#1. 서론

#데이터 불러오기

```
setwd("C:/Users/DS/Downloads")
data <- read.csv("heart_failure_clinical_records_dataset2.csv")
```

#연속형 변수 범주형으로 변환

```
#정상N, 문제P로 변환
library(survival)
data$Na_group <- as.factor(ifelse(data$serum_sodium >= 135, "N", "P"))
data$EF_Level <- as.factor(ifelse(data$ejection_fraction <= 30, "P", "N"))

summary(data)
```

#2. 본론

#1. 모수적 방법(와이블)

```
#와이블 분포 적합성 확인(by 카플란)
as = survfit(Surv(time, DEATH_EVENT)~1, data=data)
km.time = as$time
km.surv = as$surv

#로그 변환 생존 함수 계산
st.log  = -log(km.surv)
st.loglog = c(log(-log(km.surv)), -0.7)
time.log=log(km.time)

#로그-로그 생존 함수 그림
plot(stepfun(time.log, st.loglog), xlab = "log Time", ylab = "log-log S(t)",
    main = "log-logS(t)", xlim = c(1,6), ylim = c(-6, -0.5))


#와이블분포 알파 추정
library(MASS)
fit_result <- fitdistr(data$time, "weibull")
print(fit_result$estimate["shape"])
alpha <- fit_result$estimate["shape"] #알파 1.623731


# Weibull 분포(공변량x)
library(survival)
wfit <- survreg(Surv(time, DEATH_EVENT) ~ 1, data = data, dist = "weibull")
```

```
summary(wfit)
lambda <- exp(wfit$icoef["Log(scale)"])  # 람다 = exp(0.1824)


# 와이블 분포 생존함수 계산
alpha <- fit_result$estimate["shape"] #알파 1.623731
lambda <- exp(wfit$icoef["Log(scale)"])  # scale

time_seq <- seq(0, 100, length.out = 100)
surv_func <- exp(- (time_seq / lambda) ^ alpha)

plot(time_seq, surv_func, type = "l",
     xlab = "Days", ylab = "Survival Probability",
     main = "Weibull Survival Function", col = "black", lwd = 2, ylim = c(0, 1))
abline(v = 3.75, col = "red", lwd = 2, lty = 2)



#2. 비모수적 방법(카플란, 넬슨) 공변량x

#Kaplan-Meier Method (공변량X)

# Kaplan-Meier 생존 함수
kfit <- survfit(Surv(time, DEATH_EVENT) ~ 1, data = data, type = 'kaplan-meier')
summary(kfit)

# 신뢰구간 상한 하한
lower_ci <- kfit$lower
upper_ci <- kfit$upper
time_seq <- kfit$time

# Kaplan-Meier 생존 함수
plot(kfit, mark.time = FALSE, xlab = "Days", ylab = "Survival",
     main = "Survival Function of Kaplan-Meier", col = "blue", lwd = 2, conf.int = FALSE)

# 신뢰구간 색칠
polygon(c(time_seq, rev(time_seq)),
        c(lower_ci, rev(upper_ci)),
        col = rgb(0, 0, 1, 0.05),
        border = NA)

# 상한과 하한 선
lines(kfit$time, kfit$upper, col = "blue", lty = 2, lwd = 1)
lines(kfit$time, kfit$lower, col = "blue", lty = 2, lwd = 1)


#Kaplan-Meier 방법에 의한 위험률함수 그래프
h.u=-log(kfit$upper)
```

```
h.l=-log(kfit$lower)
plot(kfit$time, -log(kfit$surv),type="l",xlab="days",ylab="hazard",
    main="Hazard Function of Kaplan-Meier", lwd=2, col="red", ylim=c(0,0.8))
lines(kfit$time, h.u, lty=2, col="red")
lines(kfit$time, h.l, lty=2, col="red")

polygon(c(kfit$time, rev(kfit$time)),
        c(h.l, rev(h.u)),
        col = rgb(1, 0, 0, 0.05),
        border = NA)


# Nelson-Aalen 생존 함수 추정
afit <- survfit(Surv(time, DEATH_EVENT) ~ 1, data = data, type = "fh")
summary(afit)

# 신뢰구간 상한과 하한
lower_nel <- afit$lower
upper_nel <- afit$upper
time_seq <- afit$time

plot(afit, mark.time = FALSE, xlab = "Days", ylab = "Survival", ylim=c(0,1),
    main = "Survival Function of Nelson-Aalen", col = "blue", lwd = 2, conf.int = FALSE)

# 신뢰구간 색칠
polygon(c(time_seq, rev(time_seq)),
        c(lower_nel, rev(upper_nel)),
        col = rgb(0, 0, 1, 0.05),
        border = NA)

# 상한과 하한 선
lines(afit$time, afit$upper, col = "blue", lty = 2, lwd = 1)
lines(afit$time, afit$lower, col = "blue", lty = 2, lwd = 1)

## Nelson-Aalen of Hazard Function
h.u=-log(afit$upper)
h.l=-log(afit$lower)
plot(afit$time, -log(afit$surv),type="l",xlab="Days",ylab="hazard",
    main="Hazard Function of Nelson-Aalen", lwd=2, col="red", ylim=c(0,0.8))
lines(afit$time, h.u, lty=2, col="red")
lines(afit$time, h.l, lty=2, col="red")

polygon(c(afit$time, rev(afit$time)),
        c(h.l, rev(h.u)),
        col = rgb(1, 0, 0, 0.05),
        border = NA)

#넬슨 알렌-카플란 두 모델 비교
```

```
kfit <- survfit(Surv(time, DEATH_EVENT) ~ 1, data = data, type = 'kaplan-meier')
afit <- survfit(Surv(time, DEATH_EVENT) ~ 1, data = data, type = "fh")

#생존함수 비교
plot(kfit, mark.time = FALSE, col = "blue", lwd = 1,
    xlab = "Days", ylab = "Survival Probability",
    main = "Survival of Kaplan-Meier and Nelson-Aalen", conf.int = FALSE, ylim=c(0.5,1))

lines(afit$time, afit$surv, col = "red", lwd = 1, lty=2)
legend("topright", legend = c("Kaplan-Meier", "Nelson-Aalen"),
       col = c("blue", "red"), lwd = 1, lty=c(1,2))

#누적위험함수 비교
plot(kfit$time, -log(kfit$surv), type="l", col="blue", lwd=1,
    xlab="Days", ylab="Cumulative Hazard",
    main="Cumulative Hazard of Kaplan-Meier and Nelson-Aalen",
    ylim=c(0, max(-log(kfit$surv), -log(afit$surv))))

lines(afit$time, -log(afit$surv), col="red", lwd=1, lty=2)

legend("topleft", legend = c("Kaplan-Meier", "Nelson-Aalen"),
       col = c("blue", "red"), lwd = 1, lty=c(1,2))


# 기간별 생존함수 차이
op <- par(mfrow = c(2, 2))
p1 <- plot(kfit, mark.time = FALSE, xlab = "Days", ylab = "Survival", xlim = c(0, 70), ylim =
c(0.8, 1))
p2 <- plot(kfit, mark.time = FALSE, xlab = "Days", ylab = "Survival", xlim = c(70, 140), ylim =
c(0.6, 0.9))
p3 <- plot(kfit, mark.time = FALSE, xlab = "Days", ylab = "Survival", xlim = c(140, 210), ylim =
c(0.5, 0.9))
p4 <- plot(kfit, mark.time = FALSE, xlab = "Days", ylab = "Survival", xlim = c(210, 285), ylim =
c(0.45, 0.75))
par(op)



#3.로그순위검정

#3.1 연속형 범주형으로 코딩
data$SCR_group <- as.factor(ifelse(data$serum_creatinine <= 1.5,  "N", "P"))


data$age_group <- ifelse(data$age < median(data$age, na.rm = TRUE), "young", "old")
```

```
data$CPK_group <- ifelse(data$creatinine_phosphokinase <= 250, "Normal",
                         ifelse(data$creatinine_phosphokinase < 1000, "Stress",
                                "Severe"))
table(data$CPK_group) #Normal 151 | Stress 112 | Severe 36
Q1 <- quantile(data$platelets, 0.25, na.rm = TRUE)
Q3 <- quantile(data$platelets, 0.75, na.rm = TRUE)
data$PLT_level <- ifelse(data$platelets <= Q1, 1,
                         ifelse(data$platelets <= Q3, 2, 3))
data$PLT_level <- as.factor(data$PLT_level)
table(data$PLT_level) #1 혈소판 적음 75 | 2 적당 149 | 3 혈소판 많음 75
```

#3.2 범주형

```
library(survival)
fit <- survdiff(Surv(time, DEATH_EVENT) ~ anaemia, data=data)
fit #Chisq= 2.7  on 1 degrees of freedom, p= 0.1

fit2 <- survdiff(Surv(time, DEATH_EVENT) ~ diabetes, data=data)
fit2 #Chisq= 0  on 1 degrees of freedom, p= 0.8

fit3 <- survdiff(Surv(time, DEATH_EVENT) ~ high_blood_pressure, data=data)
fit3 #Chisq= 4.4  on 1 degrees of freedom, p= 0.04

fit4 <- survdiff(Surv(time, DEATH_EVENT) ~ sex, data=data)
fit4 #Chisq= 0  on 1 degrees of freedom, p= 0.9

fit5 <- survdiff(Surv(time, DEATH_EVENT) ~ smoking, data=data)
```

```
fit5 #Chisq= 0  on 1 degrees of freedom, p= 1


fit6 <- survdiff(Surv(time, DEATH_EVENT) ~ EF_Level, data=data)

fit6 #Chisq= 30.6  on 1 degrees of freedom, p= 3e-08


fit7 <- survdiff(Surv(time, DEATH_EVENT) ~ Na_group, data=data)

fit7 #Chisq= 15.6  on 1 degrees of freedom, p= 8e-05


#3.3 연속형

fit1 <- survdiff(Surv(time, DEATH_EVENT) ~ age_group, data=data)

fit1 #Chisq= 8.6  on 1 degrees of freedom, p= 0.003


fit2 <- survdiff(Surv(time, DEATH_EVENT) ~ CPK_group, data=data)

fit2 #Chisq= 0.9  on 2 degrees of freedom, p= 0.6


fit3 <- survdiff(Surv(time, DEATH_EVENT) ~ PLT_level, data=data)

fit3 #Chisq= 3.5  on 2 degrees of freedom, p= 0.2


fit4 <- survdiff(Surv(time, DEATH_EVENT) ~ SCR_group, data=data)

fit4 #Chisq= 39.6  on 1 degrees of freedom, p= 3e-10


#3.4 조합별 카플란 그림

op = par(mfrow=c(1,3))

afit <- survfit(Surv(time, DEATH_EVENT) ~ anaemia+high_blood_pressure, data = data,
type = 'kaplan-meier')

col_vector <- rainbow(length(afit$strata))

plot(afit, mark.time = FALSE, xlab = "Days", ylab = "Survival", ylim = c(0.3, 1),
```

```
    col = col_vector, main = "Anaemia & High blood pressure")

legend("bottomleft", legend = names(afit$strata),

       col = col_vector, lty = 1, cex = 0.8)


afit2 <- survfit(Surv(time, DEATH_EVENT) ~ SCR_group + Na_group, data = data, type =
'kaplan-meier')

col_vector <- rainbow(length(afit2$strata))

plot(afit2, mark.time = FALSE, xlab = "Days", ylab = "Survival", ylim = c(0.3, 1),

    col = col_vector, main = "Serum_creatinine & Na")

legend("bottomleft", legend = names(afit2$strata),

       col = col_vector, lty = 1, cex = 0.8)




afit3 <- survfit(Surv(time, DEATH_EVENT) ~ age_group + EF_Level, data = data, type =
'kaplan-meier')

col_vector <- rainbow(length(afit3$strata))

plot(afit3, mark.time = FALSE, xlab = "Days", ylab = "Survival", ylim = c(0.3, 1),

    col = col_vector, main = "Age & EF Level")

legend("bottomleft", legend = names(afit3$strata) ,

       col = col_vector, lty = 1, cex = 0.8)




#4. Cox비례위험모형(유의수준 0.05기준)

##전체모형(full)

library(survival)

cfit <- coxph(Surv(time,DEATH_EVENT)~age + anaemia + creatinine_phosphokinase +
diabetes + high_blood_pressure + platelets + sex

            + smoking + serum_creatinine + Na_group + EF_Level ,data=data)
```

```
summary(cfit)
```

축소모형(Subset)

```
#나이, 심박출량, 크레아티닌, CPK효소, 고혈압(유의수준 0.05기준)

cfit1=coxph(Surv(time,DEATH_EVENT)~age + EF_Level + serum_creatinine +
creatinine_phosphokinase + high_blood_pressure,data=data)

summary(cfit1)
```

축소모형(Subset) , 단계적 변수 선택법

```
# 단계적 선택법 (Stepwise Selection)

cfit <- coxph(Surv(time,DEATH_EVENT)~age + anaemia + creatinine_phosphokinase +
diabetes + high_blood_pressure + platelets + sex

            + smoking + serum_creatinine + Na_group + EF_Level ,data=data)


cox_model_empty <- coxph(Surv(time, DEATH_EVENT) ~ 1,   data = data)

cox_model_full <-  cfit


cox_model_stepwise <- step(cox_model_empty, scope = list(lower = cox_model_empty,
upper = cox_model_full ),   direction = "both")

summary(cox_model_stepwise)

#serum_creatinine + age + EF_Level + high_blood_pressure + creatinine_phosphokinase +
Na_group
```

비례 위험률모형 생존함수 그래프

```
cox.fit <- survfit(cfit) #full

cox.fit1 <- survfit(cfit1) #Subset

cox.fit2 <- survfit(cox_model_stepwise) #cox_model_stepwise
```

```
plot(cox.fit, ylim = c(0.6, 1), xlab = "Days", ylab = "Survival Probability", main = "Cox
Proportional Survival Model")

par(new = TRUE)

plot(cox.fit1, ylim = c(0.6, 1), col = 2, xlab = "Days", ylab = "", main = "")

par(new = TRUE)

plot(cox.fit2, ylim = c(0.6, 1), col = 3, xlab = "Days", ylab = "", main = "")

par(new = TRUE)

legend("bottomleft", c("Full model","pvalue(age + creatinine_phosphokinase +
high_blood_pressure + serum_creatinine  + EF_Level)","stepwise(age +
creatinine_phosphokinase + high_blood_pressure + serum_creatinine + Na_group +
EF_Level)"),

      col = c(1,2,3),  text.col = "black",lty=c(1), bg = 'white',cex = 0.9)
```

#5. 모형 비교

심박출량 기준으로 비교

```
## cox model: EF_level = P, EF_level = N
EFdata1 = data[data$EF_Level=="P", ] #심장기능 이상
effit1=coxph(Surv(time,DEATH_EVENT)~age+serum_creatinine+high_blood_pressure+crea
tinine_phosphokinase+Na_group,data=EFdata1)
summary(effit1)

EFdata2 = data[data$EF_Level=="N", ] #심장기능 정상
effit2=coxph(Surv(time,DEATH_EVENT)~age+serum_creatinine+high_blood_pressure+crea
tinine_phosphokinase+Na_group,data=EFdata2)
summary(effit2)

cox.effit1=survfit(effit1)
cox.effit2=survfit(effit2)

##생존함수
plot(cox.effit1$time, cox.effit1$surv, lwd=2, type = "l", xlab = "Days", ylab = "survival", ylim =
c(0, 1), xlim = c(0, 300), col="red")
title("EF Cox proportional Survival model")
par(new=T)
plot(cox.effit2$time, cox.effit2$surv, lwd=2, type = "l", xlab = "Days", ylab = "survival", ylim =
c(0, 1), xlim = c(0, 300), col="blue")
```

```
legend("bottomleft", c("EF_Level Problem_model","EF_Level Normal_model"), col =
c("red","blue"),
      text.col = "black",lty=c(1), lwd=2, bg = 'white')

#심장 기능 이상 신뢰구간
lower_ef1 <- cox.effit1$lower
upper_ef1 <- cox.effit1$upper
time_seq1 <- cox.effit1$time

polygon(c(time_seq1, rev(time_seq1)),
      c(lower_ef1, rev(upper_ef1)),
      col = rgb(1, 0, 0, 0.05),
      border = NA)

# 상한과 하한 선
lines(cox.effit1$time, cox.effit1$upper, col = "red", lty = 2, lwd = 1)
lines(cox.effit1$time, cox.effit1$lower, col = "red", lty = 2, lwd = 1)

#심장 기능 정상 신뢰구간
lower_ef2 <- cox.effit2$lower
upper_ef2 <- cox.effit2$upper
time_seq2 <- cox.effit2$time

polygon(c(time_seq2, rev(time_seq2)),
      c(lower_ef2, rev(upper_ef2)),
      col = rgb(0, 0, 1, 0.05),
      border = NA)

# 상한과 하한 선
lines(cox.effit2$time, cox.effit2$upper, col = "blue", lty = 2, lwd = 1)
lines(cox.effit2$time, cox.effit2$lower, col = "blue", lty = 2, lwd = 1)



##생존 함수 비교

kfit=survfit(Surv(time, DEATH_EVENT) ~1, data=data, type='kaplan-meier')

afit=survfit(Surv(time, DEATH_EVENT)~1, data=data, type="fleming-harrington")



cfit <- coxph(Surv(time,DEATH_EVENT)~age + anaemia + creatinine_phosphokinase +
diabetes + high_blood_pressure + platelets + sex

          + smoking + serum_creatinine + Na_group + EF_Level ,data=data)
```

```
cfit1=coxph(Surv(time,DEATH_EVENT)~age + EF_Level + serum_creatinine +
creatinine_phosphokinase + high_blood_pressure,data=data)



cox_model_empty <- coxph(Surv(time, DEATH_EVENT) ~ 1,   data = data)

cox_model_full <-  cfit



cox_model_stepwise <- step(cox_model_empty, scope = list(lower = cox_model_empty,
upper = cox_model_full ),   direction = "both")



cox.fit <- survfit(cfit) #full

cox.fit1 <- survfit(cfit1) #Subset

cox.fit2 <- survfit(cox_model_stepwise) #cox_model_stepwise

cox.fit3 <- survfit(coxph(Surv(time, DEATH_EVENT) ~ 1, data = data)) #con_null



op = par(mfrow=c(1,2))

plot(kfit, mark.time = FALSE, xlab = "Days", ylab = "Survival",col = 1, conf.int = FALSE, xlim
= c(0.45, 300), ylim = c(0.45, 1))

par(new = T)

plot(afit$time, afit$surv, type = "l", col = "blue", xlab = "", ylab = "", ylim = c(0.45, 1), xlim =
c(0.45, 300))  ## Fleming

par(new = T)

plot(cox.fit$time, cox.fit$surv, type = "l", col = 6, xlab = "", ylab = "survival", ylim = c(0.45, 1),
xlim = c(0, 300)) ## Cox full

par(new = T)

plot(cox.fit1$time, cox.fit1$surv, type = "l", col = 3, xlab = "", ylab = "survival", ylim = c(0.45,
1), xlim = c(0, 300))  ## cox_subset

par(new = T)

plot(cox.fit2$time, cox.fit2$surv, type = "l", col = "red", xlab = "", ylab = "survival", ylim =
c(0.45, 1), xlim = c(0, 300), lwd=2) ## cox_model_stepwise
```

```
par(new = T)

plot(cox.fit3$time, cox.fit3$surv, type = "l", col = 5, xlab = "", ylab = "survival", ylim = c(0.45, 1), xlim = c(0, 300)) ## COX NULL



title("Survival Function")

legend("bottomleft", c("Kaplan-Meier", "Nelson-Aalen", "Cox-full", "cox_model_pvalue", "cox_model_stepwise","cox_null"), col = c(1,"blue",6,3,"red",5),

       text.col = "black", lty = c(1), bg = 'white',cex = 0.9)




##위험률함수 비교(Hazard)

plot(kfit$time, -log(kfit$surv),type="l",xlab="",ylab="",xlim=c(0,300),ylim=c(0,0.65))

par(new=T)

plot(afit$time, -log(afit$surv),type="l",col="blue",xlab="",ylab="",xlim=c(0,300),ylim=c(0,0.65))

par(new=T)

plot(cox.fit$time,-log(cox.fit$surv),type="l",col=6,xlab="Days",ylab="Hazard",xlim=c(0,300),ylim=c(0,0.65))  #full

par(new=T)

plot(cox.fit1$time,-log(cox.fit1$surv),type="l",col=3,xlim=c(0,300),ylim=c(0,0.65),xlab="",ylab="")

par(new=T)

plot(cox.fit2$time,-log(cox.fit2$surv),type="l",col="red",xlim=c(0,300),ylim=c(0,0.65),xlab="",ylab="", lwd=2)

par(new=T)

plot(cox.fit3$time,-log(cox.fit3$surv),type="l",col=5,xlim=c(0,300),ylim=c(0,0.65),xlab="",ylab="")



title("Hazard Function")
```

```
legend("bottomright", c("Kaplan-Meier", "Nelson-Aalen", "Cox-full", "cox_model_pvalue",
"cox_model_stepwise","cox_null"), col = c(1,"blue",6,3,"red",5),

      text.col = "black",lty=c(1) , bg = 'white',cex = 0.8)

#6. 최종 Cox비례모형 평가

#6.1 연속형 잔차분석

#공변량 없는 경우

library(survival)

fit0 <- coxph(Surv(time, DEATH_EVENT)~1, data=data)

summary(fit0)

r1 <- resid(fit0, data=data, type="martingale")

r2 <- resid(fit0, data=data, type="deviance")


op = par(mfrow=c(1,2))

plot(data$age, r1, xlab="age", ylab="martingale residual")

lines(lowess(data$age, r1, iter=0), lty=2)

plot(data$age, r2, xlab="age", ylab="deviance residual")

lines(lowess(data$age, r2, iter=0), lty=2)


#creatinine_phosphokinase

op = par(mfrow=c(2,2))

plot(data$creatinine_phosphokinase, r1, xlab="creatinine_phosphokinase", ylab="martingale
residual")

lines(lowess(data$creatinine_phosphokinase, r1, iter=0), lty=2)#비선형


plot(data$creatinine_phosphokinase, r2, xlab="creatinine_phosphokinase", ylab="deviance
residual")

lines(lowess(data$creatinine_phosphokinase, r2, iter=0), lty=2)
```

```
plot(data$creatinine_phosphokinase, r1, xlab="creatinine_phosphokinase", ylab="martingale residual", xlim=c(0,2000))

lines(lowess(data$creatinine_phosphokinase, r1, iter=0), lty=2)


plot(data$creatinine_phosphokinase, r2, xlab="creatinine_phosphokinase", ylab="deviance residual", xlim=c(0,2000))

lines(lowess(data$creatinine_phosphokinase, r2, iter=0), lty=2)


#serum_creatinine

op = par(mfrow=c(1,2))

plot(data$serum_creatinine, r1, xlab="serum_creatinine", ylab="martingale residual")

lines(lowess(data$serum_creatinine, r1, iter=0), lty=2)

plot(data$serum_creatinine, r2, xlab="serum_creatinine", ylab="deviance residual")

lines(lowess(data$serum_creatinine, r2, iter=0), lty=2)


# 공변량 있는 경우

##6개 모형

fita <- coxph(Surv(time, DEATH_EVENT) ~ age + high_blood_pressure + Na_group +

              creatinine_phosphokinase + EF_Level + serum_creatinine, data = data)

summary(fita)


r1 <- resid(fita, data=data, type="martingale")

r2 <- resid(fita, data=data, type="deviance")

schoenfeld_residuals <- cox.zph(fita)

r4 <- resid(fita, data=data, type="partial")


age_partial <- r4[, "age"]
```

```
cpk_partial <- r4[, "creatinine_phosphokinase"]

sc_partial <- r4[, "serum_creatinine"]


#age

op=par(mfrow=c(2,2))

plot(data$age, r1, xlab="age", ylab="martingale", main="Martingale for Age")

lines(lowess(data$age, r1, iter=0), lty=2)


plot(data$age, r2, xlab="age", ylab="deviance", main="Deviance for Age")

lines(lowess(data$age, r2, iter=0), lty=2)


plot(schoenfeld_residuals, var = "age", main = "Schoenfeld for Age")


plot(data$age, age_partial, xlab="age", ylab="partial", main="Partial for Age")

lines(lowess(data$age, age_partial, iter=0), lty=2)


#CPK

op=par(mfrow=c(2,2))

plot(data$creatinine_phosphokinase, r1, xlab="creatinine_phosphokinase",
ylab="martingale", main="Martingale for CPK")

lines(lowess(data$creatinine_phosphokinase, r1, iter=0), lty=2)


plot(data$creatinine_phosphokinase, r2, xlab="creatinine_phosphokinase", ylab="deviance",
main="Deviance for CPK")

lines(lowess(data$creatinine_phosphokinase, r2, iter=0), lty=2)


plot(schoenfeld_residuals, var = "creatinine_phosphokinase", main = "Schoenfeld for CPK")
```

```
plot(data$creatinine_phosphokinase, cpk_partial, xlab="creatinine_phosphokinase",
ylab="partial", main="Partial for CPK")

lines(lowess(data$creatinine_phosphokinase, cpk_partial, iter=0), lty=2)



#크레아티닌

op=par(mfrow=c(2,2))

plot(data$serum_creatinine, r1, xlab="serum_creatinine", ylab="martingale",
main="Martingale for Serum_creatinine")

lines(lowess(data$serum_creatinine, r1, iter=0), lty=2)



plot(data$serum_creatinine, r2, xlab="serum_creatinine", ylab="deviance", main="Deviance
for Serum_creatinine")

lines(lowess(data$serum_creatinine, r2, iter=0), lty=2)



plot(schoenfeld_residuals, var = "serum_creatinine", main = "Schoenfeld for
Serum_creatinine")



plot(data$serum_creatinine, sc_partial, xlab="serum_creatinine", ylab="partial", main =
"Partial for Serum_creatinine")

lines(lowess(data$serum_creatinine, sc_partial, iter=0), lty=2)




#6.2 범주형 비례성 분석

#고혈압 만족

par(mfrow=c(1, 2))

fit1 <- survfit(coxph(Surv(time, DEATH_EVENT)~strata(factor(high_blood_pressure)),

                      data=data, method="breslow"), type="aalen")
```

```
plot(fit1, col=c("black", "red"), lty=c(1,2), xlim=c(0, 300),
     xlab="Time", ylab="Survival Probability")
legend('bottomright', legend=c("high blood pressure No", "high blood pressure Yes"),
       lty=c(1,2), lwd=1.9, col=c("black", "red"), cex=0.9)

plot(fit1, fun = 'cumhaz',
     col=c("black", "red"), lty=c(1,2), xlim=c(0,300),
     xlab="Days", ylab="Log Cumulative Hazard Rate", main="high blood pressure")
legend('bottomright', legend=c("high blood pressure No", "high blood pressure Yes"),
       lty=c(1,2), col=c("black", "red"), cex=0.8)

#EF Level 만족
par(mfrow=c(1, 2))
fit3 <- survfit(coxph(Surv(time, DEATH_EVENT)~strata(factor(EF_Level)),
                      data=data, method="breslow"), type="aalen")

plot(fit3, col=c("black", "red"), lty=c(1,2), xlim=c(0,300),
     xlab="Days", ylab="Survival Probability", main="EF_Level")
legend('bottomright', legend=c("EF Normal", "EF Problem"),
       lty=c(1,2), col=c("black", "red"), cex=1.3)


plot(fit3, fun = 'cumhaz',
     col=c("black", "red"), lty=c(1,2), xlim=c(0,300),
     xlab="Days", ylab="Log Cumulative Hazard Rate", main="EF_Level")
legend('bottomright', legend=c("EF Normal", "EF Problem"),
```

```
       lty=c(1,2), col=c("black", "red"), cex=1.3)


#나트륨 만족
fit5 <- survfit(coxph(Surv(time, DEATH_EVENT)~strata(factor(Na_group)),
                      data=data, method="breslow"), type="aalen")
plot(fit5, col=c("black", "red"), lty=c(1,2), xlim=c(0, 300),
     xlab="Time", ylab="Survival Probability")
legend('bottomright', legend=c("Na Normal", "Na Problem"),
       lty=c(1,2), lwd=1.9, col=c("black", "red"), cex=0.9)


plot(fit5, fun = 'cumhaz',
     col=c("black", "red"), lty=c(1,2), xlim=c(0,300),
     xlab="Days", ylab="Log Cumulative Hazard Rate", main="Na")
legend('bottomright', legend=c("Na Normal", "Na Problem"),
       lty=c(1,2), col=c("black", "red"), cex=0.8)


# 한번에 누적위험함수만 그리기
par(mfrow=c(1, 3))
fit1 <- survfit(coxph(Surv(time, DEATH_EVENT)~strata(factor(high_blood_pressure)),
                      data=data, method="breslow"), type="aalen")
plot(fit1, fun = 'cumhaz',
     col=c("black", "red"), lty=c(1,2), xlim=c(0,300),
     xlab="Days", ylab="Log Cumulative Hazard Rate", main="high blood pressure")
legend('bottomright', legend=c("high blood pressure No", "high blood pressure Yes"),
       lty=c(1,2), col=c("black", "red"), cex=1.3)
```

```
fit5 <- survfit(coxph(Surv(time, DEATH_EVENT)~strata(factor(Na_group)),
                     data=data, method="breslow"), type="aalen")
plot(fit5, fun = 'cumhaz',
     col=c("black", "red"), lty=c(1,2), xlim=c(0,300),
     xlab="Days", ylab="Log Cumulative Hazard Rate", main="Na")
legend('bottomright', legend=c("Na Normal", "Na Problem"),
       lty=c(1,2), col=c("black", "red"), cex=1.3)

fit3 <- survfit(coxph(Surv(time, DEATH_EVENT)~strata(factor(EF_Level)),
                     data=data, method="breslow"), type="aalen")
plot(fit3, fun = 'cumhaz',
     col=c("black", "red"), lty=c(1,2), xlim=c(0,300),
     xlab="Days", ylab="Log Cumulative Hazard Rate", main="EF_Level")
legend('bottomright', legend=c("EF Normal", "EF Problem"),
       lty=c(1,2), col=c("black", "red"), cex=1.3)

#전체 모형 비례성
#6개 모형
cox_model_stepwise <- step(cox_model_empty, scope = list(lower = cox_model_empty,
upper = cox_model_full ),   direction = "both")
summary(cox_model_stepwise)

ph_test <- cox.zph(cox_model_stepwise)
print(ph_test)  #p-value = 0.10

#6.3 다중공선성 분석
#상관분석
```

```
#연속형 변수: age, serum_creatinine, creatinine_phosphokinase
library(ggplot2)
library(reshape2)

# 상관 행렬
cor_matrix <- cor(data[, c("age", "serum_creatinine", "creatinine_phosphokinase")])
cor_matrix_melted <- melt(cor_matrix)

# 색상 정의
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))

# 히트맵 시각화
ggplot(cor_matrix_melted, aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradientn(colors = col(100), limit = c(-1, 1)) +
  theme_minimal() +
  labs(x = "", y = "") +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5)) +
  geom_text(aes(label = round(value, 2)), color = "black", size=5) +
  ggtitle("Correlation Matrix Heatmap") +
  theme(plot.title = element_text(hjust = 0.5))

#범주형 변수: EF_Level, Na_group, high_blood_pressure
chisq_test_1 <- chisq.test(table(data$EF_Level, data$Na_group))
chisq_test_2 <- chisq.test(table(data$Na_group, data$high_blood_pressure))
chisq_test_3 <- chisq.test(table(data$EF_Level, data$high_blood_pressure))
```

chisq_test_1

chisq_test_2

chisq_test_3

#범주형-연속형 독립 T검정

EF_Level과 연속형 변수들에 대한 t검정

t_test_EF_age <- t.test(age ~ EF_Level, data = data)

t_test_EF_serum_creatinine <- t.test(serum_creatinine ~ EF_Level, data = data)

t_test_EF_creatinine_phosphokinase <- t.test(creatinine_phosphokinase ~ EF_Level, data = data)

Na_group과 연속형 변수들에 대한 t검정

t_test_Na_group_age <- t.test(age ~ Na_group, data = data)

t_test_Na_group_serum_creatinine <- t.test(serum_creatinine ~ Na_group, data = data)

t_test_Na_group_creatinine_phosphokinase <- t.test(creatinine_phosphokinase ~ Na_group, data = data)

high_blood_pressure과 연속형 변수들에 대한 t검정

t_test_hbp_age <- t.test(age ~ high_blood_pressure, data = data)

t_test_hbp_serum_creatinine <- t.test(serum_creatinine ~ high_blood_pressure, data = data)

t_test_hbp_creatinine_phosphokinase <- t.test(creatinine_phosphokinase ~ high_blood_pressure, data = data)

t_test_EF_age

t_test_EF_serum_creatinine

t_test_EF_creatinine_phosphokinase

t_test_Na_group_age

t_test_Na_group_serum_creatinine

t_test_Na_group_creatinine_phosphokinase

t_test_hbp_age

t_test_hbp_serum_creatinine

t_test_hbp_creatinine_phosphokinase