39e- 1 0

1. 롱포맷데이터로 변경

이번에는 이렇게 만들어진 상관행렬의 값을 롱포맷으로 만등어서 변수간 상관계수와 유의성을 나타내는 방법도 있다. 이경우는 cor\_gather()함수를 사용한다.  
일단은 모든 변수를 나타내도록 설정하여 만들었다.

# Gather/collapse correlation matrix into long format  
mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%   
 rstatix::cor\_mat() %>%   
 cor\_gather() %>% print(n=36)

## # A tibble: 36 × 4  
## var1 var2 cor p  
## <chr> <chr> <dbl> <dbl>  
## 1 mpg mpg 1 0   
## 2 disp mpg -0.85 9.38e- 10  
## 3 hp mpg -0.78 1.79e- 7  
## 4 drat mpg 0.68 1.78e- 5  
## 5 wt mpg -0.87 1.29e- 10  
## 6 qsec mpg 0.42 1.71e- 2  
## 7 mpg disp -0.85 9.38e- 10  
## 8 disp disp 1 0   
## 9 hp disp 0.79 7.14e- 8  
## 10 drat disp -0.71 5.28e- 6  
## 11 wt disp 0.89 1.22e- 11  
## 12 qsec disp -0.43 1.31e- 2  
## 13 mpg hp -0.78 1.79e- 7  
## 14 disp hp 0.79 7.14e- 8  
## 15 hp hp 1 0   
## 16 drat hp -0.45 9.99e- 3  
## 17 wt hp 0.66 4.15e- 5  
## 18 qsec hp -0.71 5.77e- 6  
## 19 mpg drat 0.68 1.78e- 5  
## 20 disp drat -0.71 5.28e- 6  
## 21 hp drat -0.45 9.99e- 3  
## 22 drat drat 1 0   
## 23 wt drat -0.71 4.78e- 6  
## 24 qsec drat 0.091 6.2 e- 1  
## 25 mpg wt -0.87 1.29e- 10  
## 26 disp wt 0.89 1.22e- 11  
## 27 hp wt 0.66 4.15e- 5  
## 28 drat wt -0.71 4.78e- 6  
## 29 wt wt 1 2.27e-236  
## 30 qsec wt -0.17 3.39e- 1  
## 31 mpg qsec 0.42 1.71e- 2  
## 32 disp qsec -0.43 1.31e- 2  
## 33 hp qsec -0.71 5.77e- 6  
## 34 drat qsec 0.091 6.2 e- 1  
## 35 wt qsec -0.17 3.39e- 1  
## 36 qsec qsec 1 0

여기에서 자세히보면 qsec 와 mpg의 상관계수가 0.42인데, 뒷부분에 보면 mpg와 qsec의 상관계수가 0.42로 중복해서 나타나고 있다. 이런 경우는 행렬에서 upper부분을 없애고 다시 cor\_gather()을 하면 된다.  
이때 upper부분을 없애버리거나 lower부분만을 남겨서 해보자. 방법은 두가지가 있는데, pull\_lower\_triangle()를 이용하여 lower부분만을 남기는 것이다.

mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat()%>% pull\_lower\_triangle()

## rowname mpg disp hp drat wt qsec  
## 1 mpg   
## 2 disp -0.85   
## 3 hp -0.78 0.79   
## 4 drat 0.68 -0.71 -0.45   
## 5 wt -0.87 0.89 0.66 -0.71   
## 6 qsec 0.42 -0.43 -0.71 0.091 -0.17

여기에 cor\_gather()적용하여 보자. 이렇게 하면 6개의 변수에 대하여 6C2 = 15개의 pairwise list가 나타나게된다.

mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat()%>% pull\_lower\_triangle()%>%   
 cor\_gather() %>%  
 tibble()

## # A tibble: 15 × 4  
## var1 var2 cor p  
## <chr> <chr> <dbl> <dbl>  
## 1 disp mpg -0.85 9.38e-10  
## 2 hp mpg -0.78 1.79e- 7  
## 3 drat mpg 0.68 1.78e- 5  
## 4 wt mpg -0.87 1.29e-10  
## 5 qsec mpg 0.42 1.71e- 2  
## 6 hp disp 0.79 7.14e- 8  
## 7 drat disp -0.71 5.28e- 6  
## 8 wt disp 0.89 1.22e-11  
## 9 qsec disp -0.43 1.31e- 2  
## 10 drat hp -0.45 9.99e- 3  
## 11 wt hp 0.66 4.15e- 5  
## 12 qsec hp -0.71 5.77e- 6  
## 13 wt drat -0.71 4.78e- 6  
## 14 qsec drat 0.091 6.2 e- 1  
## 15 qsec wt -0.17 3.39e- 1

1. 다시 행렬로 만들기

이것을 다시 원래대로 돌려놓으려면 cor\_spread(value=“cor”)를 활용하면 된다. NA(결측)이 나타나게되면 replace\_upper\_triangle(by = ““)를 이용하여 NA를 공백으로 바꾸면 된다.

# spredd  
mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat()%>% pull\_lower\_triangle()%>%   
 cor\_gather() %>%  
 cor\_spread(value="cor") %>%   
 replace\_upper\_triangle(by = "") #NA를 제거

## rowname mpg disp hp drat wt  
## 1 disp   
## 2 hp -0.78   
## 3 drat 0.68 -0.71   
## 4 wt -0.87 0.89 0.66   
## 5 qsec 0.42 -0.43 -0.71 0.091

1. 상관행렬에 유의성 표시

논문에 넣을 때 보통 p값을 근거로 하여 ’\*’로 유의성을 표시하는 경우가 많다. 이런 경우에는 cor\_mark\_significant()를 사용하면 이 부분을 좀 더 편리하게 사용할 수 있다.

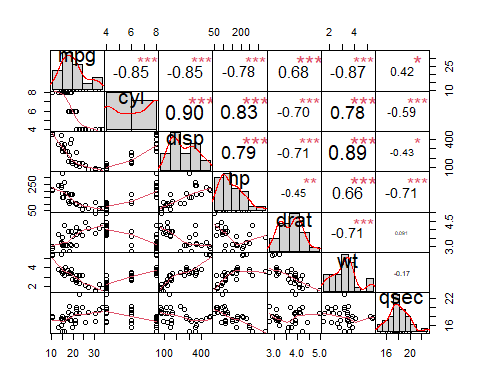
# Add Significance Levels To a Correlation Matrix  
mtcars %>%  
 dplyr::select(mpg, disp, hp, drat, wt, qsec) %>%  
 rstatix::cor\_mat() %>%  
 cor\_mark\_significant()

## rowname mpg disp hp drat wt qsec  
## 1 mpg   
## 2 disp -0.85\*\*\*\*   
## 3 hp -0.78\*\*\*\* 0.79\*\*\*\*   
## 4 drat 0.68\*\*\*\* -0.71\*\*\*\* -0.45\*\*   
## 5 wt -0.87\*\*\*\* 0.89\*\*\*\* 0.66\*\*\*\* -0.71\*\*\*\*   
## 6 qsec 0.42\* -0.43\* -0.71\*\*\*\* 0.091 -0.17

### 3.6.5. 응용패키지를 이용한 상관분석(데이터, 시각화 )

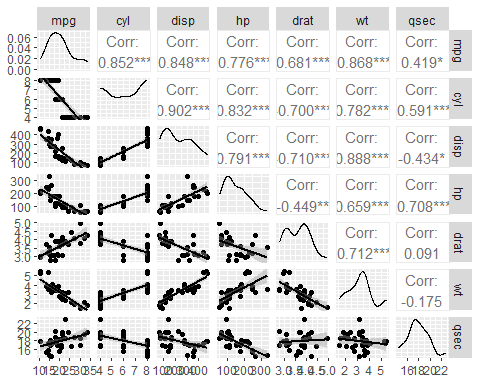
**PerformanceAnalytics::chart.Correlation()**  
이 패키지는 옵션이 간단하고, 사용법도 편리하다. 하단에는 산점도와 회귀직성, 대각선에는 히스토그램과 밀도그래프, 상단에는 상관계와 유의성을 표시해준다.  
시각적으로도 보기 좋아 활요도가 높은 패키지이다.

library(PerformanceAnalytics)  
chart.Correlation(mtcars[,1:7])



**GGally:ggpairs()** 이 패키지도 논문등에 자주 사용되는 패키지이다. 옵션설정도 다양하고, 기본적인 것만을 사용해도 보기좋은 그래프와 상관분석 결과가 얻어진다.

library(GGally)  
ggpairs(mtcars[,1:7],lower = list(continuous = "smooth"))



이러한 심미적인 그림을 제공하는 패키지는 논문을 집필할 때 활용도가 매우높다. 단점이라고 한다면 코드를 기억하기 쉽지 않다는 점이다. 따라서, 본인이 찾기좋은 곳에 메모해두거나 혹은 확인하기 좋은 곳에 적어두면 도움이 될 것이다.

## 3.7. Regression

### OLS 회귀분석

### 로지스틱 회귀분석

# 4. 응용통계분석(1)

## 4.1. 경로 분석

## 4.2. 매개효과 분석

## 4.3. 조절효과 분석

## 4.4. Process Macro

# 5. 구조방정식 모형

# 6. PLS 구조 방정식 모형

# 7. 다변량 분석

## 주성분 분석

## 인자분석

## 7.1. 다차원 척도법

## 7.2. 대응일치분석

## 7.3. 군집 분석

## 7.4. 잠재프로파일 분석

## 형상 분석

## 소셜네트워크 분석

## 연관분석

# Reference

* Creswell, J. W. (2014). Research design: Qualitative, quantitative, and mixed methods approaches (4th ed.). Thousand Oaks, CA: Sage publications.
* Hair, J. F., Jr., Hult, G. T. M., Ringle, C. M., & Sarstedt, M. (2017). A primer on partial least squares structural equation modeling (PLS-SEM). Thousand Oaks, CA: Sage publications.
* Levene, H. (1960). Robust tests for equality of variances. Contributions to probability and statistics, 278-292.
* Fligner, M. A., & Killeen, T. J. (1976). Distribution of the F statistic for the Kruskal-Wallis one-way analysis of variance. Journal of the American Statistical Association, 71(353), 207-210.