임상의학 연구자들을 위한 필수 R

## 

## CHAPTER 1. R 시작하기

## 1.8 R 나도 할 수 있다.

library(tidyverse)  
library(survival)  
library(survminer)  
library(ggsci)  
library(ggsignif)  
library(gtsummary)  
library(forestmodel)

### 1.8.1 Table 1 쉽게 만들기

dt1<-read\_csv("Example\_data/Ch1\_table1.csv")   
  
dt1 %>%  
 select(-id, -hcc\_yr, -m6\_alb) %>%  
 tbl\_summary(by=LC,missing='no') %>%   
 add\_p() %>%   
 add\_overall() %>%   
 modify\_spanning\_header( c('stat\_1','stat\_2')~'\*\*Liver Function\*\*')

### 1.8.2 Multivariable analysis table

dt2<-read\_csv("Example\_data/Ch1\_multi.csv")  
  
fit.multi <- glm(Group~CurrentUser+Age+RaceGroup+Gender+  
 HBV+HCV+Cirrhosis+IBD+Diabetes+Obesity+  
 NAFLD+Smoking,  
 family=binomial, data=dt2)  
  
tbl\_regression(fit.multi, exponentiate = T) %>%   
 bold\_labels() %>%   
 bold\_p()

### 1.8.3 Forest plot

forest\_model(fit.multi)

### 1.8.4 NEJM bar 그래프

ggplot(dt1,aes(x= Sex, y=Platelet, fill=factor(HCC)))+  
 geom\_bar(stat='identity', position='dodge')+  
 theme\_bw()+  
 scale\_fill\_nejm()+  
 geom\_signif(comparisons = list(c('F','M')))

### 1.8.5 JCO KM 그래프

dt1<-read\_csv("Example\_data/Ch1\_table1.csv")   
  
km1<-survfit(Surv(hcc\_yr, HCC)~LC, data=dt1)  
  
ggsurvplot(km1,palette = 'jco',  
 risk.table = T)

### 1.8.6 Waterfall 그래프

dt1 %>%   
 filter(id<=30) %>%   
 select(id,Sex,Albumin,m6\_alb) %>%   
 mutate(delta\_alb=Albumin-m6\_alb) %>%   
 ggbarplot(x='id', y='delta\_alb',  
 fill='Sex',  
 sort.val='desc',  
 sort.by.groups = FALSE)

### CHAPTER 2. 데이터 분석의 시작

### 

### 2.2.2 파일 불러오기

library(tidyverse)

dat <- read.csv("Example\_data/Ch2\_chb.csv")  
class(dat)

dat <- read\_csv("Example\_data/Ch2\_chb.csv")   
class(dat)

## 

## 2.3 데이터 훑어보기

View(dat)

dim(dat)

colnames(dat)

rownames(dat)

str(dat)

### 

### 2.3.2 데이터 요약 보기

head(dat)

head(dat, 10)

tail(dat)

tail(dat,8)

glimpse(dat)

## 

## 2.4 데이터 인덱싱, 슬라이싱

dat[ ,age]

dat[ , 'age']

dat[, 4]

dat$age

a1 <- dat[ ,'age']   
a1

class(a1)

b1 <- dat$age  
class(b1)

is.data.frame(b1)

is.vector(b1)

b1

a2 <-dat[ ,'age', drop=TRUE]  
a2

class(a2)

### 

### 2.4.2 Column 여러개 선택

dat[ ,c('age','treat\_gr','lc','hcc')]

var <- c('age','treat\_gr','lc','hcc')  
dat[ ,var]

dat[ , c(4,6,7,8)]

temp <- dat[ , c('id','index\_date','gender','age','last\_date','treat\_gr','lc','hcc')]  
head(temp)

temp <- dat[ ,1:8]  
head(temp)

temp <- dat[ ,c(1:10, 12:20)]  
head(temp)

temp <- dat[ , -11]  
head(temp)

### 

### 2.4.3 Row 한개 선택

pt1 <- dat[1, ]   
pt1

class(pt1)

### 

### 2.4.4 Row 여러개 선택

pt2 <- dat[1:4, ]  
pt2

class(pt2)

pt <- dat[c(1,4,9), ]  
pt

pt <- dat[c(1:4,10:14), ]  
pt

### 

### 2.4.5 Row, column 동시 선택

temp <- dat[c(1:6), c(1:8)]  
temp

dim(temp)

temp <- dat[c(1:6,11:16), c(1:8)]  
head(temp, 12)

temp <- dat[c(1:6), c(1:6, 12:18)]  
head(temp)

temp <- dat[c(1:6, 16:21), c(1:6, 12:18)]  
head(temp,10)

### 

### 2.4.6 특정 조건으로 선택하기

dat[12, 'alt']

dat[dat$age>=50, ]

dat$age[dat$age>=50]

dat[dat$age>=50 & dat$gender=="M", ]

### 

### 2.4.7 Subset

subset(dat, subset=(age>=50 & gender=='M'))

subset(dat, select=c('lc','alt','bil','inr','alb'))

subset(dat,   
 subset=(age>=50 & lc==1 & gender=='M'),  
 select=c('id','alb','bil','cr'))

high.risk<-subset(dat,   
 subset=(age>=50 & lc==1 & gender=='M'),  
 select=c('id','alb','bil','cr'))  
high.risk

## 2.5 Factor 다루기

### 2.5.1 Factor로 변환하기

sapply(dat, class)

dat1<-dat

class(dat1$gender)

table(dat1$gender)

dat1$sex<-factor(dat1$gender)  
class(dat1$sex)

table(dat1$sex)

dat1$sex1<-as.factor(dat1$gender)  
class(dat1$sex1)

table(dat1$sex1)

dat1$sex2<-as\_factor(dat1$gender)  
class(dat1$sex2)

table(dat1$sex2)

### 

### 2.5.2 Factor level 이해하기

dat1$risk\_gr1<-as.factor(dat1$risk\_gr)  
levels(dat1$risk\_gr1)

dat1$risk\_gr2<-factor(dat1$risk\_gr,  
 levels=c('low','intermediate','high'))  
levels(dat1$risk\_gr2)

### 

### 2.5.3 forcat 패키지

dat1$region<-factor(dat1$region)  
table(dat1$region)

dat1$region1<-fct\_infreq(dat1$region)  
levels(dat1$region1)

dat1$region2 <-fct\_recode(dat1$region,  
 'north' = 'seoul',  
 'north' = 'incheon',  
 'south' = 'gwangju',  
 'south' = 'jeju',  
 'south' = 'sejong',  
 'south' = 'daejun',  
 'east' = 'daegu',  
 'east' = 'busan')

dat1 %>%   
 count(region, region2)

table(dat1$region2)

dat1$region3 <- fct\_collapse(dat1$region,  
 'north' = c('seoul','incheon'),  
 'south' = c('gwangju','jeju','sejong','daejun'),  
 'east' = c('daegu','busan'))

dat1 %>%   
 count(region, region2, region3)

dat1$region4 <- fct\_lump(dat1$region, n=4)

dat1 %>%   
 count(region,region4)

dat1$region4 <- fct\_lump(dat1$region, n=2)

dat1 %>%   
 count(region,region4)

## 

## 2.6 기술 통계

mean(dat$age)

median(dat$age)

sd(dat$age)

var(dat$age)

min(dat$age)

max(dat$age)

range(dat$age)

IQR(dat$age)

quantile(dat$age)

quantile(dat$age, prob=c(0.3,0.6))

quantile(dat$age, prob=c(1:10/10))

prob <- c(1:10/10)

summary(dat$age)

Hmisc::describe(dat$age)

str(dat)

sapply(dat, class)

class(dat$age)

is.character(dat$age)

is.character(dat$gender)

is.numeric(dat$age)

is.numeric(dat$gender)

is.integer(dat$age)

## 

## 2.7 데이터 수정 및 결측치

class(dat$dna)

dat$dna

dat[dat$dna=='undetectable',]

dat[19, 'dna'] <- 0

dat[dat$dna=='undetectable',]

dat[19, 'dna'] <- '0'

dat$dna

class(dat$dna)

dat$dna <- as.numeric(dat$dna)

class(dat$dna)

### 

### 2.7.1 결측값 확인하기

mean(dat$alb)

mean(dat$alb, na.rm=T)

is.na(dat$alb)

sum(is.na(dat$alb))

### 

### 2.7.2 결측값 한번에 확인하기

rowSums(is.na(dat))

barplot(rowSums(is.na(dat)))

na.count <- apply(dat, 2, function(x) sum(is.na(x)))

barplot(na.count[na.count>0])

### 

### 2.7.3 VIM 패키지 이용하기

library(VIM)

missing <- aggr(dat, col=c('navyblue','yellow'),  
 numbers=TRUE, sortVars=TRUE,  
 labels=names(dat1), cex.axis=.7,  
 gap=3,  
 ylab=c('Missing data','Pattern'))

### 

### 2.8.1 tapply

mean(dat$age[dat$gender=='M'])

mean(dat$age[dat$gender=='F'])

tapply(dat$age, dat$gender, mean)

dat %>%   
 group\_by(gender) %>%   
 summarise(mean(age))

### 

### 2.8.2 sapply

sapply(dat, class)

### 

### 2.8.3 lapply

lapply(dat[,c('age','gender')],class)

## 2.9 if, for

### 2.9.1 If

age<-60  
  
if(age>=50){  
print('old age')  
}else{  
print('young age')  
}

if(age>=50){  
 print('old age')  
 }else{  
 print('young age')  
}

age<-45  
if(age>=50){  
 print('old age')  
 }else if(age<30){  
 print('young age')  
 }else{  
 print('middle age')  
}

age<-c(40, 50, 60)  
ifelse(age<40, 'young', 'old')

### 

### 2.9.2 for

for(i in 1:10){  
 print(i)  
}

for(i in 1:10){  
 if(i %% 2 ==0){  
 print(i)  
 }  
}

## 

## 2.10 중복 결과값 다루기

### 2.10.1 Unique

dat$cr

length(dat$cr)

unique(dat$cr)

length(unique(dat$cr))

unique(dat$gender, dat$cr)

unique(dat[,c('gender','cr')])

dat %>%   
 distinct(gender, cr)

### 

### 2.10.2 Duplicated

dat$gender

duplicated(dat$gender)

temp<-rbind(dat, dat[c(3,6,10),])  
dim(temp)

tail(temp)

temp$id

duplicated(temp$id)

!duplicated(temp$id)

temp1<-temp[!duplicated(temp$id),]

unique(temp1$id)

## 

## 2.11 Table 1 만들기

### 2.11.1 moonBook 패키지

library(moonBook)  
mytable(treat\_gr~age+gender+lc+alt+bil+  
 inr+cr+plt+alb+eag+dna\_log,  
 data=dat)

mytable(treat\_gr~age+gender+lc+alt+bil+  
 inr+cr+plt+alb+eag+dna\_log,   
 data=dat, method=2)

mytable(treat\_gr~age+gender+lc+alt+bil+  
 inr+cr+plt+alb+eag+dna\_log,   
 data=dat, show.total=T)

mytable(~age+gender+lc+alt+bil+  
 inr+cr+plt+alb+eag+dna\_log,   
 data=dat)

table1 <- mytable(treat\_gr~age+gender+lc+alt+bil+  
 inr+cr+plt+alb+eag+dna\_log,   
 data=dat, show.total=T)

mycsv(table1, file='table1.csv')

### 

### 2.11.2 tableone 패키지

library(tableone)  
listVars <- names(dat[, c('age','gender','lc','alt','bil',  
 'inr','cr','plt','alb','eag','dna\_log')])   
catVars <- c('gender','lc')

table1 <- CreateTableOne(vars = listVars,  
 factorVars = catVars,  
 strata = c('treat\_gr'),  
 data = dat)  
table1

summary(table1)

print(table1)

write.csv(print(table1), file='tableone.csv')

### 

### 2.11.3 gtsummary 패키지

library(gtsummary)  
  
dat.temp<-dat %>%   
 select(treat\_gr,gender, age, lc, bil, inr, cr, dna\_log)   
  
colnames(dat.temp)

tbl\_summary(dat.temp)

tbl\_summary(dat.temp, by=treat\_gr)

tbl\_summary(dat.temp, by=treat\_gr, missing='no')

tbl\_summary(dat.temp, by='treat\_gr', missing\_text='missing value')

tbl\_summary(dat.temp, by=treat\_gr,  
 statistic=all\_continuous()~ '{mean} \u00b1 {sd}',  
 missing='no')

tbl\_summary(dat.temp, by=treat\_gr,  
 missing='no') %>%  
 add\_p()

tbl\_summary(dat.temp, by=treat\_gr,  
 missing='no') %>%  
 add\_n()

tbl\_summary(dat.temp, by=treat\_gr,  
 missing='no') %>%  
 add\_n()

dat.temp %>%   
 tbl\_summary(by=treat\_gr,  
 missing='no') %>%   
 modify\_caption('\*\*Baseline Characteristics\*\*')

dat.temp %>%   
 tbl\_summary(by=treat\_gr,  
 missing='no') %>%   
 modify\_spanning\_header( c('stat\_1','stat\_2')~'\*\*Antiviral Treatment\*\*')

tbl\_summary(dat.temp,  
 by=treat\_gr,  
 missing='no') %>%   
 add\_p() %>%   
 add\_overall() %>%   
 modify\_spanning\_header( c('stat\_1','stat\_2')~'\*\*Antiviral Treatment\*\*') %>%   
 modify\_caption('\*\*Baseline Characteristics\*\*')

# Chapter 3. 데이터 핸들링

library(tidyverse)

dat <- read\_csv("Example\_data/Ch3\_chb.csv")

dat1 <- dat

class(dat1)

dim(dat1)

colnames(dat1)

iris %>%   
 group\_by(Species) %>%  
 summarize\_if(is.numeric, mean) %>%  
 ungroup() %>%  
 gather(measure, value, -Species) %>%  
 arrange(value)

iris %>% group\_by(Species) %>% summarize\_all(mean) %>%   
ungroup %>% gather(measure, value, -Species) %>%  
arrange(value)

## 

## 3.2 Select

dat1 %>%   
 select(id, b\_alt, b\_bil, b\_alb) %>%   
 print(n=4)

dat1[ , c('id','b\_alt','b\_bil','b\_alb')]

dat1 %>%   
 select(1, 8, 9, 13) %>%   
 print(n=4)

dat1 %>%   
 select(1:4, 7:9, 13) %>%   
 print(n=4)

dat1 %>%   
 select(m6\_alt:m6\_plt) %>%   
 print(n=4)

dat1 %>%   
 select(-id, -index\_date) %>%   
 colnames()

colnames(dat1)

dat1 %>%   
 select(-1, -2) %>%   
 colnames()

dat1 %>%   
 select(-(1:4), -(10:13)) %>%   
 ncol()

dat1 %>%   
 select(b\_alt, m6\_alt, m12\_alt) %>%   
 print(n=4)

dat1 %>%   
 select(contains('alt')) %>%   
 print(n=4)

dat1 %>%   
 select(contains('m6')) %>%   
 print(n=4)

dat1 %>%   
 select(starts\_with('m6')) %>%   
 print(n=4)

dat1 %>%   
 select(ends\_with('cr')) %>%   
 print(n=4)

dat1 %>%   
 select(sex = gender) %>%   
 print(n=4)

dat1 %>%   
 rename(sex = gender) %>%   
 print(n=4)

dat1 %>%   
 rename(emr\_id = id, baseline\_date = index\_date, sex = gender) %>%   
 colnames()

dat1 %>%   
 select(treat\_gr, hcc, id, everything()) %>%   
 colnames()

dat1 %>%   
 select(id, gender, contains('m12')) %>%  
 rename(sex = gender) %>%  
 select(sex, everything()) %>%  
 print(n=4)

dat1 %>%   
 select(sex = gender, id, contains('m12')) %>%  
 print(n=4)

dat1 %>%   
 select(sex = gender,  
 id,  
 contains('m12')) %>%   
 print(n=4)

dat1 %>%   
 select(-4:7) %>%   
 colnames()

dat1 %>%   
 select(-c(4:7)) %>%   
 colnames()

## 

## 3.3 Filter

dat1 %>%   
 filter(id<=20) %>%   
 print(n=4)

dat1 %>%   
 filter(id<=20) %>%   
 select(id, starts\_with('b\_')) %>%   
 print(n=4)

dat1 %>%   
 filter(gender =='M') %>%   
 print(n=4)

dat1 %>%   
 filter(age>=50, gender=='M') %>%   
 print(n=4)

dat1 %>%   
 filter(age >=30 & age<=60) %>%   
 print(n=4)

dat1 %>%   
 filter(between (age, 30, 60)) %>%   
 print(n=4)

dat1 %>%   
 filter(age >=50 | lc==1) %>%   
 print(n=4)

dat1 %>%   
 filter( (age>=60 & gender=="M") | (age<=50 & gender=="F")) %>%   
 print(n=4)

dat1 %>%   
 filter(gender!="M") %>%  
 print(n=4)

dat1 %>%   
 filter(lc !=1 & age >=50) %>%   
 print(n=4)

dat1 %>%   
 count(gender)

dat1 %>%   
 filter(gender == 'M' & hcc ==1) %>%   
 count()

dat1 %>%   
 filter(gender == 'M' & lc ==1) %>%   
 count(hcc)

dat1 %>%   
 filter(gender == 'M' & lc ==1) %>%   
 count(treat\_gr, hcc)

dat1 %>%   
 filter(is.na(b\_inr)) %>%   
 count()

dat1 %>%   
 filter(!is.na(b\_inr)) %>%   
 print(n=4)

dat1 %>%   
 filter(!is.na(b\_inr),  
 !is.na(b\_alt),  
 !is.na(b\_plt)) %>%   
 count()

dat1 %>%   
 drop\_na(b\_inr, b\_alt, b\_plt) %>%   
 count()

dat1 %>%   
 drop\_na() %>%   
 count()

na.count <- apply(dat1, 2, function(x)sum(is.na(x)))  
na.count

## 

## 3.4 Mutate

dat1 %>%   
 mutate(alt\_plt = b\_alt / b\_plt) %>%   
 select(b\_alt, b\_plt, alt\_plt) %>%   
 print(n=4)

dat1 %>%   
 mutate(alt\_plt = b\_alt /b\_plt) %>%   
 select(id, b\_alt, b\_plt, alt\_plt) %>%   
 filter(is.na(b\_alt) | is.na(b\_plt))

dat1 %>%   
 drop\_na(b\_alt, b\_plt) %>%   
 mutate(alt\_plt = b\_alt /b\_plt) %>%   
 select(id, b\_alt, b\_plt, alt\_plt) %>%   
 print(n=4)

dat1 %>%   
 mutate(age\_gr=ifelse(age >=50,'above\_50','below\_50')) %>%  
 select(id, age, age\_gr) %>%   
 print(n=6)

dat1 %>%   
 mutate(age\_gr=ifelse(age >=50,'above\_50','below\_50')) %>%  
 count(age\_gr)

dat1 %>%   
 mutate(bil\_gr=ifelse(b\_bil<2,'A',ifelse(b\_bil<3,'B','C'))) %>%   
 select(id, b\_bil, bil\_gr) %>%   
 print(n=6)

dat1 %>%   
 mutate(bil\_gr=ifelse(b\_bil<2,'A',  
 ifelse(b\_bil<3,'B','C'))) %>%   
 count(bil\_gr)

dat1 %>%   
 mutate(risk\_gr=ifelse(age>=50 & lc==1,'high\_risk',  
 ifelse(age<50 & lc==0, 'low\_risk', 'intermediate\_risk'))) %>%   
 count(risk\_gr)

dat1 %>%   
 transmute(risk\_gr=ifelse(age>=50 & lc==1,'high\_risk',  
 ifelse(age<50 & lc==0, 'low\_risk', 'intermediate\_risk'))) %>%   
 print(n=4)

dat1 %>%   
 mutate(age\_rank = min\_rank(age)) %>%   
 select(id, age, age\_rank) %>%   
 print(n=6)

dat1 %>%   
 mutate(age\_rank = min\_rank(desc(age))) %>%   
 select(id, age, age\_rank) %>%   
 print(n=6)

dat1 %>%   
 mutate(age\_rank = dense\_rank(age)) %>%   
 select(id, age, age\_rank) %>%   
 print(n=6)

dat1 %>%   
 mutate(age\_rank = percent\_rank(age)) %>%   
 select(id, age, age\_rank) %>%   
 arrange(age\_rank) %>%   
 print(n=6)

dat1 %>%   
 mutate(id\_sum = cumsum(id)) %>%   
 select(id, id\_sum) %>%   
 print(n=4)

dat1 %>%   
 mutate(age\_gr=ifelse(age>=50,'above 50','below 50')) %>%   
 mutate(bil\_gr=ifelse(b\_bil<2,'A',  
 ifelse(b\_bil>3,'C','B'))) %>%   
 select(age, age\_gr, b\_bil, bil\_gr) %>%   
 print(n=6)

dat1 %>%   
 mutate(age\_gr=ifelse(age>=60,'>=60',  
 ifelse(age>=50,'>=50',  
 ifelse(age>=40,'>=40',  
 ifelse(age>=30,'>=30','<30'))))) %>%   
 select(age, age\_gr) %>%   
 print(n=6)

dat1 %>%   
 mutate(age\_gr=cut(age,  
 c(-Inf,30,40,50,60,Inf),  
 c('<30','>=30','>=40','>=50','>=60'))) %>%

select(age, age\_gr) %>%   
 print(n=6)

dat1 %>%   
 mutate(age\_gr=cut\_width(age, width=10)) %>%   
 select(age, age\_gr) %>%   
 print(n=6)

dat1 %>%   
 mutate(age\_gr=cut\_width(age, width=10, boundary=0)) %>%  
 mutate(age\_gr2=cut\_width(age, width=10, boundary=9)) %>%   
 select(age, age\_gr, age\_gr2) %>%   
 print(n=6)

dat1 %>%   
 mutate(age\_gr=cut\_interval(age, n=4)) %>%   
 select(age, age\_gr) %>%   
 count(age\_gr)

dat1 %>%   
 mutate(age\_gr = cut\_number(age, n=5)) %>%   
 select(age, age\_gr) %>%   
 print(n=5)

dat1 %>%   
 mutate(age\_gr = cut\_number(age, n=5)) %>%   
 select(age, age\_gr) %>%   
 count(age\_gr)

## 

## 3.5 Arrange

dat1 %>%   
 arrange(age) %>%   
 print(n=4)

dat1 %>%   
 arrange(desc(age)) %>%   
 print(n=4)

dat1 %>%   
 arrange(age, b\_alt) %>%   
 select(age, b\_alt) %>%   
 print(n=6)

## 

## 3.6. Summarise

mean(dat1$age)

dat1 %>%   
 summarise(mean\_age = mean(age))

dat1 %>%   
 summarise(mean\_age = mean(age),  
 median\_age = median(age),  
 iqr\_age = IQR(age))

dat1 %>%   
 summarise(n\_distinct(age))

unique(dat1$age)

length(unique(dat1$age))

dat1 %>%   
 summarise( patient\_number = n(),  
 event = sum(hcc),  
 person\_year = sum(hcc\_yr),  
 incidence\_rate = event / person\_year)

## 

## 3.7 Group\_by

dat1 %>%   
 group\_by(gender) %>%   
 summarise(mean\_age = mean(age))

dat1 %>%   
 group\_by(gender, lc) %>%   
 summarise(patient\_no = n(),  
 mean\_age = mean(age))

dat1 %>%   
 group\_by(gender, lc) %>%   
 summarise(patient\_no = n(),  
 mean\_inr = mean(b\_inr))

dat1 %>%   
 group\_by(gender, lc) %>%   
 summarise(patient\_no = n(),  
 mean\_inr = mean(b\_inr, na.rm=T))

dat1 %>%   
 group\_by(gender, lc) %>%   
 summarise( N\_alt = sum(!is.na(b\_alt)),  
 MEAN\_alt = mean(b\_alt, na.rm=T),  
 MEDIAN\_alt = median(b\_alt, na.rm=T),  
 MIN\_alt = min(b\_alt, na.rm=T),  
 MAX\_alt = max(b\_alt, na.rm=T))

# Chapter 4. 데이터 분리, 합치기

## 

## 4.1 Merge

library(tidyverse)  
dat <- read\_csv("Example\_data/Ch4\_chb.csv")

### 

### 4.1.1 rbind

a <- c(1,2,3)  
b <- c(4,5,6)  
c <- rbind(a,b)  
c

a <- c(1,2,3,4)  
b <- c(5,6,7)  
c <- rbind(a,b)

### 

### 4.1.2 cbind

a <- c(1,2,3)  
b <- c(4,5,6)  
c <- cbind(a,b)  
c

temp1 <- dat %>%   
 select(id, age, lc, hcc) %>%   
 filter(id<5) %>%   
 print()

temp2 <- dat %>%   
 filter(id<5) %>%   
 select(b\_alt, b\_bil, b\_inr) %>%   
 print()

temp3 <- cbind(temp1, temp2)  
temp3

temp4 <- dat %>%   
 select(id, age, lc, hcc) %>%   
 filter(id<5) %>%   
 print()

temp5 <- dat %>%   
 select(id, age, lc, hcc) %>%   
 filter(id>6) %>%   
 print()

temp6 <- rbind(temp4, temp5)  
temp6

### 

### 4.1.3 merge

temp1 <- dat %>%   
 select(id, age, lc, hcc) %>%   
 filter(id <5) %>%   
 print()

temp2 <- dat %>%   
 select(id, b\_alt, b\_bil, b\_inr) %>%   
 filter(id <5) %>%   
 arrange(desc(id)) %>%   
 print()

temp3 <- cbind(temp1, temp2)  
temp3

temp4 <- merge(temp1, temp2, by='id')   
temp4

temp1 <- dat %>%   
 filter(id <4) %>%   
 select(id, age, gender) %>%   
 print()

temp2 <- dat %>%   
 filter(id %in% c(1,2,4,5,6)) %>%   
 select(id, lc, hcc) %>%   
 print()

merge(temp1, temp2, by='id')

merge(temp1, temp2, by='id', all=TRUE)

merge(temp1, temp2, by='id', all.x=TRUE)

merge(temp1, temp2, by='id', all.y=TRUE)

## 

## 4.2 Tidyverse를 이용한 merge

### 

### 4.2.1 inner join

inner\_join(temp1, temp2, by='id')

### 

### 4.2.2 full join

full\_join(temp1, temp2, by='id')

### 

### 4.2.3 left join

left\_join(temp1, temp2, by='id')

### 

### 4.2.4 right join

right\_join(temp1, temp2, by='id')

### 

### 4.2.5 semi join, anti join

semi\_join(temp1, temp2, by='id')

semi\_join(temp1, temp2, by='id')

inner\_join(temp1, temp2, by='id')

anti\_join(temp1, temp2, by='id')

temp1

temp2

### 

### 4.2.6 intersect, union, setdiff

temp.x <- dat %>%   
 select(id, age, gender) %>%   
 filter(id<4) %>%   
 print( )

temp.y <- dat %>%   
 select(id, age, gender) %>%   
 filter(between (id, 2, 4)) %>%   
 print( )

intersect(temp.x, temp.y)

union(temp.x, temp.y)

setdiff(temp.x, temp.y)

### 

### 4.2.7 bind\_rows, bind\_cols

bind\_rows(temp.x, temp.y)

bind\_cols(temp.x, temp.y)

temp.x

temp.y

bind\_rows(entecavir=temp.x, tenofovir=temp.y, .id = 'treatment')

## 

## 4.3 Tidy 데이터

library(tidyverse)  
dat <- read\_csv('Example\_data/Ch4\_chb2.csv')

### 4.3.2 Tidy 데이터 만들기 연습

alt.long <- dat %>%   
 gather(b\_alt,m6\_alt,m12\_alt,m18\_alt,m24\_alt, key='observation', value='alt\_result') %>%   
 arrange(id)

alt.long <- dat %>%   
 gather(2:6, key='observation', value='alt\_result') %>%   
 arrange(id)

head(alt.long,10)

dim(dat)

dim(alt.long)

alt.wide<-alt.long %>%   
 spread(observation, alt\_result)  
head(alt.wide,10)

dim(alt.long)

dim(alt.wide)

# Chapter 5. 데이터 시각화

library(tidyverse)  
dat <- read\_csv('Example\_data/Ch5\_chb.csv')

dim(dat)

colnames(dat)

dat1<-dat

## 

## 5.1 R base 그래프

### 5.1.1 기본 그래프 그려보기

table(dat1$gender)

barplot(table(dat1$gender))

pie(table(dat1$gender))

hist(dat1$age)

boxplot(dat1$age)

dotchart(dat1$age)

stem(dat1$age)

plot(dat1$b\_alb, dat1$b\_plt)

plot(dat1$age, type='l')

### 

### 5.1.2 기본 그래프의 옵션

plot(dat1$b\_alb, dat1$b\_plt)   
plot(dat1$b\_alb, dat1$b\_plt, pch=17)

plot(dat1$b\_alb, dat1$b\_plt, pch=17)   
plot(dat1$b\_alb, dat1$b\_plt, pch=17, cex=2)

plot(dat1$b\_alb, dat1$b\_plt)   
plot(dat1$b\_alb, dat1$b\_plt, col='blue')

plot(dat1$age, type='l')  
plot(dat1$age, type='l', lty=2)

plot(dat1$age, type='l')  
plot(dat1$age, type='l', lwd=3)

plot(dat1$b\_alb, dat1$b\_plt)   
plot(dat1$b\_alb, dat1$b\_plt, col.axis='blue')

plot(dat1$b\_alb, dat1$b\_plt)   
plot(dat1$b\_alb, dat1$b\_plt, col.lab='blue')

plot(dat1$b\_alb, dat1$b\_plt,  
 main='Albumin and Platelet')   
plot(dat1$b\_alb, dat1$b\_plt,  
 main='Albumin and Platelet',  
 col.main='red')

plot(dat1$b\_alb, dat1$b\_plt)   
plot(dat1$b\_alb, dat1$b\_plt, xlab='Baseline Albumin')  
plot(dat1$b\_alb, dat1$b\_plt, ylab='Baseline Platelet')

plot(dat1$age, dat1$b\_plt,   
 xlab='Age', ylab='Baseline platelet')  
  
fit<-lm(b\_plt~age, data=dat1)

plot(dat1$age, dat1$b\_plt,   
 xlab='Age', ylab='Baseline platelet')  
lines(dat1$age, fit$fitted.values, col='blue', lwd=2)

plot(dat1$age, dat1$b\_plt,   
 xlab='Age', ylab='Baseline platelet')  
abline(v=mean(dat1$age), col='red', lwd=2)  
abline(h=mean(dat1$b\_plt), col='blue', lwd=2)

plot(dat1$id, dat1$b\_alb,   
 xlab='ID', ylab='Baseline Albumin')  
text(dat1$id, dat1$b\_alb,   
 labels=dat1$b\_alb, pos=3)

plot(dat1$b\_alb, dat1$b\_plt)   
plot(dat1$b\_alb, dat1$b\_plt, pch=17)   
plot(dat1$b\_alb, dat1$b\_plt, col='blue')   
plot(dat1$b\_alb, dat1$b\_plt,  
 main='Association between Albumin and Platelet')

## 

## 5.2 ggplot2

### 5.2.1 ggplot2 기본 문법

ggplot(dat1, aes(x=id, y=b\_alb))

ggplot(dat1, aes(x=id, y=b\_alb))+  
 geom\_point()

### 

### 5.2.2 막대 그래프

ggplot(dat1, aes(x=hcc))+  
 geom\_bar()

class(dat1$hcc)

ggplot(dat1, aes( x=factor(hcc) ))+  
 geom\_bar()

ggplot(dat1, aes(x=id, y=b\_alb))+  
 geom\_bar(stat='identity')

ggplot(dat1, aes(x=id, y=b\_alb))+  
 geom\_bar(stat='identity', fill='lightblue')  
  
ggplot(dat1, aes(x=id, y=b\_alb, fill=gender))+  
 geom\_bar(stat='identity')

ggplot(dat1, aes(x=id, y=b\_alb))+  
 geom\_bar(stat='identity', fill=gender)

ggplot(dat1)+  
 geom\_bar(aes(id, b\_alb, fill=gender), stat='identity')

ggplot(dat1)+  
 geom\_bar(aes(dat1$id, dat1$b\_alb, fill=dat1$gender), stat='identity')

ggplot(dat1, aes(x=id, y=b\_alb, fill=gender))+  
 geom\_bar(stat='identity', fill='lightblue')

ggplot(dat1, aes(x= gender, fill=factor(hcc)))+  
 geom\_bar(stat='count')  
  
ggplot(dat1, aes(x= gender, fill=factor(hcc)))+  
 geom\_bar(stat='count', position='dodge')

ggplot(dat1, aes(x= gender, fill=factor(hcc)))+  
 geom\_bar(position='fill')

ggplot(dat1, aes(x= gender, fill=factor(hcc)))+  
 geom\_bar(stat='count', position='dodge')+  
 coord\_flip()

ggplot(dat1, aes(x= gender, fill=factor(hcc)))+  
 geom\_bar(stat='count', position='dodge')+  
 scale\_fill\_manual(values=c('blue','red'))

ggplot(dat1, aes(x= gender, fill=factor(hcc)))+  
 geom\_bar(stat='count', position='dodge')+  
 scale\_fill\_brewer(palette='Greens')

ggplot(dat1, aes(x= gender, fill=factor(hcc)))+  
 geom\_bar(stat='count', position='dodge', color='black')

ggplot(dat1, aes(x= gender, fill=factor(hcc)))+  
 geom\_bar(stat='count', position='dodge', color='black', width=0.5)

ggplot(dat1, aes(x= gender, fill=factor(hcc)))+  
 geom\_bar(stat='count', position='dodge', color='black', alpha=0.5)

ggplot(dat1, aes(x=id, y=b\_alb))+  
 geom\_bar(stat='identity')+  
 geom\_text(aes(label=b\_alb))  
  
ggplot(dat1, aes(x=id, y=b\_alb))+  
 geom\_bar(stat='identity')+  
 geom\_text(aes(label=b\_alb), vjust=-0.5)

### 

### 5.2.3 박스 그래프

ggplot(dat1, aes(x=gender, y=b\_alb))+  
 geom\_boxplot()

ggplot(dat1, aes(x=gender, y=b\_alb))+  
 geom\_boxplot(fill='blue')

ggplot(dat1, aes(x=gender, y=b\_alb))+  
 geom\_boxplot(fill='lightblue', color='blue')

ggplot(dat1, aes(x=gender, y=b\_alb))+  
 geom\_boxplot(fill='blue', color='blue', alpha=0.5)

### 

### 5.2.4 선 그래프

ggplot(dat1, aes(x=id, y=age))+  
 geom\_line()

ggplot(dat1, aes(x=id, y=age))+  
 geom\_line(color='red', linetype=6)  
  
ggplot(dat1, aes(x=id, y=age))+  
 geom\_line(size=3)

ggplot(dat1, aes(x=id, y=age))+  
 geom\_line()+  
 ylim(c(10,80))

### 

### 5.2.5 산점도

ggplot(dat1, aes(x=id, y=age))+  
 geom\_point()

ggplot(dat1, aes(x=id, y=age))+  
 geom\_point(shape=1)

ggplot(dat1, aes(x=id, y=age, color=factor(lc)))+  
 geom\_point()

ggplot(dat1, aes(x=id, y=age, shape=factor(lc)))+  
 geom\_point()

ggplot(dat1, aes(x=id, y=age))+  
 geom\_point(shape=factor(dat1$lc))

ggplot(dat1, aes(x=id, y=age, color=factor(lc), size=b\_alb))+  
 geom\_point()+  
 geom\_text(aes(label=b\_alb), vjust=-2, size=3)+  
 scale\_color\_brewer(palette='Set1')

### 

### 5.2.6 버블 그래프

ggplot(dat1, aes(x=age, y=b\_alb))+  
 geom\_point(aes(size=b\_alt),shape=21, color='black', fill='orange')

### 

### 5.2.7 히스토그램

ggplot(dat1, aes(x=b\_alb))+  
 geom\_histogram()

ggplot(dat1, aes(x=b\_alb))+  
 geom\_histogram(binwidth = 0.5)

ggplot(dat1, aes(x=b\_alb))+  
 geom\_histogram(binwidth=0.2, fill='lightblue',color='black')

ggplot(dat1, aes(x=b\_alb, fill=factor(lc)))+  
 geom\_histogram(binwidth = 0.2)

ggplot(dat1, aes(x=b\_alb, fill=factor(lc)))+  
 geom\_histogram(position='identity', alpha=0.5, binwidth = 0.2)

ggplot(dat1, aes(x=b\_alb))+  
 geom\_histogram(binwidth = 0.2)+  
 facet\_grid(lc~.)

### 

### 5.2.8 밀도 그래프

ggplot(dat1, aes(x=b\_alb))+  
 geom\_density()

ggplot(dat1, aes(x=b\_alb, color=factor(lc)))+  
 geom\_density()

ggplot(dat1, aes(x=b\_alb, fill=factor(lc)))+  
 geom\_density(alpha=0.5)

## 

## 5.3 ggplot2의 다양한 옵션

albu<-dat1 %>%   
 select(id, age, gender, treat\_gr, lc, contains('alb')) %>%   
 gather(6:10, key='observation', value='albumin')  
  
head(albu,10)

str(albu)

albu$gender<-factor(albu$gender, levels = c('M','F'))  
albu$treat\_gr<-factor(albu$treat\_gr)  
albu$lc<-factor(albu$lc)

### 5.3.1 Axis (축)

ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()

ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()+  
 coord\_flip()

ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()+  
 ylim(1,5.5)

ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()+  
 scale\_y\_continuous(limits=c(1,5), breaks=c(seq(1,5,0.5)))

ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()+  
 ylim(0,6)+  
 scale\_y\_continuous(limits=c(1,5), breaks=c(seq(1,5,0.5)))

ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()+  
 scale\_y\_continuous(labels=c('very low','low','normal','high'))

ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()+  
 scale\_y\_continuous(labels=c('very low','low','normal','high'))+  
 theme(axis.text.y=element\_text(angle=45))

ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()+  
 xlab('Sex')+  
 ylab('Serum Albumin Level at Baseline')

ggplot(albu, aes(x=age, y=albumin))+  
 geom\_point()+  
 scale\_x\_continuous(breaks=c(20,30,40,50,60,70))  
  
ggplot(albu, aes(x=age, y=albumin))+  
 geom\_point()+  
 scale\_x\_log10(breaks=c(20,30,40,50,60,70))

### 5.3.2 Annotate (주석)

ggplot(albu, aes(x=age, y=albumin))+  
 geom\_point()+  
 annotate('text',x=20, y=5, label='Young Age', color='red', size=5)+  
 annotate('text',x=50, y=5, label='Old Age', color='blue', size=5)

ggplot(albu, aes(x=age, y=albumin))+  
 geom\_point()+  
 geom\_abline(intercept=0, slope=0.1, color='red')

albu1<-albu %>%  
 group\_by(lc) %>%   
 summarize( mean\_albumin = mean(albumin, na.rm=T),  
 sd\_albumin = sd(albumin, na.rm=T))  
head(albu1)

ggplot(albu1, aes(x=lc, y=mean\_albumin))+  
 geom\_bar(stat='identity')+  
 ylim(c(0,5))+  
 geom\_errorbar(aes(ymin=mean\_albumin-sd\_albumin,  
 ymax=mean\_albumin+sd\_albumin))

ggplot(albu1, aes(x=lc, y=mean\_albumin))+  
 geom\_bar(stat='identity')+  
 ylim(c(0,5))+  
 geom\_errorbar(aes(ymin=mean\_albumin-sd\_albumin,  
 ymax=mean\_albumin+sd\_albumin), width=0.2)

### 

### 5.3.3 Legend (범례)

ggplot(albu, aes(x=gender, y=albumin, fill=lc))+  
 geom\_boxplot()+  
 theme(legend.position = 'top')

ggplot(albu, aes(x=gender, y=albumin, fill=lc))+  
 geom\_boxplot()+  
 theme(legend.position = 'bottom')

ggplot(albu, aes(x=gender, y=albumin, fill=lc))+  
 geom\_boxplot()+  
 theme(legend.position = 'left')

ggplot(albu, aes(x=gender, y=albumin, fill=lc))+  
 geom\_boxplot()+  
 theme(legend.position = 'right')

ggplot(albu, aes(x=gender, y=albumin, fill=lc))+  
 geom\_boxplot()+  
 theme(legend.position = c(1,1))

ggplot(albu, aes(x=gender, y=albumin, fill=lc))+  
 geom\_boxplot()+  
 theme(legend.position = c(0.9,0.8))

ggplot(albu, aes(x=gender, y=albumin, fill=lc))+  
 geom\_boxplot()+  
 theme(legend.position = c(0.9,0.8),  
 legend.background = element\_blank())+  
 labs(fill='Liver Cirrhosis')

### 

### 5.3.4 Facet grid (분할)

ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()+  
 facet\_grid(lc~.)

ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()+  
 facet\_grid(~lc)

ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()+  
 facet\_grid(~lc+treat\_gr)

### 

### 5.3.5 Theme (테마)

ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()+  
 facet\_grid(~lc)+  
 ggtitle('Albumin and LC')

ggplot(albu, aes(x=age, y=albumin, color=lc))+  
 geom\_point()+  
 theme\_bw()

### 

### 5.3.6 한 화면에 그래프 여러개 그리기

alb1<-ggplot(albu, aes(x=gender, y=albumin))+  
 geom\_boxplot()+  
 facet\_grid(~lc)  
alb1

alb2<-ggplot(albu, aes(x=age, y=albumin, color=lc))+  
 geom\_point()  
alb2

library(gridExtra)  
  
grid.arrange(alb1,alb2)

grid.arrange(alb1, alb2, nrow=1)

### 

### 5.3.7 ggplot2 클릭만으로 하기

library(esquisse)  
library(officer)  
library(rvg)

esquisser()

### 

### 5.3.8 출판을 위한 출력

pdf('plot1.pdf',width=8, height=8)  
ggplot(albu, aes(x=gender, y=albumin, fill=lc))+  
 geom\_boxplot()

tiff('plot1.tiff',width=300, height=400)  
ggplot(albu, aes(x=gender, y=albumin, fill=lc))+  
 geom\_boxplot()

tiff('plot1.tiff',width=1200, height=1800, res=300)  
ggplot(albu, aes(x=gender, y=albumin, fill=lc))+  
 geom\_boxplot()

ggplot(albu, aes(x=gender, y=albumin, fill=lc))+  
 geom\_boxplot()

ggsave('albumin\_graph.pdf', width = 10, height=12)

ggplot(albu, aes(x=gender, y=albumin, fill=lc))+  
 geom\_boxplot()

ggsave('albumin\_graph.tiff', width = 6, height=8, dpi=300)

## 

## 5.4 ggpubr 패키지

install.packages('ggpubr')

library(ggpubr)

### 

### 5.4.1 히스토그램

gghistogram(dat1,  
 x='age',  
 color='gender',  
 fill='gender')

gghistogram(dat1,  
 x='age',  
 color='gender', fill='gender',  
 palette=c('blue','red'),  
 rug=TRUE)

### 

### 5.4.2 밀도 그래프

ggdensity(dat1,  
 x='age',  
 color='gender', fill='gender',  
 rug=TRUE)

### 

### 5.4.3 박스 그래프

ggboxplot(dat1,  
 x='gender', y='b\_alb',  
 color='gender')

ggboxplot(dat1,  
 x='gender', y='b\_alb',  
 color='gender')+  
 stat\_compare\_means(comparisons= list(c('M','F')))

### 

### 5.4.4 막대 그래프

ggbarplot(dat1,  
 x='id', y='b\_alb',  
 fill='gender')

ggbarplot(dat1,  
 x='id', y='b\_alb',  
 fill='gender',  
 sort.val='desc')

ggbarplot(dat1,  
 x='id', y='b\_alb',  
 fill='gender',  
 sort.val='desc',  
 sort.by.group=FALSE)

### 

### 5.4.5 워터폴 (waterfall) 그래프

water.dt<-dat1 %>%   
 select(id,gender,b\_alb,m6\_alb) %>%   
 mutate(delta\_alb=b\_alb-m6\_alb)  
   
water.dt

ggbarplot(water.dt,   
 x='id', y='delta\_alb',  
 fill='gender')

ggbarplot(water.dt,   
 x='id', y='delta\_alb',  
 fill='gender',  
 sort.val='desc',  
 sort.by.groups = FALSE)

ggbarplot(water.dt,   
 x='id', y='delta\_alb',  
 fill='gender',  
 sort.val='desc',  
 sort.by.groups = FALSE,  
 rotate=TRUE)

### 

### 5.4.6 산점도

ggscatter(dat1,  
 x='age',  
 y='b\_alb',  
 color='gender')

install.packages('ggExtra')

library(ggExtra)  
  
ggscatter(dat1, x='age', y='b\_alb',  
 color='gender') %>%  
 ggMarginal(type='density')

ggscatter(dat1, x='age', y='b\_alb',  
 color='gender') %>%  
 ggMarginal(type='boxplot')

ggscatter(dat1, x='age', y='b\_alb',  
 color='gender') %>%  
 ggMarginal(type='histogram', fill='orange')

## 

## 5.5 상관관계를 그려주는 패키지

### 5.5.1 GGally 패키지

install.packages('GGally')

library(GGally)

temp<-dat1[, c('age','gender','b\_alt','b\_plt','b\_alb')]  
ggpairs(temp)

### 

### 5.5.2 corrplot 패키지

install.packages('corrplot')

library(corrplot)

cor.dt<-dat1 %>%   
 select(age, b\_alt, b\_plt, b\_alb) %>%   
 na.omit()

corrplot(cor(cor.dt))

par(mfrow=c(2,2))   
corrplot(cor(cor.dt), method='square')  
corrplot(cor(cor.dt), method='ellipse')  
corrplot(cor(cor.dt), method='number')  
corrplot(cor(cor.dt), method='color')

# Chapter 6. 임상 연구관련 의학 통계

## 6.1 회귀 분석

### 6.1.1 단순 선형 회귀 분석

library(tidyverse)  
  
dt<-read\_csv("Example\_data/Ch6\_regression.csv")  
head(dt)

plot(osm~na, data=dt)

fit<-lm(osm~na, data = dt)  
fit

plot(osm~na, data=dt)  
abline(fit, col='red', lwd=2)

summary(fit)

par(mfrow=c(2,2))  
plot(fit)

### 

### 6.1.2 다중 회귀 분석

plot(osm~bun, data=dt, main="BUN")  
plot(osm~glucose, data=dt, main="Glucose")  
plot(osm~height, data=dt, main="Height")  
plot(osm~weight, data=dt, main="Weight")

fit.multi<-lm(osm~na+bun+glucose+height+weight, data=dt)  
fit.multi

summary(fit.multi)

plot(fit.multi)

fit1<-lm(osm~na, data=dt)  
f1<-summary(fit1)  
f1$adj.r.squared

fit2<-lm(osm~na+bun, data=dt)  
f2<-summary(fit2)  
f2$adj.r.squared

fit3<-lm(osm~na+bun+glucose, data=dt)  
f3<- summary(fit3)  
f3$adj.r.squared

fit4<-lm(osm~na+bun+glucose+height, data=dt)  
f4<-summary(fit4)  
f4$adj.r.squared

fit5<-lm(osm~na+bun+glucose+height+weight, data=dt)  
f5<-summary(fit5)  
f5$adj.r.squared

fit.multi<-lm(osm~na+bun+glucose+height+weight, data=dt)  
step(fit.multi)

install.packages('olsrr')

library(olsrr)  
  
fit.multi<-lm(osm~na+bun+glucose+height+weight, data=dt)

ols\_step\_all\_possible(fit.multi)

ols\_step\_best\_subset(fit.multi)

ols\_step\_forward\_aic(fit.multi)

plot(ols\_step\_forward\_aic(fit.multi))+theme\_bw()

ols\_step\_backward\_aic(fit.multi)

plot(ols\_step\_backward\_aic(fit.multi))+theme\_bw()

## 

## 6.2 일반화 선형 분석

### 6.2.1 로지스틱 회귀 분석

dt1<-read\_csv("Example\_data/Ch6\_logistic.csv")  
head(dt1)

dt1 %>%   
 count(group)

plot(dt1$id, dt1$group)

sbp<-c(110, 120, 130, 135, 140, 150, 145, 142, 130, 128)  
mean(sbp)

plot(sbp)

library(moonBook)  
mytable(group~aspirin+ibd+diabetes+gender+age, data=dt1)

fit<-glm(group~age+gender+ibd+cirrhosis+diabetes+htn+aspirin,   
 family=binomial, data=dt1)  
fit

summary(fit)

step(fit, type='backward')

final.fit<-glm(group~age+ibd+cirrhosis+aspirin,   
 family=binomial, data=dt1)  
extractOR(fit.final)

install.packages(fmsb)

library(fmsb)  
NagelkerkeR2(fit.final)

### 

### 6.2.2 모형의 성능

install.packages('performance')  
install.packages('see')  
install.packages('patchwork')

library(performance)  
library(see)  
library(patchwork)

fit.final<-glm(group~age+ibd+cirrhosis+aspirin,   
 family=binomial, data=dt1)  
r2\_nagelkerke(fit)

performance\_hosmer(fit.final)

fit<-lm(osm~na+bun+glucose+height+weight, data=dt)  
check\_model(fit)

fit<-lm(osm~na+bun+glucose+height+weight, data=dt)  
model\_performance(fit)

fit.logi<-glm(group~age+gender+ibd+cirrhosis+diabetes+htn+aspirin,   
 family=binomial, data=dt1)

fit<-lm(osm~na+bun+glucose+height+weight, data=dt)  
fit1<-lm(osm~na+bun+glucose, data=dt)  
compare\_performance(fit, fit1, rank = TRUE)

## 

## 6.3 ROC 관련 분석

### 6.3.4 ROC 곡선 직접 그려보기

roc.ex<-read\_csv("Example\_data/Ch6\_afp.csv")  
head(roc.ex)

roc.ex<-roc.ex %>%   
 arrange(desc(afp))  
roc.ex

### 

### 6.3.5 pROC 패키지

install.packages('pROC')

library(pROC)  
  
afp<-roc(roc.ex$group, roc.ex$afp, ci=TRUE)  
afp

plot(afp)

afp<-roc(roc.ex$group, roc.ex$afp)  
plot(afp,legacy.axes=TRUE)

afp<-roc(roc.ex$group, roc.ex$afp)  
pivka<-roc(roc.ex$group, roc.ex$pivka)  
  
plot(afp, col='blue', legacy.axes=TRUE)  
plot(pivka, col='red', legacy.axes=TRUE, add=TRUE)  
legend(0.3, 0.2, legend=c("AFP", "PIVKA"),  
 col=c("blue", "red"), lty=1:1, cex=0.8)

roc.test(afp, pivka)

### 

### 9.3.6 최적의 cut-off 찾기

ci.thresholds(afp, conf.level=0.95, boot.n=1000,  
 thresholds='best')

### 

### 9.3.7 특정 cut-off에서 민감도, 특이도 계산하기

metric<-c('sensitivity','specificity','ppv','npv')  
  
afp.cutoff<-ci.coords(afp, x=5, input="threshold", metric)  
afp.cutoff

afp.cutoff$sensitivity

afp.cutoff$specificity

afp.cutoff$ppv

afp.cutoff$npv

### 

### 6.3.8 Epi 패키지

library(Epi)  
  
roc.ex$new\_gr<-ifelse(roc.ex$group=='HCC',1,0)

ROC(form=new\_gr~afp, data=roc.ex, plot='ROC')  
ROC(form=new\_gr~pivka, data=roc.ex, plot='ROC')  
ROC(form=new\_gr~afp+pivka, data=roc.ex, plot='ROC')

## 

## 6.4 생존 분석

### 6.4.1 Time to event 분석

library(survival)  
  
suv.dt<-read\_csv('Example\_data/Ch6\_survival.csv')

library(lubridate)  
  
suv.dt$hcc\_period <- suv.dt$hcc\_date - suv.dt$start\_date  
suv.dt$hcc\_period <- as.numeric(suv.dt$hcc\_period)  
summary(suv.dt$hcc\_period)

suv.dt$hcc\_period <- suv.dt$hcc\_period / 365.25  
summary(suv.dt$hcc\_period)

suv.dt<-suv.dt %>%   
 mutate(hcc\_period = hcc\_date - start\_date) %>%   
 mutate(hcc\_period = as.numeric(hcc\_period)/365.25) %>%   
  
summary(suv.dt$hcc\_period)

### 

### 6.4.2 Kaplan-Meier 곡선

f1<-survfit(Surv(hcc\_period, hcc)~1, data=suv.dt)

plot(f1,  
 xlab='Obervation period',  
 ylab='Survival probability')

library(survminer)  
  
ggsurvplot(f1, risk.table = TRUE)

### 

### 6.4.3 5년 생존율 계산

f1<-survfit(Surv(hcc\_period, hcc)~1, data=suv.dt)

summary(survfit(Surv(hcc\_period, hcc)~1, data=suv.dt),  
 times=5)

plot(f1, conf.int=FALSE)  
points(x=5, y=0.665, pch=19)  
segments(5,-0.1, 5,0.665, col='red')  
segments(-1,0.665, 5,0.665,col='red', lty=2)  
text(x=0+0.1, y=0.665+0.1, labels=c('66.5%'), col='red')

### 

### 6.4.4 Median survival 계산

median(suv.dt$hcc\_period)

survfit(Surv(hcc\_period, hcc)~1, data=suv.dt)

plot(f1, conf.int=FALSE)  
points(x=8, y=0.5, pch=19)  
segments(8,-0.1, 8,0.5, col='red')  
segments(-1,0.5, 8,0.5,col='red', lty=2)  
text(x=8, y=0.6, labels=c('50.0%'), col='red')

### 

### 6.4.5 두그룹에서 생존 함수 비교

survdiff(Surv(hcc\_period, hcc)~lc, data=suv.dt)

f1<-survfit(Surv(hcc\_period, hcc)~lc, data=suv.dt)  
plot(f1, conf.int=FALSE, col=c('blue','red'))  
text(x=9, y=0.8, labels=c('lc=0'), col='blue')  
text(x=9, y=0.2, labels=c('lc=1'), col='red')

### 

### 6.4.6 Survminer 패키지

suv.dt1<-read\_csv('Example\_data/Ch6\_survival1.csv')  
dim(suv.dt1)

f1<-survfit(Surv(death\_yr, death)~1, data=suv.dt1)  
ggsurvplot(f1)

f1.hcc<-survfit(Surv(death\_yr, death)~hcc, data=suv.dt1)  
ggsurvplot(f1.hcc)

f2<-survfit(Surv(death\_yr, death)~1, data=suv.dt1)  
ggsurvplot(f2,  
 conf.int = FALSE,  
 fun = 'event',  
 ylim=c(0,1),  
 ggtheme=theme\_bw())

f2.hcc<-survfit(Surv(death\_yr, death)~hcc, data=suv.dt1)  
ggsurvplot(f2.hcc,  
 fun='event',  
 pval=TRUE,  
 risk.table='abs\_pct',  
 palette=c('red','blue'),  
 break.time.by=1,  
 legend='top',  
 legend.title='HCC',  
 legend.labs=c('None','Present'),  
 xlab=c('Years after treatment'),  
 ylab=c('Cumulative incidence of HCC'),  
 ylim=c(0,1),  
 surv.median.line = 'hv',  
 ncensor.plot=TRUE)

### 6.4.6 Cox 비례위험모형

library(moonBook)  
suv.dt2 <-read\_csv('Example\_data/Ch6\_survival2.csv')  
  
f1.lc<-coxph(Surv(hcc\_yr, hcc)~lc, data=suv.dt2)  
extractHR(f1.lc)

f1.lc<-survfit(Surv(hcc\_yr, hcc)~lc, data=suv.dt2)  
ggsurvplot(f1.lc,  
 fun='event',  
 pval=TRUE,  
 risk.table=TRUE,  
 break.time.by=1,  
 xlab=c('Year after treatment'),  
 ylab=c('Cumulative incidence of HCC'),  
 ylim=c(0,1))

### 6.4.7 Cox 모형을 이용하여 단변량, 다변량 분석

suv.dt2$TS<-Surv(suv.dt2$hcc\_yr,suv.dt2$hcc)  
  
mycph(TS~gender+age+lc+dm+hbeag, data=suv.dt2)

library(gtsummary)  
  
suv.dt2 %>%   
 select(-id, -TS, -death, -death\_yr) %>%   
 tbl\_uvregression(method=coxph,  
 y=Surv(hcc\_yr, hcc),  
 exponentiate=TRUE)

f1.multi<-coxph(Surv(hcc\_yr,hcc)~age+gender+lc+dm+hbeag, data=suv.dt2)  
extractHR(f1.multi)

f1.final<-step(f1.multi, direction = 'backward')

extractHR(f1.final)

coxph(Surv(hcc\_yr, hcc==1)~age+gender+lc+dm+hbeag, data=suv.dt2) %>%   
 tbl\_regression(exponentiate=TRUE)

cox.uni<-suv.dt2 %>%   
 select(hcc, hcc\_yr, age, gender, lc, dm, hbeag) %>%   
 tbl\_uvregression(method=coxph,  
 y=Surv(hcc\_yr, hcc),  
 exponentiate = TRUE)  
cox.uni

cox.multi<-coxph(Surv(hcc\_yr, hcc)~age+lc, data=suv.dt2) %>%   
 tbl\_regression(exponentiate=TRUE)  
cox.multi

cox.table<-tbl\_merge(  
 tbls = list(cox.uni, cox.multi),  
 tab\_spanner = c("\*\*Univariate analysis\*\*","\*\*Multivariable analysis\*\*")  
)  
cox.table

### 

### 6.4.9 Forest plot 그리기

f1.cox<-coxph(Surv(hcc\_yr, hcc==1)~age+gender+lc+dm+hbeag, data=suv.dt2)  
ggforest(f1.cox)

### 

### 6.4.10 Cox 모형 검증

f1.cox<-coxph(Surv(hcc\_yr, hcc==1)~age+gender+lc+dm+hbeag, data=suv.dt2)  
  
cox.zph(f1.cox)

ftest<-cox.zph(f1.cox)  
ggcoxzph(ftest)

## 6.5 Time dependent Cox model

dt.time<-read\_csv('Example\_data/Ch6\_survival3.csv')  
head(dt.time)

f1.hcc<-survfit(Surv(hcc\_yr, hcc)~alt\_nl, data=dt.time)  
  
ggsurvplot(f1.hcc,  
 fun='event',  
 risk.table=TRUE,  
 break.time.by=1,  
 xlim=c(0,5),  
 ylim=c(0,0.3),  
 pval = TRUE)

dt.time1<-tmerge(dt.time, dt.time, id=id, HCC=event(hcc\_yr, hcc))  
  
dt.time1<-tmerge(dt.time1, dt.time1, id=id, ALT=tdc(alt\_duration, alt\_nl))  
  
dt.time1$ALT[is.na(dt.time1$ALT)]<-c('abnormal')  
  
head(dt.time[,c('id','hcc\_yr','hcc','alt\_nl','alt\_duration')])

head(dt.time1[,c('id','hcc\_yr','hcc','alt\_nl','alt\_duration','tstart','tstop','HCC','ALT')],11)

f1.time<-coxph(Surv(tstart, tstop, HCC==1)~ALT+cluster(id), data=dt.time1)  
  
extractHR(f1.time)

# Chapter 7. 연구 따라하기 (1): 아스피린과 간담도암

## 7.3 데이터 분석

### 7.3.1 데이터 구조 파악하기

library(tidyverse)  
library(moonBook)  
library(gtsummary)

dat <- read\_csv('Example\_data/Ch7\_aspirin.csv')

dim(dat)

### 

### 7.3.2 데이터 전처리하기 (1)

colnames(dat)

dat1<-dat  
names(dat1) <- tolower(names(dat1))  
names(dat1)

glimpse(dat1)

sapply(dat1, class)

colSums(is.na(dat1))

dat1 <- dat1 %>%   
 select(id, age, everything())   
colnames(dat1)

dat1$group <- as.factor(dat1$group)  
dat1$gender <- as.factor(dat1$gender)  
dat1$racegroup <- as.factor(dat1$racegroup)

for (i in 3:19){   
 dat1[ ,i]<-as.factor(dat1[ ,i])   
}

dat1 <- dat1 %>%   
 mutate\_at(vars(group:smoking), as.factor)   
sapply(dat1, class)

### 

### 7.3.3 데이터 전처리하기 (2)

dat1$id <- factor(dat1$id)  
dat1$group <- factor(dat1$group,   
 labels=c('control', 'case'))  
dat1$gender <- factor(dat1$gender,   
 labels=c('male', 'female'))  
dat1$racegroup <- factor(dat1$racegroup,   
 labels=c('white','black','asian'))  
dat1$location <- factor(dat1$location,   
 labels=c('iCCA','pCCA','dCCA'))  
dat1$obesity <- factor(dat1$obesity,   
 labels=c('normal','obesity','unknown'))  
dat1$aspirin <- factor(dat1$aspirin,   
 labels=c('non-user','aspirin-user'))  
dat1$frequency <- factor(dat1$frequency,   
 labels=c('non-user','non-daily user','daily user'))  
dat1$duration <- factor(dat1$duration,   
 labels=c('non-user','<3 years','\U2265 3 years','unknown'))  
dat1$dose <- factor(dat1$dose,   
 labels=c('non-user','81-162mg/day','\U2265 325mg/day','unknown'))

dat1 <- dat1 %>%   
 mutate\_at(vars(cva:smoking), ~ifelse(.==1, 'present','none')) %>%   
 mutate\_at(vars(cva:smoking), as.factor)

glimpse(dat1)

dat1<-dat1 %>%   
 mutate(age\_gr=cut(age,  
 c(-Inf,40,50,60,70,Inf),  
 c('<40','40-49','50-59','60-69','\U2265 70')))

dat1 %>%   
 count(age\_gr)

### 7.3.4 Table 1 만들기

colnames(dat1)  
  
mytable(group~aspirin+age\_gr+gender+racegroup+  
 obesity+cva+cad+htn+diabetes+psc+  
 cirrhosis+hbv+smoking,  
 data=dat1)

colnames(dat1)  
mytable(group~.-id-location-frequency-duration-dose,   
 data=dat1)

dat.sub <- dat1 %>%   
 filter(group=='case')  
  
mytable(location~.-id-group-frequency-duration-dose,   
 data=dat.sub)

## 

## 7.4 단변량 분석

### 7.4.1 Crosstable 만들기

table(dat1$group, dat1$gender)

xtabs(~group+gender, data=dat1)

install.packages('crosstable')

library(crosstable)  
crosstable(dat1, c('gender'), by=group)

crosstable(dat1, c('gender'), by=group) %>%   
 as\_flextable()

crosstable(dat1, c('gender'), by=group, total='both') %>%   
 as\_flextable()

crosstable(dat1, c('gender','hbv'), by=group, total='both') %>%   
 as\_flextable()

crosstable(dat1, c('gender'), by=group, test=TRUE) %>%   
 as\_flextable()

### 

### 7.4.2 Odds ratio 계산하기

fit.gender <- glm(group~gender, family=binomial, data = dat1)  
summary(fit.gender)

extractOR(fit.gender)

### 

### 7.4.3 Odds ratio 그래프

ORplot(fit.gender, type=1)  
ORplot(fit.gender, type=2)  
ORplot(fit.gender, type=3)

### 

### 7.4.4 모든 변수 단변량 분석 하기

fit.aspirin <- glm(group~aspirin, family=binomial, data=dat1)   
fit.age\_gr <- glm(group~age\_gr, family=binomial, data=dat1)   
fit.gender <- glm(group~gender, family=binomial, data=dat1)   
fit.smoking <- glm(group~smoking, family=binomial, data=dat1)

uni.log<-function(variable){  
 formula <- as.formula(paste('group~', variable))  
 result.log <- glm(formula, data=dat1, family=binomial)  
 extractOR(result.log)  
}

uni.log('age\_gr')

var <- c('aspirin','age\_gr','gender','racegroup','cirrhosis')  
lapply(var, function(x)uni.log(x))

### 

### 7.4.5 gtsummary 패키지 이용하기

dat1 %>%   
 select(-id,-age,-location,-frequency,-duration,-dose) %>%   
 tbl\_uvregression(method=glm,  
 y=group,  
 method.args = list(family=binomial),  
 exponentiate = TRUE)

dat1 %>%   
 select(-id,-age,-location,-frequency,-duration,-dose) %>%   
 tbl\_uvregression( method=glm,  
 y=group,  
 method.args = list(family=binomial),  
 exponentiate = TRUE,  
 show\_single\_row = c('aspirin','gender','cva',  
 'cad','htn','diabetes',  
 'psc','cirrhosis','hbv',  
 'smoking')) %>%   
 bold\_labels() %>%   
 bold\_p()

uni.var<-names(dat1[ ,c(12:19)])   
  
dat1 %>%   
 select(-id,-age,-location,-frequency,-duration,-dose) %>%   
 tbl\_uvregression( method=glm,  
 y=group,  
 method.args = list(family=binomial),  
 exponentiate = TRUE,  
 show\_single\_row = uni.var) %>%   
 bold\_labels() %>%   
 bold\_p()

dat1 %>%   
 select(-id,-age,-location,-frequency,-duration,-dose) %>%   
 tbl\_uvregression( method=glm,  
 y=group,  
 method.args = list(family=binomial),  
 exponentiate = TRUE,  
 show\_single\_row = uni.var,  
 pvalue\_fun = function(x) style\_pvalue(x, digits = 2)) %>%   
 bold\_labels() %>%   
 bold\_p(t=0.001)

## 

## 7.5 다변량 분석

### 7.5.1 다변량 분석하기

fit.multi <- glm(group~aspirin+age\_gr+gender+racegroup+  
 obesity+cva+cad+htn+diabetes+  
 psc+cirrhosis+hbv+smoking,  
 family=binomial,  
 data=dat1)  
summary(fit.multi)

extractOR(fit.multi)

fit.multi2 <-step(fit.multi)

step(fit.multi, direction='backward')  
step(fit.multi, direction='forward')  
step(fit.multi, direction='both')

extractOR(fit.multi2)

ORplot(fit.multi2, type=3)

### 

### 7.5.2 Forestmodel 패키지

install.packages(forestmodel)  
install.packages(rlang)  
install.packages(broom)

library(forestmodel)  
library(rlang)  
library(broom)  
forest\_model(fit.multi2)

### 

### 7.5.3 Finalfit 패키지

library(finalfit)  
  
var1 <- c('aspirin','diabetes','psc','cirrhosis','smoking')  
  
dat1 %>%   
 finalfit('group', var1, metrics=TRUE)

dat1 %>%   
 or\_plot('group', var1)

## 

## 7.5 결과 제시

### 7.5.1 Table 1

dat2<-dat1   
  
dat2$location<-as.character(dat2$location)   
dat2<-dat2 %>%   
 mutate(subgroup = ifelse(group=='case', location, 'control'))  
  
dat2 %>%   
 count(subgroup, location)

dat2$subgroup<-factor(dat2$subgroup,   
 levels=c('iCCA','pCCA','dCCA','control'))

mytable(subgroup~aspirin+age\_gr+gender+racegroup+obesity+  
 cva+cad+htn+diabetes+psc+cirrhosis+hbv+smoking,   
 data=dat2, show.total=TRUE)

table1<-dat2 %>%   
 select(-id, -age, -location, -frequency, -duration, -dose) %>%   
 tbl\_summary(by=subgroup) %>%   
 add\_overall() %>%   
 modify\_spanning\_header( c('stat\_1','stat\_2','stat\_3')~'\*\*Case group\*\*') %>%   
 modify\_caption('\*\*Table 1. Baseline characteristics of the study population\*\*')

table1 %>%   
 as\_flex\_table()7.5.2 Table 2

colnames(dat2)

[1] "id" "age" "group" "gender" "racegroup" "location"   
 [7] "obesity" "aspirin" "frequency" "duration" "dose" "cva"   
[13] "cad" "htn" "diabetes" "psc" "cirrhosis" "hbv"   
[19] "smoking" "age\_gr" "subgroup"

single<-c('aspirin','cva','cad','htn',  
 'diabetes','psc','cirrhosis','hbv','smoking') # none은 표기를 하지 않을 변수 저장해두기  
  
uni.table<-dat2 %>%   
 select(group:aspirin, cva:smoking) %>%   
 tbl\_uvregression( method=glm,  
 y=group,  
 method.args = list(family=binomial),  
 exponentiate = TRUE,  
 show\_single\_row = all\_of(single)) %>%   
 bold\_labels() %>%   
 bold\_p()  
  
  
multi.table<-tbl\_regression(fit.multi2,   
 exponentiate = T,  
 show\_single\_row = c('aspirin','diabetes',  
 'psc','cirrhosis','smoking')) %>%   
 bold\_labels() %>%   
 bold\_p()  
  
  
table2<-tbl\_merge(tbls=list(uni.table, multi.table),  
 tab\_spanner = c('\*\*Univariate analysis\*\*','\*\*Multivariable analysis\*\*')) %>%   
 modify\_caption('\*\*Table 2. Risk factors for CCA\*\*')

table2 %>%   
 as\_flex\_table()

### 

### 7.5.3 Table 3

icca<-dat2 %>%   
 filter(location=='iCCA')  
  
fit.icca <- glm(group~aspirin+age\_gr+gender+racegroup+  
 obesity+cva+cad+htn+diabetes+  
 psc+cirrhosis+hbv+smoking,  
 family=binomial,  
 data=icca)  
fit.icca2<-step(fit.icca, direction='backward')

multi.icca<-tbl\_regression(fit.icca2, exponentiate = T,  
 show\_single\_row = c('aspirin','diabetes','psc','cirrhosis')) %>%   
 bold\_labels() %>%   
 bold\_p()

pcca<-dat2 %>%   
 filter(location=='pCCA')  
  
fit.pcca <- glm(group~aspirin+age\_gr+gender+racegroup+  
 obesity+cva+cad+htn+diabetes+  
 psc+cirrhosis+hbv+smoking,  
 family=binomial,  
 data=pcca)  
fit.pcca2<-step(fit.pcca, direction='backward')

multi.pcca<-tbl\_regression(fit.pcca2, exponentiate = T,  
 show\_single\_row = c('aspirin','cad','htn','diabetes','psc','cirrhosis','hbv','smoking')) %>%   
 bold\_labels() %>%   
 bold\_p()

dcca<-dat2 %>%   
 filter(location=='dCCA')  
  
fit.dcca <- glm(group~aspirin+age\_gr+gender+racegroup+  
 obesity+cva+cad+htn+diabetes+  
 psc+cirrhosis+hbv+smoking,  
 family=binomial,  
 data=dcca)  
fit.dcca2<-step(fit.dcca, direction='backward')

multi.dcca<-tbl\_regression(fit.dcca2, exponentiate = T,  
 show\_single\_row = c('aspirin','obesity','diabetes')) %>%   
 bold\_labels() %>%   
 bold\_p()  
  
table3<-tbl\_merge(tbls=list(multi.icca, multi.pcca, multi.dcca),  
 tab\_spanner = c('\*\*Intrahepatic CCA\*\*','\*\*Perihilar CCA\*\*','\*\*Distal CCA\*\*')) %>%   
 modify\_caption('\*\*Table 3. Multivariable analysis for each CCA subtype\*\*')

table3 %>%   
 as\_flex\_table()

### 7.5.4 Table 4

mytable(group+aspirin~age+age\_gr+gender+racegroup+obesity+  
 cva+cad+htn+diabetes+psc+cirrhosis+hbv+smoking,  
 data=dat2)

### 

### 7.5.5 Table 5

dat2 %>%   
 count(aspirin)

asp<-dat2 %>%   
 filter(aspirin=='aspirin-user')  
  
asp %>%   
 select(group, duration, frequency, dose) %>%   
 tbl\_summary(by=group) %>%   
 add\_p()table5<-asp %>%   
 select(group, duration, frequency, dose) %>%   
 tbl\_uvregression(method=glm,  
 y=group,  
 method.args = list(family=binomial),  
 exponentiate = TRUE) %>%   
 bold\_labels() %>%   
 bold\_p() %>%   
 modify\_caption('\*\*Table 5. Association between duration, dose, and frequency of aspirin use and the risk of CCA\*\*')

table5 %>%   
 as\_flex\_table()

# Chapter 8. 연구 따라하기 (2) 종양표지자와 간암

## 8.3 데이터 분석하기

library(tidyverse)  
library(readr)  
library(moonBook)  
library(gtsummary)  
library(lubridate)  
library(ggpubr)  
library(gridExtra)  
  
dat <- read\_csv('Example\_data/Ch8\_afp.csv')

### 8.3.1 데이터 구조 파악

dim(dat)

colnames(dat)

dat %>%   
 count(group)

### 

### 8.3.2 결측값 확인

colSums(is.na(dat))

### 

### 8.3.3 데이터 전처리

dat1 <- dat

dat1$l3\_m12 <- ifelse(dat1$l3\_m12==0, 0.5, dat1$l3\_m12)  
dat1$l3\_m6 <- ifelse(dat1$l3\_m6==0, 0.5, dat1$l3\_m6)  
dat1$l3\_m0 <- ifelse(dat1$l3\_m0==0, 0.5, dat1$l3\_m0)

dat1 <- dat1 %>%   
 mutate(l3\_m12 = ifelse(l3\_m12==0, 0.5, l3\_m12)) %>%   
 mutate(l3\_m6 = ifelse(l3\_m6==0, 0.5, l3\_m6)) %>%   
 mutate(l3\_m0 = ifelse(l3\_m0==0, 0.5, l3\_m0))

dat1 <- dat1 %>%   
 mutate\_at(vars(l3\_m12, l3\_m6, l3\_m0), ~ifelse(.==0,0.5,.))

library(lubridate)  
  
class(dat1$dob)

class(dat1$m0\_date)

dat1 <- dat1 %>%   
 mutate(age = (m0\_date - dob)/365)   
head(dat1$age)

class(dat1$age)

dat1$age <- as.numeric(dat1$age)  
class(dat1$age)

colnames(dat1)

### 8.3.4 Table 1 만들기

mytable(group~sex+age+etio+plt+alb+alt+bili+  
 inr+bmi+dm+fhx,  
 data=dat1)

dat1 %>%   
 select(group, sex, age, etiology=etio, plt:fhx) %>%   
 tbl\_summary(by=group) %>%   
 add\_p() %>%   
 modify\_caption('\*\*Table 1. Baseline Characteristics\*\*')

### 

### 8.3.5 Biomarker figure 만들기

gghistogram(dat1, x='afp\_m0', bins=20, color='red',   
 add='median', rug=TRUE)  
gghistogram(dat1, x='afp\_m6', bins=20, color='blue',  
 add='median', rug=TRUE)  
gghistogram(dat1, x='afp\_m12', bins=20,   
 add='median', rug=TRUE)

gghistogram(dat1, x='piv\_m0', bins=20, color='red',   
 add='median', rug=TRUE)  
gghistogram(dat1, x='piv\_m6', bins=20, color='blue',  
 add='median', rug=TRUE)  
gghistogram(dat1, x='piv\_m12', bins=20,   
 add='median', rug=TRUE)

gghistogram(dat1, x='l3\_m0', bins=20, color='red',   
 add='median', rug=TRUE)  
gghistogram(dat1, x='l3\_m6', bins=20, color='blue',  
 add='median', rug=TRUE)  
gghistogram(dat1, x='l3\_m12', bins=20,   
 add='median', rug=TRUE)

mytable(group~afp\_m12+afp\_m6+afp\_m0+  
 piv\_m12+piv\_m6+piv\_m0+  
 l3\_m12+l3\_m6+l3\_m0,  
 data=dat1, method=2)

mytheme<-theme\_bw()+  
 theme(panel.grid.major=element\_blank(),  
 panel.grid.minor=element\_blank(),  
 axis.title.x=element\_text(size=16),  
 axis.text.x=element\_text(size=12),  
 axis.title.y=element\_text(size=16),  
 axis.text.y=element\_text(size=12),  
 plot.title=element\_text(size=18, face="bold"),  
 legend.position="none")

afp<-dat1 %>%   
 select(group,contains('afp'))  
  
head(afp)

afp1<-afp %>%   
 gather('interval', 'result', 2:4)   
  
head(afp1)

afp1$interval<-factor(afp1$interval, levels = c('afp\_m12','afp\_m6','afp\_m0'))

afp.boxplot<-ggplot(afp1, aes (x=interval, y=result))+  
 geom\_boxplot()+  
 scale\_y\_continuous(trans = 'log10', breaks=c(1,10,20,50,100),name="AFP (ng/mL)")+  
 mytheme+  
 facet\_grid(.~group, labeller = label\_parsed)  
afp.boxplot

piv<-dat1 %>%   
 select(group,contains('piv')) %>%   
 gather('interval', 'result', 2:4) %>%   
 mutate(interval = factor(interval,   
 levels=c('piv\_m12','piv\_m6','piv\_m0')))

piv.boxplot<-ggplot(piv, aes (x=interval, y=result))+  
 geom\_boxplot()+  
 scale\_y\_continuous(trans = 'log10', breaks=c(1,10,20,50,100),name="piv (ng/mL)")+  
 mytheme+  
 facet\_grid(.~group, labeller = label\_parsed)  
piv.boxplot

l3.boxplot<-dat1 %>%   
 select(group,contains('l3')) %>%   
 gather('interval', 'result', 2:4) %>%   
 mutate(interval = factor(interval,   
 levels = c('l3\_m12','l3\_m6','l3\_m0'))) %>%   
 ggplot(aes (x=interval, y=result))+  
 geom\_boxplot()+  
 scale\_y\_continuous(breaks=c(1,10,20,50,100),name="l3 (ng/mL)")+  
 mytheme+  
 facet\_grid(.~group, labeller = label\_parsed)  
  
l3.boxplot

grid.arrange(afp.boxplot, piv.boxplot, l3.boxplot, nrow=1, ncol=3)

## 

## 8.4 ROC curve

### 8.4.1 개별 ROC curve 그리기

library(pROC)  
  
afp\_month0<-roc(form=group~afp\_m0, data=dat1, ci=T)  
afp\_month0

plot(afp\_month0)

piv\_month0<-roc(form=group~piv\_m0, data=dat1, ci=T)  
piv\_month0

plot(piv\_month0)

l3\_month0<-roc(form=group~l3\_m0, data=dat1, ci=T)  
l3\_month0

plot(l3\_month0)

### 

### 8.4.2 ROC curve 겹쳐 그리기

plot(afp\_month0)  
plot(piv\_month0, add=TRUE, col='blue')  
plot(l3\_month0, add=TRUE, col='red')

## 

## 8.5. Cut-off

### 8.5.1 Optimal cut-off

ci.thresholds(afp\_month0, conf.level=0.95, boot.n=1000, thresholds='best')

ci.thresholds(piv\_month0, conf.level=0.95, boot.n=1000, thresholds='best')

ci.thresholds(l3\_month0, conf.level=0.95, boot.n=1000, thresholds='best')

### 

### 8.5.2 특정 cut-off에서 metrics구하기

metric<-c('sensitivity','specificity','ppv','npv')  
a1<-ci.coords(afp\_month0, x=4.65, input="threshold", ret=metric)  
a1

a1$sensitivity

a1$specificity

a1$ppv

a1$npv

a2<-ci.coords(afp\_month0, x=7, input="threshold", ret=metric)  
a2

a1$sensitivity

a2$sensitivity

sens.table<-rbind(a1$sensitivity, a2$sensitivity)  
rownames(sens.table)<-c('cutoff\_4.65','cutoff\_7')  
sens.table

metric2<-c("specificity", "sensitivity", "accuracy", "tn", "tp", "fn", "fp", "npv","ppv", "1-specificity", "1-sensitivity", "1-accuracy", "1-npv", "1-ppv")  
ci.coords(afp\_month0, x=4.65, input="threshold", ret=metric2)

### 

### 8.5.3 특정 민감도에서 특이도 구하기

a3<-ci.coords(afp\_month0, x=0.8, input="sensitivity", ret=metric2)  
a3$sensitivity

a3$specificity

a4<-ci.coords(afp\_month0, x=0.9, input="sensitivity", ret=metric2)

spec.table<-rbind(a3$specificity, a4$specificity)  
rownames(spec.table)<-c('sensitivity\_80%','sensitivity\_90%')  
spec.table

### 

### 8.5.4 ROC curve 비교

roc.test(afp\_month0, piv\_month0)

### 

### 8.5.5 여러개 변수 ROC그리기

library(Epi)  
  
temp<-dat1 %>%   
 select(group, afp\_m0, piv\_m0, l3\_m0) %>%   
 mutate(gr=ifelse(group=='case',1,0))  
  
head(temp)

afp\_piv<-ROC(form=gr~afp\_m0+piv\_m0, data=temp, plot = 'ROC')

afp\_piv$AUC

l<-glm(gr~afp\_m0+piv\_m0+l3\_m0,data=temp, family=binomial)  
prob=predict(l,type=c("response"))  
temp$prob=prob  
g<-roc(gr~prob,data=temp)  
head(temp)

plot.roc(dat1$group, dat1$afp\_m0, ci=F, of="thresholds", identity.col="grey", main="HCC cases vs. Controls")  
plot.roc(dat1$group, dat1$piv\_m0, ci=F, of="thresholds", add=T, col="orange", lty=2)  
plot.roc(dat1$group, dat1$l3\_m0, ci=F, of="thresholds", add=T, col="blue")  
plot(g,add=T,col="red")

### 

### 8.5.6 자동 계산 table 만들기

a<-seq(0.5,1,0.1)   
a

cal\_specificity<-function(x, y){  
 spec<-ci.coords(y, x=x, input="sensitivity", ret='specificity')  
 spec$specificity  
}  
  
  
table.afp<-cal\_specificity(a, afp\_month0)  
  
rownames(table.afp)<-paste0('Sensitivity',' ',seq(0.5,1,0.1)\*100, '%')  
table.afp

table.piv<-cal\_specificity(a, piv\_month0)  
rownames(table.piv)<-paste0('Sensitivity',' ',seq(0.5,1,0.1)\*100, '%')  
table.piv

table.l3<-cal\_specificity(a, l3\_month0)  
rownames(table.l3)<-paste0('Sensitivity',' ',seq(0.5,1,0.1)\*100, '%')  
table.l3

# Chapter 9. 연구 따라하기 (3) ALT 정상화와 간암 발생 위험

## 

## 9.3 데이터 전처리 및 탐색

### 9.3.1 데이터 불러오기, 결측값 확인

library(tidyverse)  
library(moonBook)  
library(gtsummary)  
library(survival)  
library(survminer)  
library(lubridate)

dat <- read\_csv('Example\_data/Ch9\_alt.csv')

dim(dat)

colnames(dat)

dat1<-dat

colSums(is.na(dat1))

dat1 %>%   
 count(fatty)

dat1$fatty[is.na(dat1$fatty)]<-c('not\_available')  
dat1 %>%   
 count(fatty)

### 

### 9.3.2 관찰 기간 계산

dat1$age<-(dat1$index\_date-dat1$dob)/365.25  
dat1$age<-as.numeric(dat1$age)  
summary(dat1$age)

dat1$index\_date <- ymd(dat1$index\_date)  
dat1$dob <-ymd(dat1$dob)

dat1$hcc\_yr<-(dat1$hcc\_date-dat1$index\_date)/365.25  
dat1$hcc\_yr<-as.numeric(dat1$hcc\_yr)  
summary(dat1$hcc\_yr)

### 9.3.3 데이터 전처리

dat1$dna\_log<-log10(dat1$dna)  
  
summary(dat1$dna\_log)

## 

## 9.4 환자 기저 특성

### 9.4.1 Table 1 만들기

mytable(~age+sex+plt+inr+alt+bil+alb+cr+fatty+lc+hbeag+dna\_log+dm,  
 data=dat1)

## 

## 9.5 간암 발생률

### 9.5.1 Kaplan-Meier 그리기

fit.hcc<-survfit(Surv(hcc\_yr,hcc)~1, data=dat1)  
  
ggsurvplot(fit.hcc,  
 fun='event',  
 ylim=c(0,0.3),  
 break.time.by=2,  
 risk.table = TRUE,  
 censor = FALSE)

### 

### 9.5.2 연간 간암 발생률 구하기

dat1 %>%   
 summarise(patient\_number = n(),  
 n\_hcc = sum(hcc),  
 person\_year = sum(hcc\_yr),  
 incidence\_rate = n\_hcc / person\_year)

### 

### 9.5.3 누적 간암 발생률 구하기

fit.hcc<-survfit(Surv(hcc\_yr,hcc)~1, data=dat1)  
  
survest<-stepfun(fit.hcc$time, c(1,fit.hcc$surv))  
y2<-1-survest(2)  
y3<-1-survest(3)  
y4<-1-survest(4)  
y5<-1-survest(5)  
y10<-1-survest(10)  
year<-c("2 Year","3 Year","4 Year", "5 Year","10 Year")  
rate<-c(y2,y3,y4,y5,y10)  
  
cum\_hcc\_rate<-data.frame(time=year,rate=rate\*100)  
cum\_hcc\_rate

## 

## 9.6 ALT 정상화

### 9.6.1 ALT 정상화 KM 그리기

fit.alt<-survfit(Surv(alt\_duration, alt\_nl=='normal')~1, data=dat1)  
  
ggsurvplot(fit.alt,  
 fun='event',  
 conf.int = FALSE,  
 xlim=c(0,5),  
 ylim=c(0,1),  
 break.time.by=1,  
 risk.table = TRUE,  
 censor = FALSE)

### 

### 9.6.2 누적 ALT 정상화율 구하기

fit.alt<-survfit(Surv(alt\_duration, alt\_nl=='normal')~1, data=dat1)  
  
survest<-stepfun(fit.alt$time, c(1,fit.alt$surv))  
y1<-round(1-(survest(1)),3)  
y2<-round(1-(survest(2)),3)  
y3<-round(1-(survest(3)),3)  
y4<-round(1-(survest(4)),3)  
y5<-round(1-(survest(5)),3)  
year<-c("1 Year","2 Year","3 Year","4 Year","5 Year")  
rate<-c(y1,y2,y3,y4,y5)  
cum\_ALT\_rate<-data.frame(time=year,rate=rate\*100)  
cum\_ALT\_rate

### 

### 9.6.4 ALT 정상화와 간암, landmark 분석

dat1 <- dat1 %>%   
 mutate(yr1.alt\_nl = ifelse(alt\_nl=='normal' & alt\_duration<=1, 'normal', 'abnormal'))

dat1 %>%   
 count(alt\_nl, yr1.alt\_nl)

fit.alt.1yr<-survfit(Surv(hcc\_yr, hcc)~yr1.alt\_nl, data=dat1)  
  
ggsurvplot(fit.alt.1yr,  
 fun='event',  
 conf.int = FALSE,  
 xlim=c(0,13),  
 ylim=c(0,0.5),  
 break.time.by=2,  
 risk.table = TRUE,  
 censor = FALSE,  
 pval = TRUE)

survdiff(Surv(hcc\_yr, hcc)~yr1.alt\_nl, data=dat1)

dat.land <- dat1 %>%   
 filter(hcc\_yr>=2) %>%   
 mutate(yr2.alt\_nl = ifelse(alt\_nl=='normal' & alt\_duration<=2, 'normal', 'abnormal'))   
  
dat.land %>%   
 count(alt\_nl, yr2.alt\_nl)

fit.alt.2yr<-survfit(Surv(hcc\_yr, hcc)~yr2.alt\_nl, data=dat.land)  
  
ggsurvplot(fit.alt.2yr,  
 fun='event',  
 conf.int = FALSE,  
 xlim=c(0,13),  
 ylim=c(0,0.5),  
 break.time.by=2,  
 risk.table = TRUE,  
 censor = FALSE,  
 pval = TRUE)

survdiff(Surv(hcc\_yr, hcc)~yr2.alt\_nl, data=dat.land)

### 

### 9.6.5 ALT 정상화 시점과 간암 발생

summary(dat1$alt\_duration)

dat1$alt\_nl\_cate<-ifelse(dat1$alt\_duration<=0.5, "<6\_months",  
 ifelse(dat1$alt\_duration<=1, "<12\_months",  
 ifelse(dat1$alt\_duration<=2,"<24\_months","abnormal")))  
  
dat1 %>%   
 count(alt\_nl\_cate)

dat1$alt\_nl\_cate<-factor(dat1$alt\_nl\_cate,  
 levels = c("<6\_months","<12\_months","<24\_months","abnormal"))  
  
dat1 %>%  
 count(alt\_nl\_cate)

fit.alt.gr<-survfit(Surv(hcc\_yr, hcc)~alt\_nl\_cate, data=dat1)  
  
ggsurvplot(fit.alt.gr,  
 fun='event',  
 conf.int = FALSE,  
 xlim=c(0,13),  
 ylim=c(0,0.5),  
 break.time.by=2,  
 risk.table = TRUE,  
 censor = FALSE,  
 pval = TRUE)

survdiff(Surv(hcc\_yr, hcc)~alt\_nl\_cate, data=dat1)

## 

## 9.7 간암 발생 위험인자

### 9.7.1 단변량 분석

cox.uni<-dat1 %>%   
 select(hcc\_yr,hcc, age, sex, plt, inr,  
 alt, bil, alb, cr, fatty, lc, hbeag, dm, dna\_log, alt\_nl\_cate) %>%   
 tbl\_uvregression(method=coxph,  
 y=Surv(hcc\_yr,hcc),  
 exponentiate = TRUE)  
cox.uni

### 9.7.2 다변량 분석

fit<-coxph(Surv(hcc\_yr, hcc==1)~age+sex+plt+inr+alt+bil+alb+cr+fatty+lc+hbeag+dm+dna\_log+alt\_nl\_cate,  
 data=dat1)  
fit.multi<-step(fit, direction = 'backward')  
extractHR(fit.multi)

cox.multi<-coxph(Surv(hcc\_yr, hcc==1)~age+sex+plt+inr+cr+fatty+lc+alt\_nl\_cate,  
 data=dat1) %>%   
 tbl\_regression(exponentiate=TRUE,   
 add\_n=FALSE)  
cox.multi

cox.table<-tbl\_merge(  
 tbls = list(cox.uni, cox.multi),  
 tab\_spanner = c("\*\*Univariate analysis\*\*","\*\*Multivariable analysis\*\*")  
) %>%   
 modify\_caption('\*\*Table. Risk factors for hepatocellular carcinoma develoment\*\*')  
cox.table

cox.table %>%   
 as\_flex\_table() %>%   
 flextable::save\_as\_docx(path='coxtable.docx')

library(forestmodel)  
  
forest\_model(fit.multi)

## 

## 9.8 Subgroup 분석

### 9.8.1 지방간이 없는 환자

nofatty<-dat1 %>%   
 filter(fatty=='No fatty liver')  
  
dim(nofatty)

fit.nofatty<-survfit(Surv(hcc\_yr, hcc)~alt\_nl\_cate, data=nofatty)  
  
ggsurvplot(fit.nofatty,  
 fun='event',  
 conf.int = FALSE,  
 xlim=c(0,13),  
 ylim=c(0,0.6),  
 break.time.by=2,  
 risk.table = TRUE,  
 censor = FALSE,  
 pval = TRUE)

survdiff(Surv(hcc\_yr, hcc)~alt\_nl\_cate, data=nofatty)

cox.uni.nofatty<-nofatty %>%   
 select(hcc\_yr,hcc, age, sex, plt, inr,  
 alt, bil, alb, cr, lc, hbeag, dm, dna\_log, alt\_nl\_cate) %>%   
 tbl\_uvregression(method=coxph,  
 y=Surv(hcc\_yr,hcc),  
 exponentiate = TRUE)  
cox.uni.nofatty

fit.nofatty<-coxph(Surv(hcc\_yr, hcc==1)~age+sex+plt+inr+alt+bil+alb+cr+lc+hbeag+dm+dna\_log+alt\_nl\_cate,  
 data=nofatty)  
fit.multi.nofatty<-step(fit.nofatty, direction = 'backward')  
extractHR(fit.multi.nofatty)

cox.multi.nofatty<-coxph(Surv(hcc\_yr, hcc==1)~age+sex+plt+alb+cr+lc+alt\_nl\_cate,  
 data=nofatty) %>%   
 tbl\_regression(exponentiate=TRUE,   
 add\_n=FALSE) %>%   
 modify\_caption('\*\*Table. Risk factors for hepatocellular carcinoma in patients without fatty liver\*\*')  
cox.multi.nofatty

### 

### 9.8.2 간경변증이 있는 환자

lc<-dat1 %>%   
 filter(lc==1)  
dim(lc)

fit.lc<-survfit(Surv(hcc\_yr, hcc)~alt\_nl\_cate, data=lc)  
  
ggsurvplot(fit.lc,  
 fun='event',  
 conf.int = FALSE,  
 xlim=c(0,13),  
 ylim=c(0,0.6),  
 break.time.by=2,  
 risk.table = TRUE,  
 censor = FALSE,  
 pval = TRUE)

lc %>%   
 select(hcc\_yr,hcc, alt\_nl\_cate) %>%   
 tbl\_uvregression(method=coxph,  
 y=Surv(hcc\_yr,hcc),  
 exponentiate = TRUE)

cox.uni.lc<-lc %>%   
 select(hcc\_yr,hcc, age, sex, plt, inr,  
 alt, bil, alb, cr, fatty, hbeag, dm, dna\_log, alt\_nl\_cate) %>%   
 tbl\_uvregression(method=coxph,  
 y=Surv(hcc\_yr,hcc),  
 exponentiate = TRUE)  
cox.uni.lc

fit.lc<-coxph(Surv(hcc\_yr, hcc==1)~age+sex+plt+inr+alt+bil+alb+cr+fatty+hbeag+dm+dna\_log+alt\_nl\_cate,  
 data=lc)  
fit.multi.lc<-step(fit.lc, direction = 'backward')  
extractHR(fit.multi.lc)

cox.multi.lc<-coxph(Surv(hcc\_yr, hcc==1)~age+sex+plt+inr+fatty+alt\_nl\_cate,  
 data=lc) %>%   
 tbl\_regression(exponentiate=TRUE) %>%   
 modify\_caption('\*\*Table. Risk factors for hepatocellular carcinoma in patients with cirrhosis\*\*')  
cox.multi.lc

### 9.8.3 Subgroup 분석 결과 한개 table로 제시하기

subgroup.table<-tbl\_merge(  
 tbls = list(cox.multi.nofatty, cox.multi.lc),  
 tab\_spanner = c("\*\*No fatty liver\*\*","\*\*Liver cirrhosis\*\*")  
) %>%   
 modify\_caption('\*\*Table. Risk factors for hepatocellular carcinoma in subgroups\*\*')   
subgroup.table