CH6

2023-08-20

```
library(tidyverse)
dt <- read.csv("C:\\Users\\phl02\\Desktop\\P\\bio\\ch6\\Ch6_regression.csv")
head(dt)
```

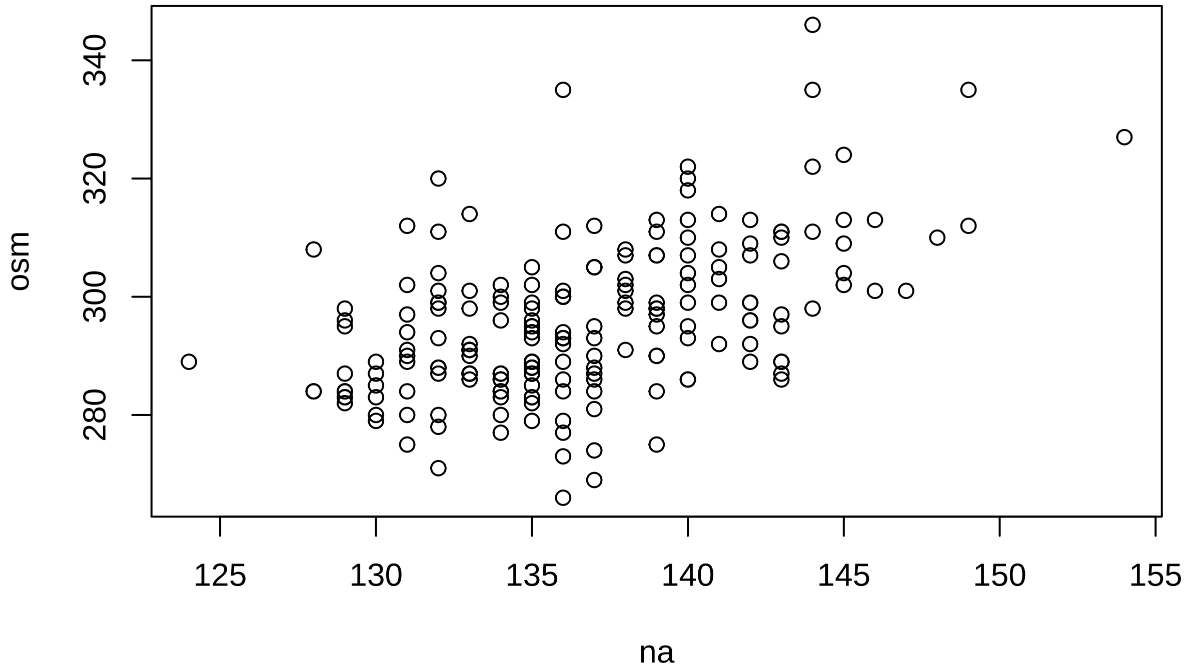
```
##   id osm  na bun glucose weight height
## 1  1 296 142   9     184     50    167
## 2  2 298 144  12     144     59    159
## 3  3 278 132  13      99     90    161
## 4  4 307 142  13     104     90    152
## 5  5 307 139  11     115     88    175
## 6  6 293 140  11     105     99    161
```

1. 회귀 분석

1.1 단순 선형 회귀 분석

1) 상관관계 알아보기

```
plot(osm~na, data=dt)
```

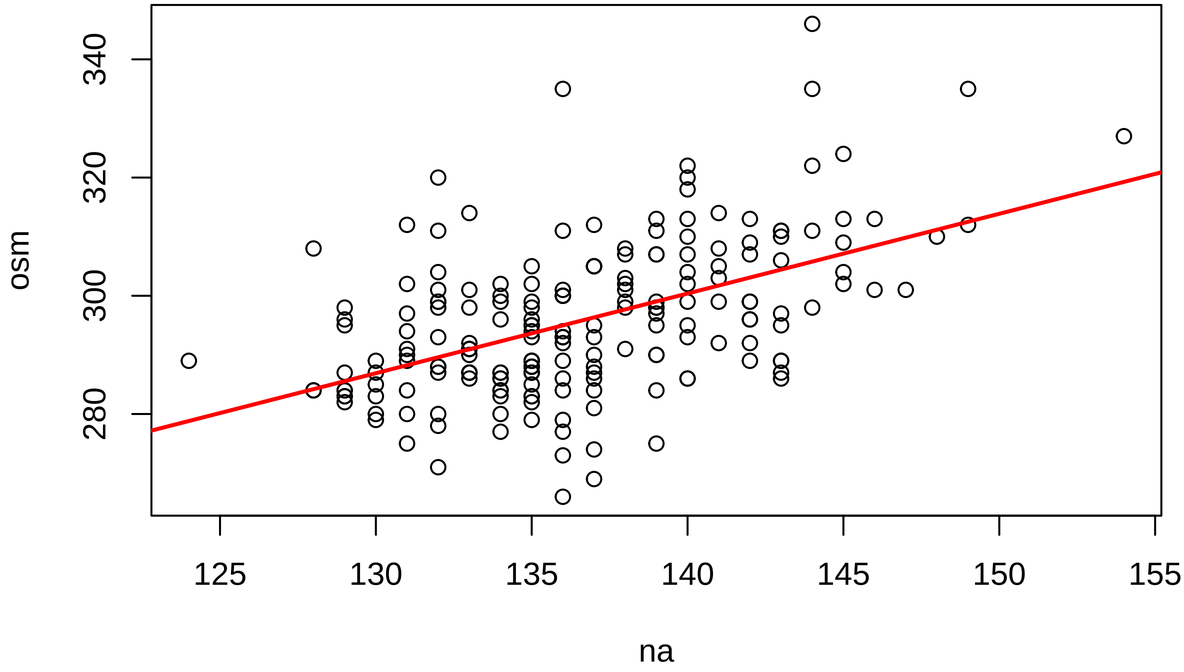

osm
280
300
320
340
125
130
135
140
145
150
155
na

2) 회귀식 추정

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

```
##
## Call:
## lm(formula = osm ~ na, data = dt)
##
## Coefficients:
## (Intercept)           na
##     111.632        1.348
```

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



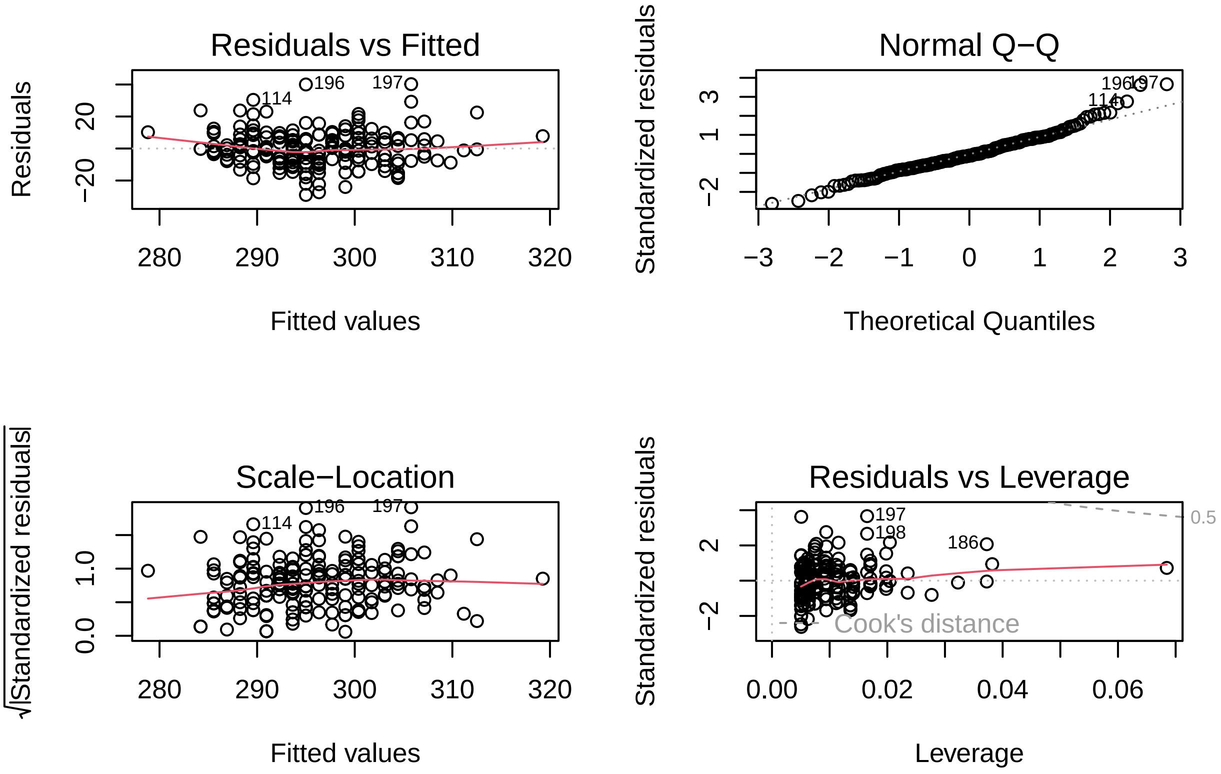
3) 결정계수 찾고 4)유의한 회귀식 모형인지 검증

```r
summary(fit)
```

```
##
## Call:
## lm(formula = osm ~ na, data = dt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.995  -6.950  -1.039   6.568  40.219
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 111.6318    21.9028   5.097 8.04e-07 ***
## na            1.3483     0.1603   8.413 7.87e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.08 on 198 degrees of freedom
## Multiple R-squared:  0.2634, Adjusted R-squared:  0.2596
## F-statistic: 70.78 on 1 and 198 DF,  p-value: 7.865e-15
```

6) 기본 가정이 충족하는지 확인

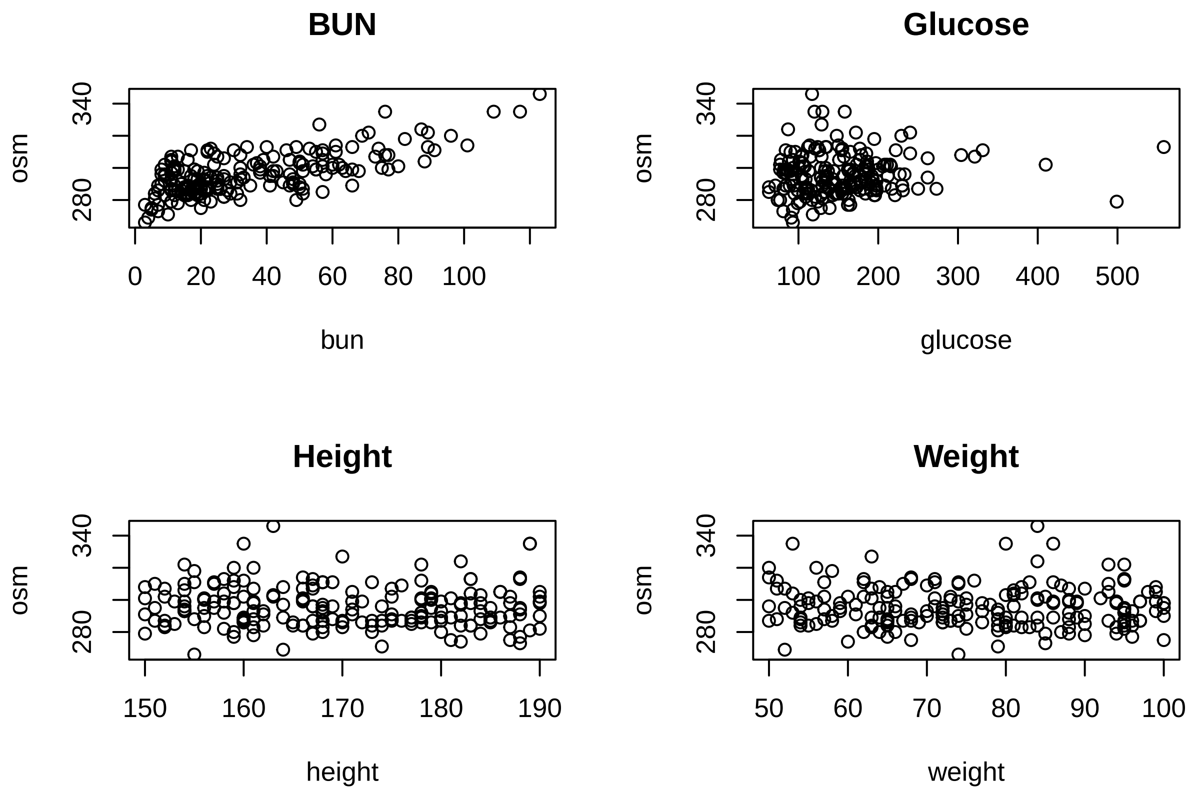
```
par(mfrow=c(2,2))
plot(fit)
```



1.2 다중 회귀 분석

1) 산점도 이용하여 데이터 분포 살펴보기

```r
par(mfrow=c(2,2))
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")
```



2) 다중 선형 회귀식 추정

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

```
##
## Call:
## lm(formula = osm ~ na + bun + glucose + height + weight, data = dt)
##
## Coefficients:
## (Intercept)           na          bun      glucose       height       weight
##   84.917430     1.409028     0.369772     0.028286     0.002827     0.011948
```

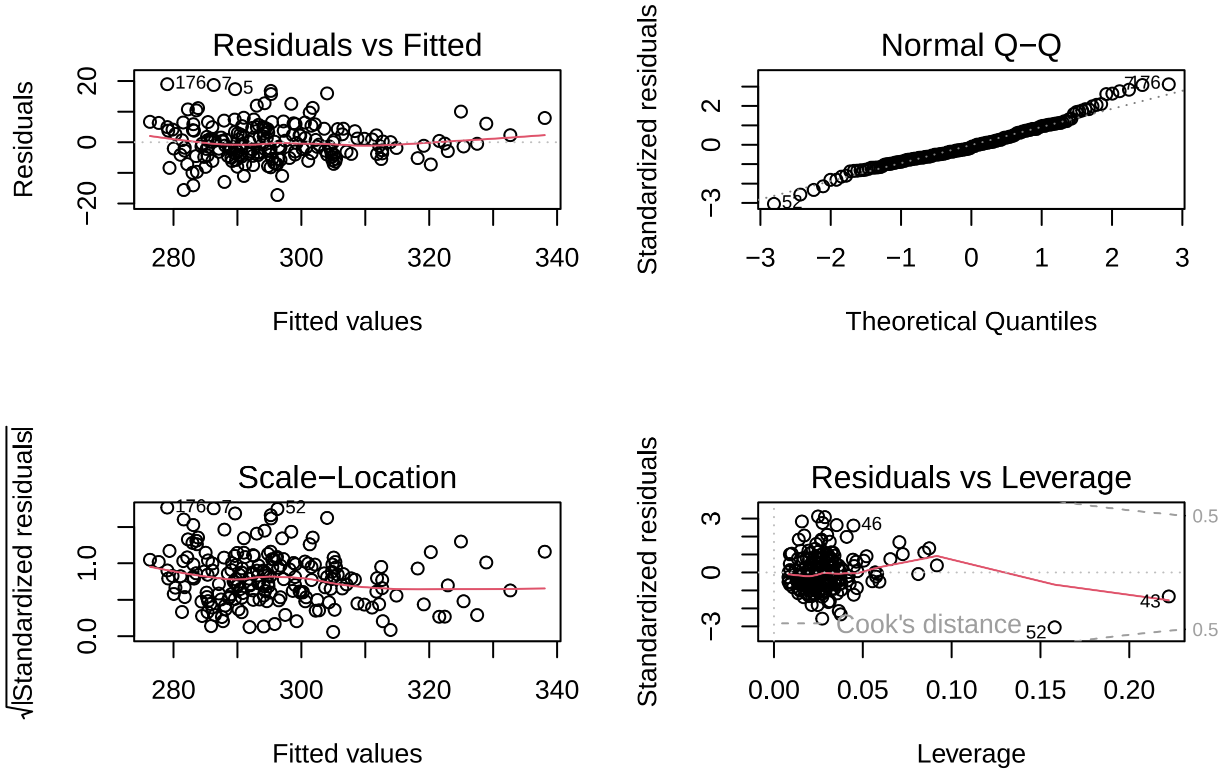
3) 유의성 및 결정계수 검정

```
summary(fit.multi)
```

```
##
## Call:
## lm(formula = osm ~ na + bun + glucose + height + weight, data = dt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.2508  -3.8833  -0.7117   3.7632  18.9770
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 84.917430  14.663494   5.791 2.78e-08 ***
## na           1.409028   0.090263  15.610  < 2e-16 ***
## bun          0.369772   0.017821  20.749  < 2e-16 ***
## glucose      0.028286   0.006709   4.216 3.81e-05 ***
## height       0.002827   0.037403   0.076    0.940
## weight       0.011948   0.030551   0.391    0.696
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.166 on 194 degrees of freedom
## Multiple R-squared:  0.7763, Adjusted R-squared:  0.7705
## F-statistic: 134.6 on 5 and 194 DF,  p-value: < 2.2e-16
```

4) 가정 검정

```r
par(mfrow=c(2,2))
plot(fit.multi)
```



6) 선택의 기준: AIC

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

```
## [1] 0.2596306
```

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

```
## [1] 0.7532313
```

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

```
## [1] 0.7726867
```

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

```
## [1] 0.7715272
```

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

```
## [1] 0.7705304
```

- 한 번에 하는 방법

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

```
## Start:  AIC=733.52
## osm ~ na + bun + glucose + height + weight
## 
##           Df Sum of Sq     RSS    AIC
## - height   1       0.2  7375.6 731.52
## - weight   1       5.8  7381.2 731.67
## <none>                  7375.3 733.52
## - glucose  1     675.8  8051.2 749.05
## - na       1    9264.1 16639.5 894.24
## - bun      1   16367.1 23742.5 965.34
## 
## Step:  AIC=731.52
## osm ~ na + bun + glucose + weight
## 
##           Df Sum of Sq     RSS    AIC
## - weight   1       5.8  7381.4 729.68
## <none>                  7375.6 731.52
## - glucose  1     677.2  8052.8 747.09
## - na       1    9392.0 16767.5 893.78
## - bun      1   16367.9 23743.4 963.35
## 
## Step:  AIC=729.68
## osm ~ na + bun + glucose
## 
##           Df Sum of Sq     RSS    AIC
## <none>                  7381.4 729.68
## - glucose  1     672.6  8054.0 745.12
## - na       1    9500.7 16882.1 893.14
## - bun      1   16509.7 23891.0 962.59
```

```
## 
## Call:
## lm(formula = osm ~ na + bun + glucose, data = dt)
## 
## Coefficients:
## (Intercept)           na          bun      glucose  
##    86.08723      1.41118      0.36897      0.02795
```

7) 또 다른 방법

```
library(olsrr)

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

- 모델 평가

```
ols_step_all_possible(fit.multi)
```

```
##    Index N                   Predictors    R-Square Adj. R-Square Mallow's Cp
## 2      1 1                          bun 0.469580229  0.4669013413  263.989252
## 1      2 1                           na 0.263351044  0.2596305952  442.834789
## 3      3 1                      glucose 0.010670380  0.0056737662  661.963856
## 4      4 1                       height 0.004144047 -0.0008855289  667.623606
## 5      5 1                       weight 0.001187392 -0.0038571161  670.187668
## 6      6 2                       na bun 0.755711356  0.7532312680   17.851362
## 10     7 2                  bun glucose 0.487944403  0.4827458693  250.063521
## 11     8 2                   bun height 0.473043212  0.4676933969  262.986093
## 12     9 2                   bun weight 0.471180988  0.4658122670  264.601046
## 7     10 2                   na glucose 0.275353003  0.2679961805  434.426482
## 9     11 2                    na weight 0.269830290  0.2624174000  439.215874
## 8     12 2                    na height 0.263353145  0.2558744972  444.832967
## 13    13 2               glucose height 0.015274811  0.0052776009  659.970814
## 14    14 2               glucose weight 0.011121737  0.0010823641  663.572431
## 15    15 2                height weight 0.005426749 -0.0046704413  668.511224
## 16    16 3               na bun glucose 0.776113536  0.7726867019    2.158237
## 17    17 3                na bun height 0.755760031  0.7520216643   19.809150
## 18    18 3                na bun weight 0.755749093  0.7520105587   19.818635
## 22    19 3           bun glucose height 0.491958079  0.4841819272  248.582791
## 23    20 3           bun glucose weight 0.491418246  0.4836338314  249.050944
## 24    21 3            bun height weight 0.474543000  0.4665002903  263.685451
## 20    22 3            na glucose weight 0.279830133  0.2688071251  432.543836
## 19    23 3            na glucose height 0.275377682  0.2642865239  436.405079
## 21    24 3             na height weight 0.269835424  0.2586594352  441.211422
## 25    25 3        glucose height weight 0.015775970  0.0007113166  661.536200
## 27    26 4        na bun glucose weight 0.776289414  0.7717004785    4.005712
## 26    27 4        na bun glucose height 0.776119635  0.7715272175    4.152947
## 28    28 4         na bun height weight 0.755796796  0.7507874994   21.777267
## 30    29 4    bun glucose height weight 0.495302430  0.4849496597  247.682512
## 29    30 4     na glucose height weight 0.279858999  0.2650868756  434.518804
## 31    31 5 na bun glucose height weight 0.776296000  0.7705304333    6.000000
```

- 단계적 선택법으로 모델 비교

```r
ols_step_best_subset(fit.multi)
```

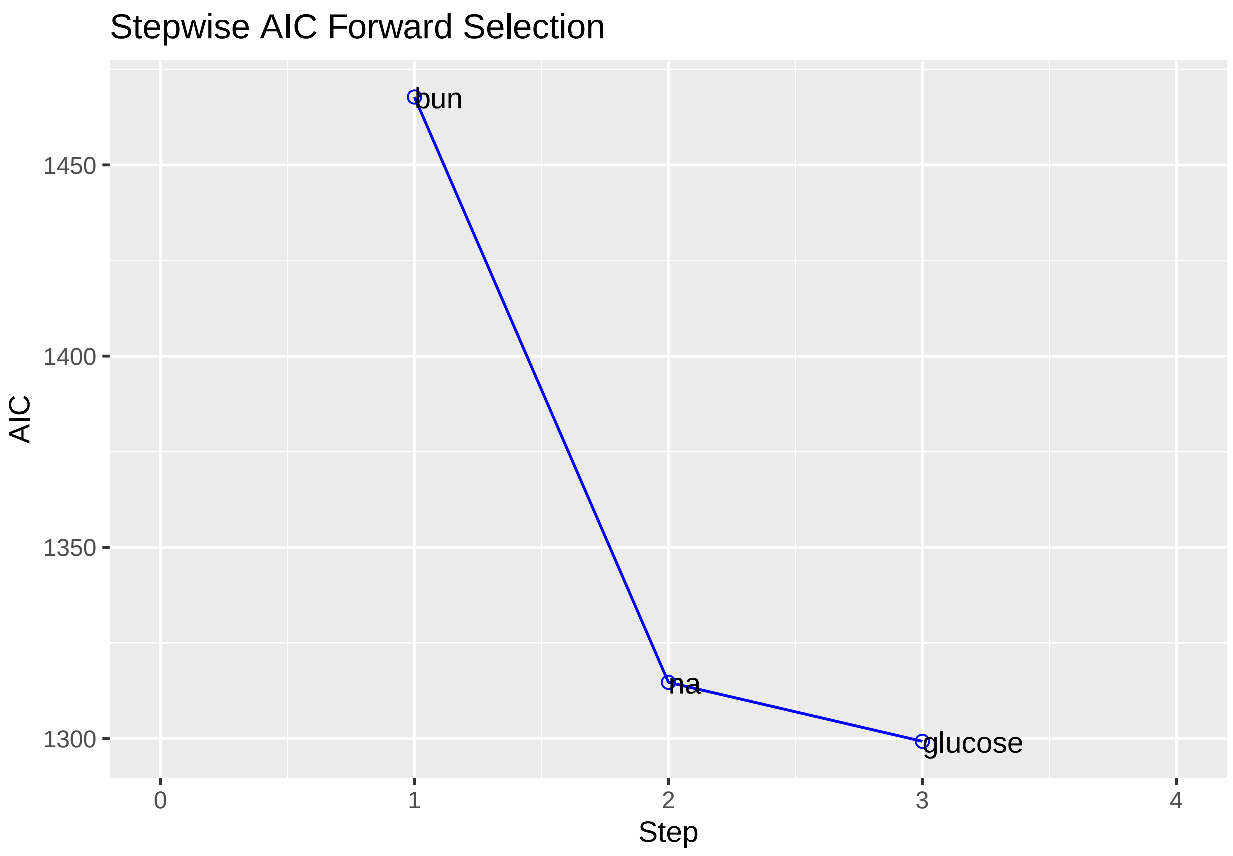
```
##           Best Subsets Regression
## -------------------------------------------
## Model Index    Predictors
## -------------------------------------------
##      1         bun
##      2         na bun
##      3         na bun glucose
##      4         na bun glucose weight
##      5         na bun glucose height weight
## -------------------------------------------
##
##                                                         Subsets Regression Summary
## ----------------------------------------------------------------------------------------------------
##                         Adj.        Pred
## Model    R-Square    R-Square    R-Square      C(p)         AIC         SBIC         SBC          MSI
## ----------------------------------------------------------------------------------------------------
##   1        0.4696      0.4669      0.4574    263.9893    1467.7603    897.2852    1477.6553    17664
##   2        0.7557      0.7532      0.7484     17.8514    1314.6967    746.7794    1327.8899     8176
##   3        0.7761      0.7727      0.7633      2.1582    1299.2544    731.9179    1315.7460     7532
##   4        0.7763      0.7717      0.7609      4.0057    1301.0972    733.8289    1320.8871     7565
##   5        0.7763      0.7705      0.7585      6.0000    1303.0913    735.8851    1326.1795     7604
## ----------------------------------------------------------------------------------------------------
## AIC: Akaike Information Criteria
##  SBIC: Sawa's Bayesian Information Criteria
##  SBC: Schwarz Bayesian Criteria
##  MSEP: Estimated error of prediction, assuming multivariate normality
##  FPE: Final Prediction Error
##  HSP: Hocking's Sp
##  APC: Amemiya Prediction Criteria
```

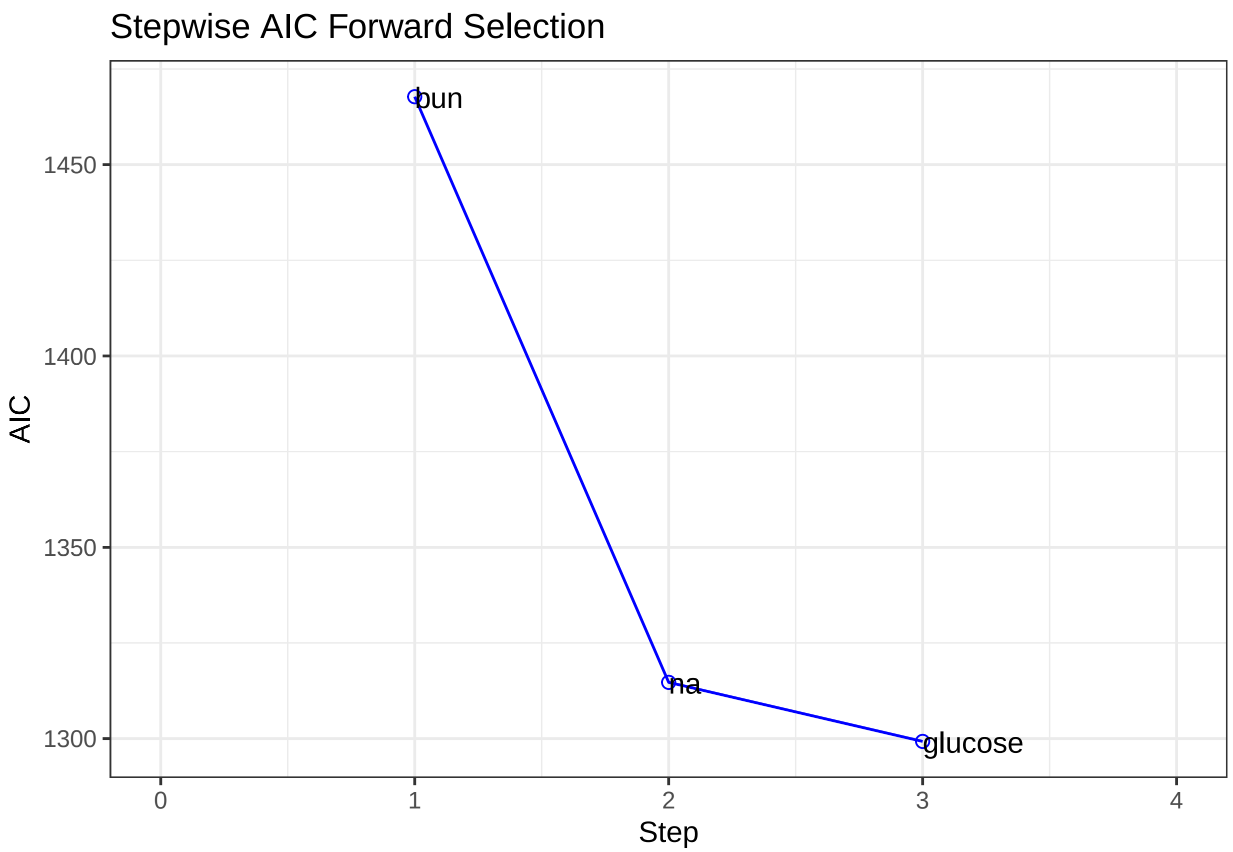
- 전진 선택법

```
ols_step_forward_aic(fit.multi)
```

```
##
##                            Selection Summary
## ------------------------------------------------------------------------
## Variable        AIC          Sum Sq          RSS          R-Sq       Adj. R-Sq
## ------------------------------------------------------------------------
## bun           1467.760     15481.675     17487.505     0.46958       0.46690
## na            1314.697     24915.184      8053.996     0.75571       0.75323
## glucose       1299.254     25587.827      7381.353     0.77611       0.77269
## ------------------------------------------------------------------------
```

```
plot(ols_step_forward_aic(fit.multi))+theme_bw()
```



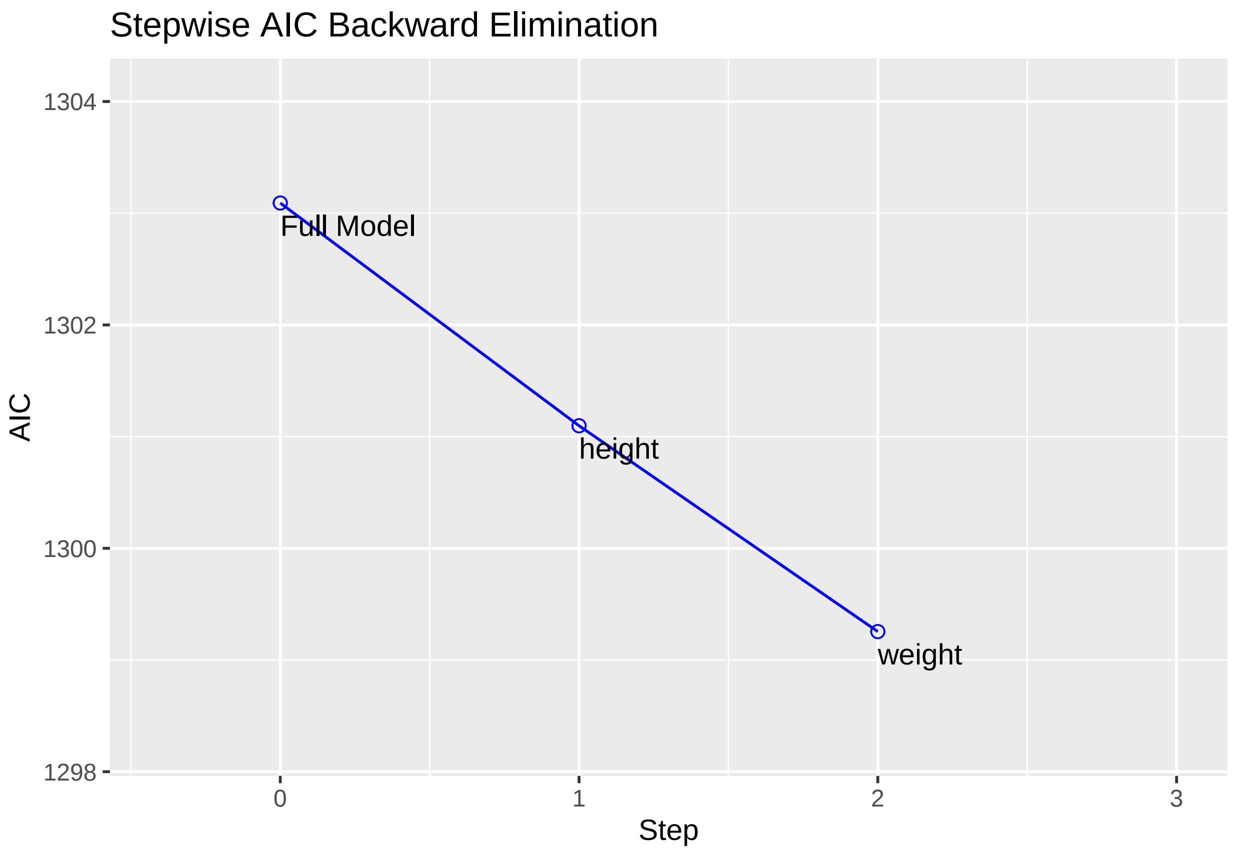

Stepwise AIC Forward Selection
AIC
1450
1400
1350
1300
Step
0
1
2
3
4
bun
na
glucose

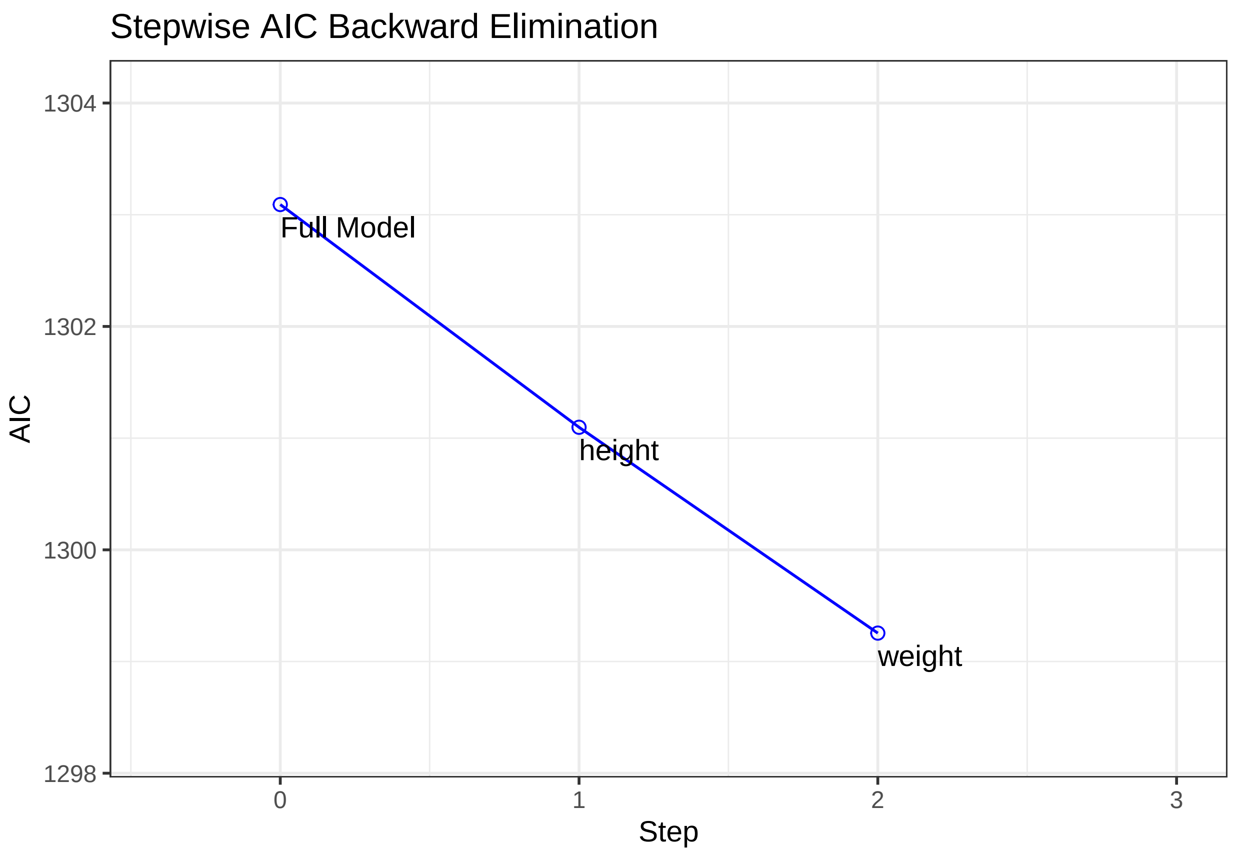
- 후진 선택법

```
ols_step_backward_aic(fit.multi)
```

```
##
##
##                        Backward Elimination Summary
## -----------------------------------------------------------------------
## Variable        AIC         RSS        Sum Sq      R-Sq      Adj. R-Sq
## -----------------------------------------------------------------------
## Full Model    1303.091    7375.337    25593.843    0.77630     0.77053
## height        1301.097    7375.555    25593.625    0.77629     0.77170
## weight        1299.254    7381.353    25587.827    0.77611     0.77269
## -----------------------------------------------------------------------
```

```
plot(ols_step_backward_aic(fit.multi))+theme_bw()
```




Stepwise AIC Backward Elimination
AIC
1304
1302
1300
1298
Full Model
height
weight
0
1
2
3
Step

2. 일반화 선형 분석

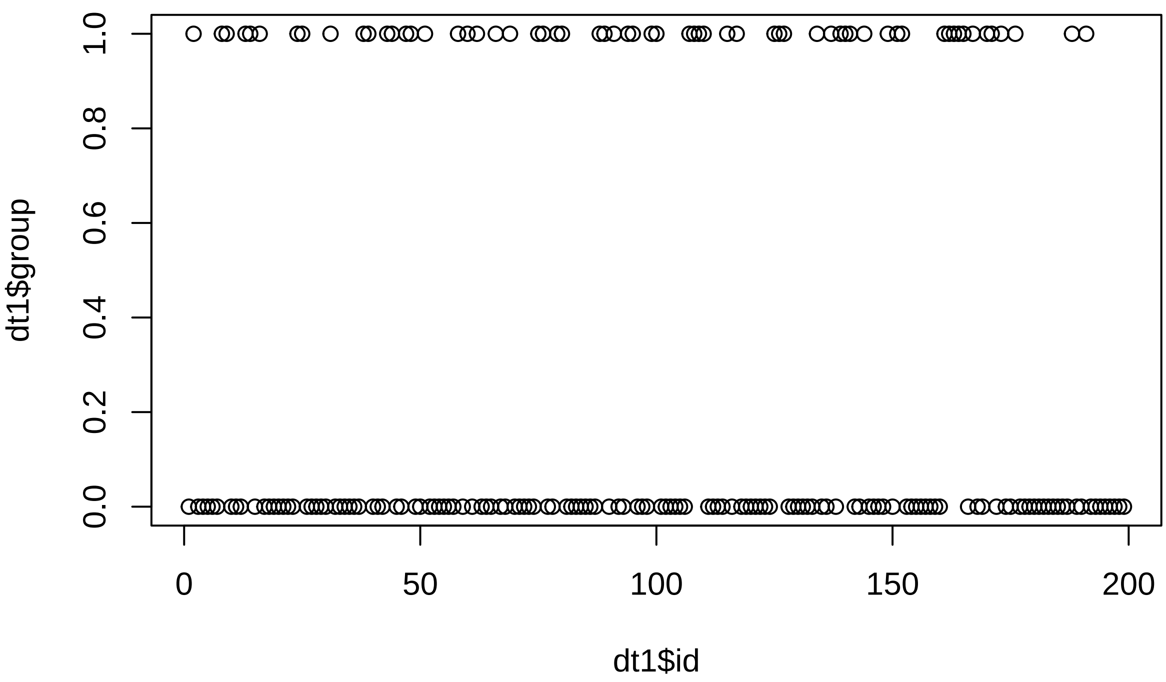
2.1 로지스틱 회귀 분석

1) 로지스틱 회귀분석의 기본

```r
dt1<-read_csv("C:\\Users\\phl02\\Desktop\\P\\bio\\ch6\\Ch6_logistic.csv")
head(dt1)
```

```
## # A tibble: 6 x 9
##      id   age gender group ibd   cirrhosis diabetes htn   aspirin
##   <dbl> <dbl> <chr>  <dbl> <chr> <chr>     <chr>    <chr> <chr>
## 1     1    65 female     0 none  none      none     htn   aspirin_user
## 2     2    78 male       1 none  none      none     htn   aspirin_user
## 3     3    59 female     0 none  none      none     none  no_aspirin
## 4     4    28 male       0 ibd   none      none     none  no_aspirin
## 5     5    77 female     0 none  none      diabetes none  no_aspirin
## 6     6    52 female     0 ibd   none      none     htn   aspirin_user
```

```r
plot(dt1$id, dt1$group)
```



4) 우도비검정

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

```
##
##        Descriptive Statistics by 'group'
## -----------------------------------------------
##                         0           1        p
##                      (N=137)     (N=62)
## -----------------------------------------------
##  aspirin                                  0.001
##    - aspirin_user 63 (46.0%)  12 (19.4%)
##    - no_aspirin   74 (54.0%)  50 (80.6%)
##  ibd                                      0.000
##    - ibd           3 ( 2.2%)  11 (17.7%)
##    - none         134 (97.8%) 51 (82.3%)
##  diabetes                                 0.087
##    - diabetes     13 ( 9.5%)  12 (19.4%)
##    - none         124 (90.5%) 50 (80.6%)
##  gender                                   0.992
##    - female       69 (50.4%)  32 (51.6%)
##    - male         68 (49.6%)  30 (48.4%)
##  age              62.1 ± 13.7 62.2 ± 11.8 0.956
## -----------------------------------------------
```

5) 공식

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

```
##
## Call:  glm(formula = group ~ age + gender + ibd + cirrhosis + diabetes +
##     htn + aspirin, family = binomial, data = dt1)
##
## Coefficients:
##       (Intercept)                age         gendermale            ibdnone
##            1.8461             0.0266            -0.1868            -2.5252
##     cirrhosisnone       diabetesnone            htnnone  aspirinno_aspirin
##           -2.4032            -0.6224            -0.1894             1.5741
##
## Degrees of Freedom: 198 Total (i.e. Null);  191 Residual
## Null Deviance:       246.9
## Residual Deviance: 203.5     AIC: 219.5
```

```r
summary(fit)
```

```
##
## Call:
## glm(formula = group ~ age + gender + ibd + cirrhosis + diabetes +
##     htn + aspirin, family = binomial, data = dt1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0773  -0.8040  -0.5290   0.6526   2.2647
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.84611    1.64532   1.122 0.261845
## age                 0.02660    0.01498   1.776 0.075779 .
## gendermale         -0.18685    0.35357  -0.528 0.597175
## ibdnone            -2.52519    0.72481  -3.484 0.000494 ***
## cirrhosisnone      -2.40317    1.11002  -2.165 0.030390 *
## diabetesnone       -0.62241    0.54008  -1.152 0.249137
## htnnone            -0.18937    0.40104  -0.472 0.636788
## aspirinno_aspirin  1.57407    0.43921   3.584 0.000339 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 246.90  on 198  degrees of freedom
## Residual deviance: 203.53  on 191  degrees of freedom
## AIC: 219.53
##
## Number of Fisher Scoring iterations: 4
```

6) 유의한 독립변수만 포함

```r
step(fit, type='backward')
```

```
## Start:  AIC=219.53
## group ~ age + gender + ibd + cirrhosis + diabetes + htn + aspirin
##
##             Df Deviance    AIC
## - htn        1   203.75 217.75
## - gender     1   203.81 217.81
## - diabetes   1   204.83 218.83
## <none>           203.53 219.53
## - age        1   206.82 220.82
## - cirrhosis  1   210.62 224.62
## - aspirin    1   218.21 232.21
## - ibd        1   218.80 232.80
##
## Step:  AIC=217.75
## group ~ age + gender + ibd + cirrhosis + diabetes + aspirin
##
##             Df Deviance    AIC
## - gender     1   204.01 216.01
## - diabetes   1   205.50 217.50
## <none>           203.75 217.75
## - age        1   207.36 219.36
## - cirrhosis  1   210.74 222.74
## - aspirin    1   218.54 230.54
## - ibd        1   218.90 230.90
##
## Step:  AIC=216.01
## group ~ age + ibd + cirrhosis + diabetes + aspirin
##
##             Df Deviance    AIC
## - diabetes   1   205.59 215.59
## <none>           204.01 216.01
## - age        1   207.86 217.86
## - cirrhosis  1   211.39 221.39
## - aspirin    1   218.69 228.69
## - ibd        1   219.00 229.00
##
## Step:  AIC=215.59
## group ~ age + ibd + cirrhosis + aspirin
##
##             Df Deviance    AIC
## <none>           205.59 215.59
## - age        1   210.27 218.27
## - cirrhosis  1   215.30 223.30
## - aspirin    1   220.28 228.28
## - ibd        1   220.35 228.35
```

```
##
## Call:  glm(formula = group ~ age + ibd + cirrhosis + aspirin, family = binomial,
##     data = dt1)
##
## Coefficients:
##       (Intercept)                age            ibdnone      cirrhosisnone
##           1.11990            0.03074           -2.47163           -2.69886
## aspirinno_aspirin
##           1.51259
##
## Degrees of Freedom: 198 Total (i.e. Null);  194 Residual
## Null Deviance:       246.9
## Residual Deviance: 205.6     AIC: 215.6
```

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

```
extractOR(final.fit)
```

```
##                     OR  lcl   ucl      p
## (Intercept)       3.06 0.14 65.82 0.4742
## age               1.03 1.00  1.06 0.0355
## ibdnone           0.08 0.02  0.35 0.0006
## cirrhosisnone     0.07 0.01  0.58 0.0137
## aspirinno_aspirin 4.54 1.98 10.39 0.0003
```

7) 회귀모형 평가

```r
library(fmsb)
NagelkerkeR2(final.fit)
```

```
## $N
## [1] 199
##
## $R2
## [1] 0.2637072
```

2.2 모형의 성능

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

1) Nagelkerke 결정계수

```
r2_nagelkerke(final.fit)
```

```
## Nagelkerke's R2
##       0.2637072
```

2) Hosmer-Lemeshow goodness-of fit test

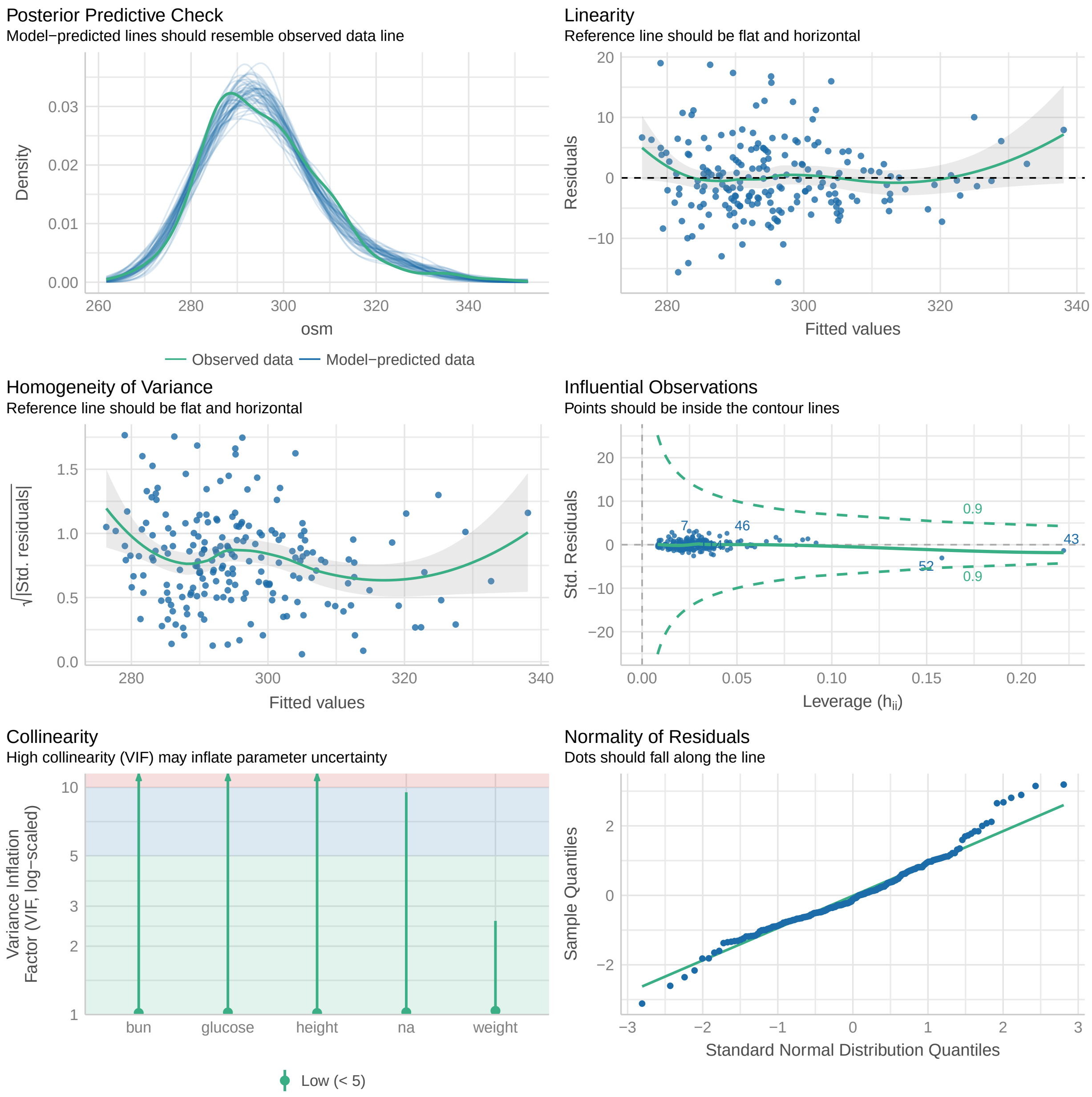
```
performance_hosmer(final.fit)
```

```
## # Hosmer-Lemeshow Goodness-of-Fit Test
##
##   Chi-squared: 5.092
##            df: 8
##       p-value: 0.748
```

```
## Summary: model seems to fit well.
```

3) 회귀모형 가정에 위배되는지 확인

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



Posterior Predictive Check
Model−predicted lines should resemble observed data line

Linearity
Reference line should be flat and horizontal

Homogeneity of Variance
Reference line should be flat and horizontal

Influential Observations
Points should be inside the contour lines

Collinearity
High collinearity (VIF) may inflate parameter uncertainty

Normality of Residuals
Dots should fall along the line

4) 더 나은 모형 선택

```
model_performance(fit)
```

```
## # Indices of model performance
##
## AIC      |     AICc |      BIC |    R2 | R2 (adj.) |  RMSE | Sigma
## ------------------------------------------------------------------
## 1303.091 | 1303.675 | 1326.180 | 0.776 |     0.771 | 6.073 | 6.166
```

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

```
## # Comparison of Model Performance Indices
##
## Name | Model |    R2 | R2 (adj.) |  RMSE | Sigma | AIC weights | AICc weights | BIC weights | Perfor
## ----------------------------------------------------------------------------------------------------
## fit1 |    lm | 0.776 |     0.773 | 6.075 | 6.137 |       0.872 |        0.887 |       0.995 |
## fit  |    lm | 0.776 |     0.771 | 6.073 | 6.166 |       0.128 |        0.113 |       0.005 |
```

3. ROC 관련 분석

3.4 ROC 곡선 직접 그려보기

```r
roc.ex<-read_csv("C:\\Users\\phl02\\Desktop\\P\\bio\\ch6\\Ch6_afp.csv")
head(roc.ex)
```

```
## # A tibble: 6 x 3
##   group   afp pivka
##   <chr> <dbl> <dbl>
## 1 HCC    49.5    38
## 2 HCC    14.6    18
## 3 HCC     9.4    22
## 4 HCC     4.2    66
## 5 HCC     8.9    28
## 6 HCC     8.9    32
```

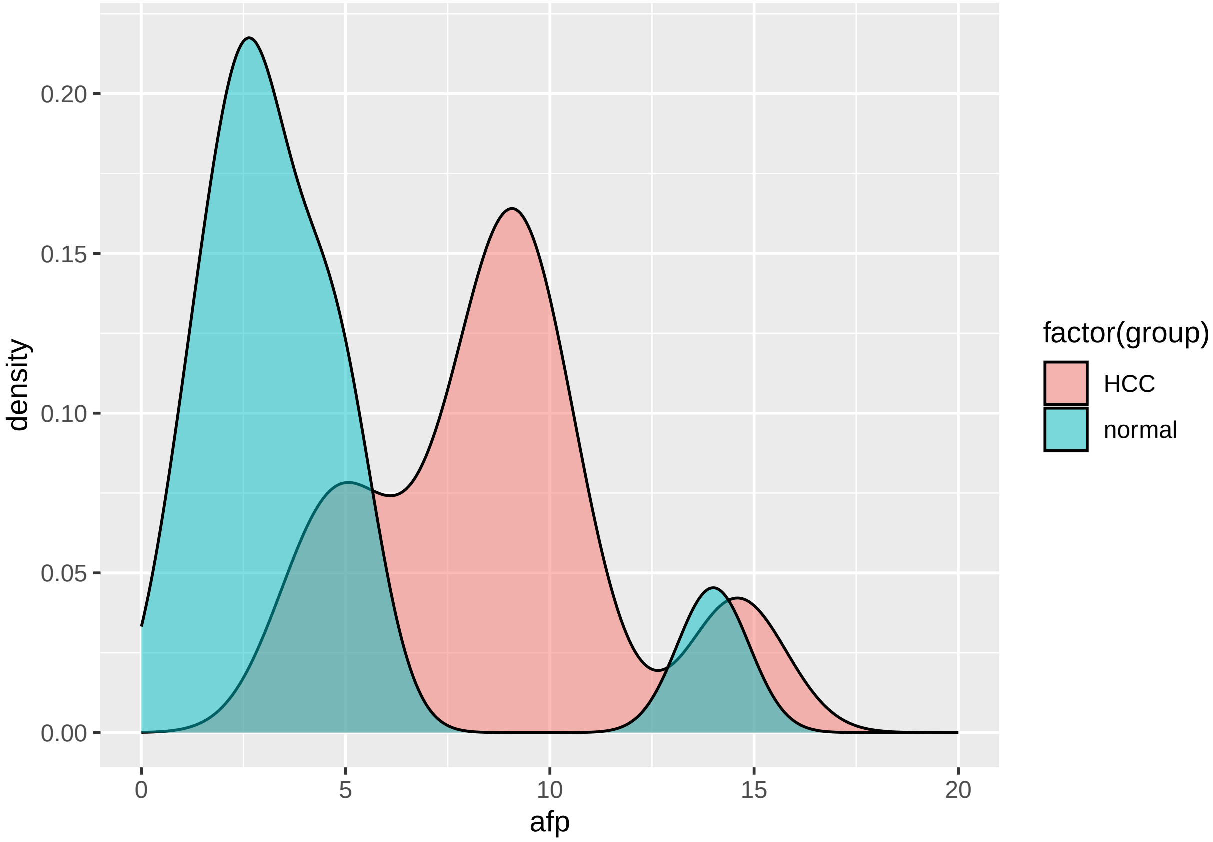
편의상 afp변수를 내림차순으로 정렬
간암/정상: 10명

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

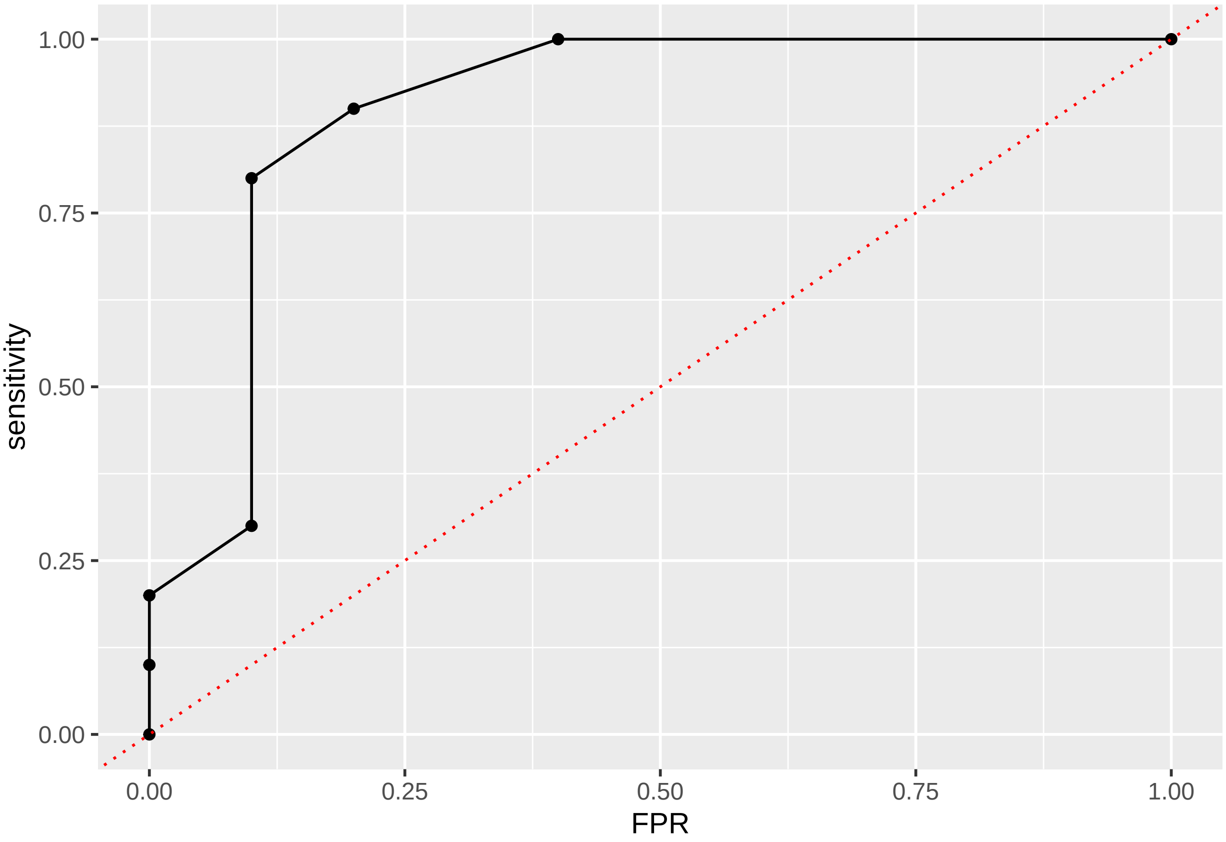
```
## # A tibble: 20 x 3
##    group    afp pivka
##    <chr>  <dbl> <dbl>
##  1 HCC     86.5    20
##  2 HCC     49.5    38
##  3 HCC     14.6    18
##  4 normal  14      17
##  5 HCC     10.4    20
##  6 HCC      9.4    22
##  7 HCC      8.9    28
##  8 HCC      8.9    32
##  9 HCC      7.5    33
## 10 HCC      5.4    46
## 11 normal   5.3    29
## 12 normal   4.5    17
## 13 normal   4.3    10
## 14 HCC      4.2    66
## 15 normal   3      31
## 16 normal   3      11
## 17 normal   2.6    14
## 18 normal   2.3    11
## 19 normal   1.7    32
## 20 normal   1      18
```

3.4 부분은 책에서 코드 없이 예시로만 보여줘서 직접했음

```r
library(ggplot2)
ggplot(roc.ex,aes(x=afp,fill=factor(group)))+
  geom_density(alpha=0.5)+xlim(0,20)
```



```r
sensitivity <-c(0,0.1,0.2,0.3,0.8,0.9,1,1)
FPR <-1-c(1,1,1,0.9,0.9,0.8,0.6,0)
result <- data.frame(cbind(sensitivity,FPR))
ggplot(result,aes(x=FPR,y=sensitivity))+geom_point()+
  geom_line()+
  geom_abline(intercept = 0,slope = 1,color='red',linetype=3)
```



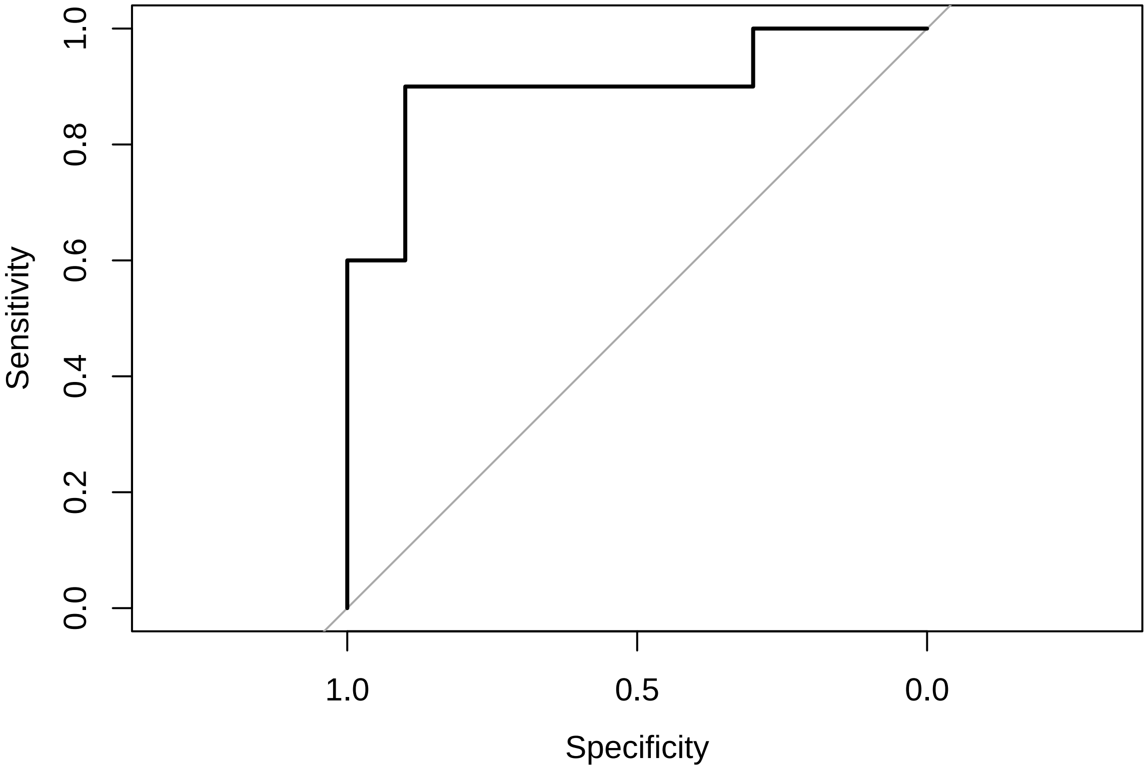
3.5 pROC 패키지

1) ROC 객체 생성

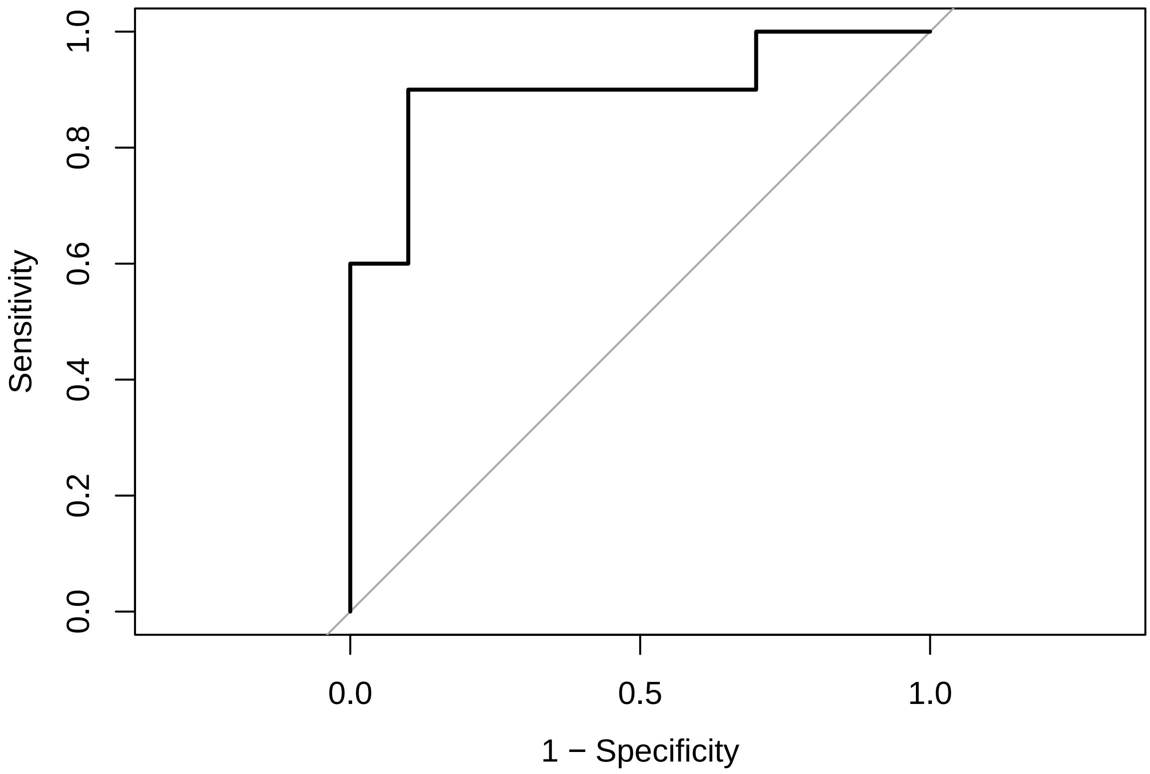
```r
library(pROC)
afp<-roc(roc.ex$group, roc.ex$afp, ci=TRUE)
afp
```

```
##
## Call:
## roc.default(response = roc.ex$group, predictor = roc.ex$afp,     ci = TRUE)
##
## Data: roc.ex$afp in 10 controls (roc.ex$group HCC) > 10 cases (roc.ex$group normal).
## Area under the curve: 0.9
## 95% CI: 0.7482-1 (DeLong)
```

```r
plot(afp)
```

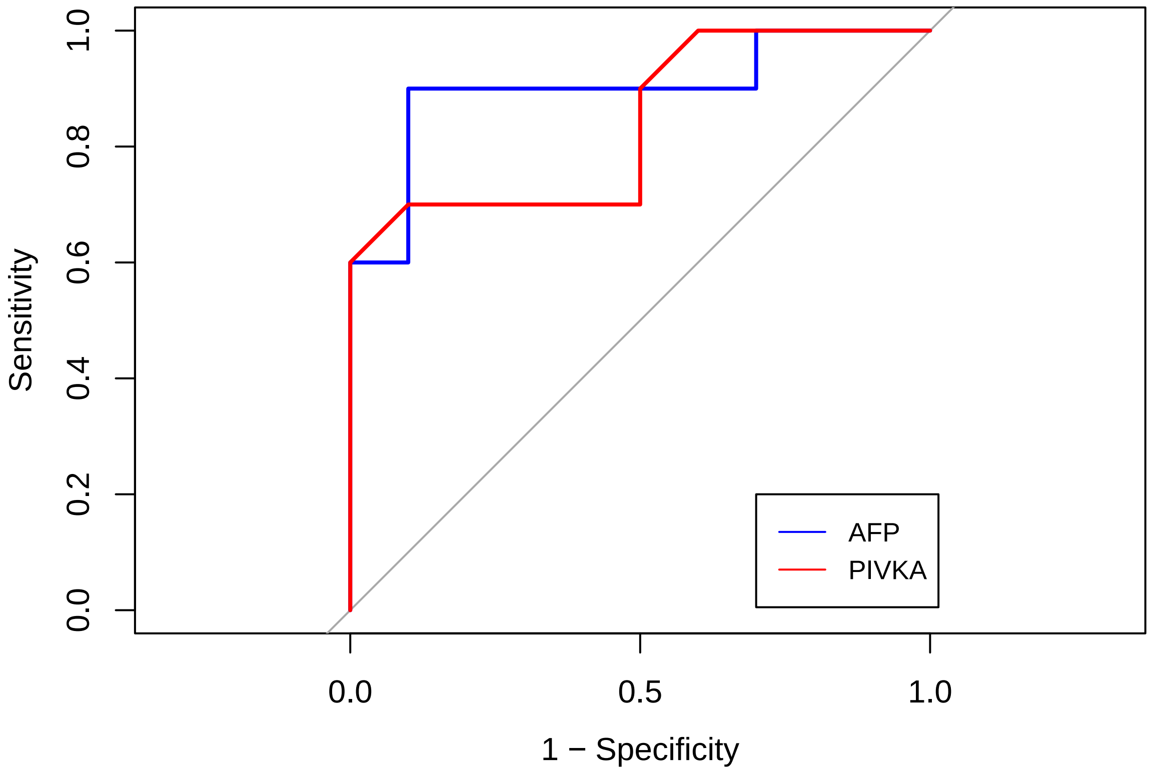


```
afp<-roc(roc.ex$group, roc.ex$afp)
plot(afp,legacy.axes=TRUE)
```



2) 겹쳐 그리기

```
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)
```



3) ROC비교

```r
roc.test(afp, pivka)
```

```
##
##  DeLong's test for two correlated ROC curves
##
## data:  afp and pivka
## Z = 0.44073, p-value = 0.6594
## alternative hypothesis: true difference in AUC is not equal to 0
## 95 percent confidence interval:
##  -0.206824  0.326824
## sample estimates:
## AUC of roc1 AUC of roc2
##        0.90        0.84
```

4) 최적의 cut-off 찾기

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

```
## 95% CI (1000 stratified bootstrap replicates):
##  thresholds sp.low sp.median sp.high se.low se.median se.high
##        5.35    0.7       0.9       1    0.7       0.9       1
```

5) 특정 cut-off에서 민감도, 특이도 계산하기

```r
metric<-c('sensitivity','specificity','ppv','npv')

afp.cutoff<-ci.coords(afp, x=5, input="threshold", metric)
afp.cutoff
```

```
## 95% CI (2000 stratified bootstrap replicates):
##   threshold sensitivity.low sensitivity.median sensitivity.high specificity.low
## 5         5             0.5                0.8                1             0.7
##   specificity.median specificity.high ppv.low ppv.median ppv.high npv.low
## 5                0.9                1  0.6998     0.8889        1  0.6427
##   npv.median npv.high
## 5     0.8182        1
```

- 민감도

```r
afp.cutoff$sensitivity
```

```
##      2.5% 50% 97.5%
## [1,]  0.5 0.8     1
```

- 특이도

```r
afp.cutoff$specificity
```

```
##      2.5% 50% 97.5%
## [1,]  0.7 0.9     1
```

- ppv

```r
afp.cutoff$ppv
```

```
##           2.5%       50% 97.5%
## [1,] 0.6998077 0.8888889     1
```

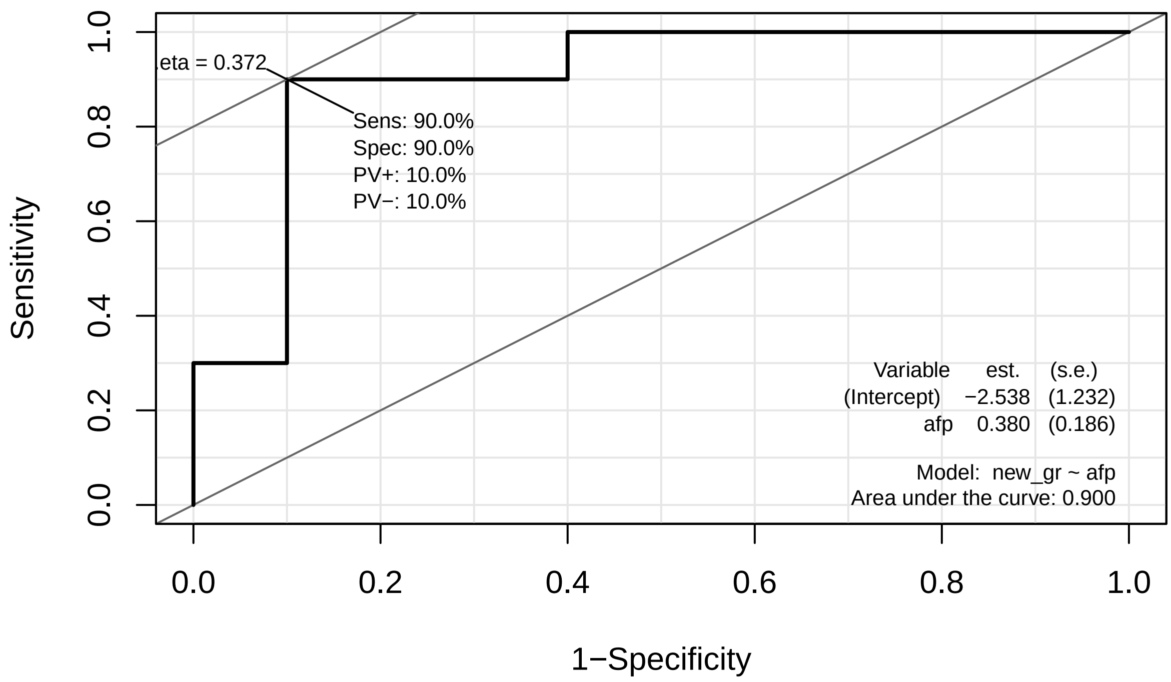
- npv

```r
afp.cutoff$npv
```

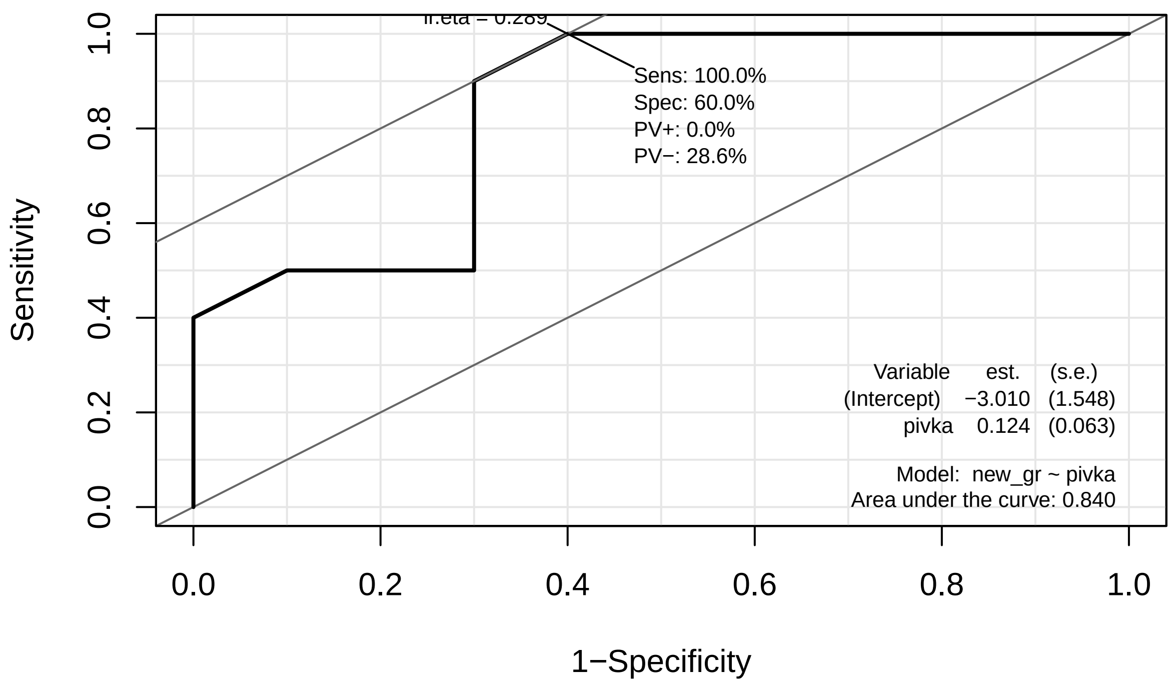
```
##           2.5%       50% 97.5%
## [1,] 0.6426948 0.8181818     1
```

3.6 Epi 패키지

```r
library(Epi)
roc.ex$new_gr<-ifelse(roc.ex$group=='HCC',1,0)
ROC(form=new_gr~afp, data=roc.ex, plot='ROC')
```

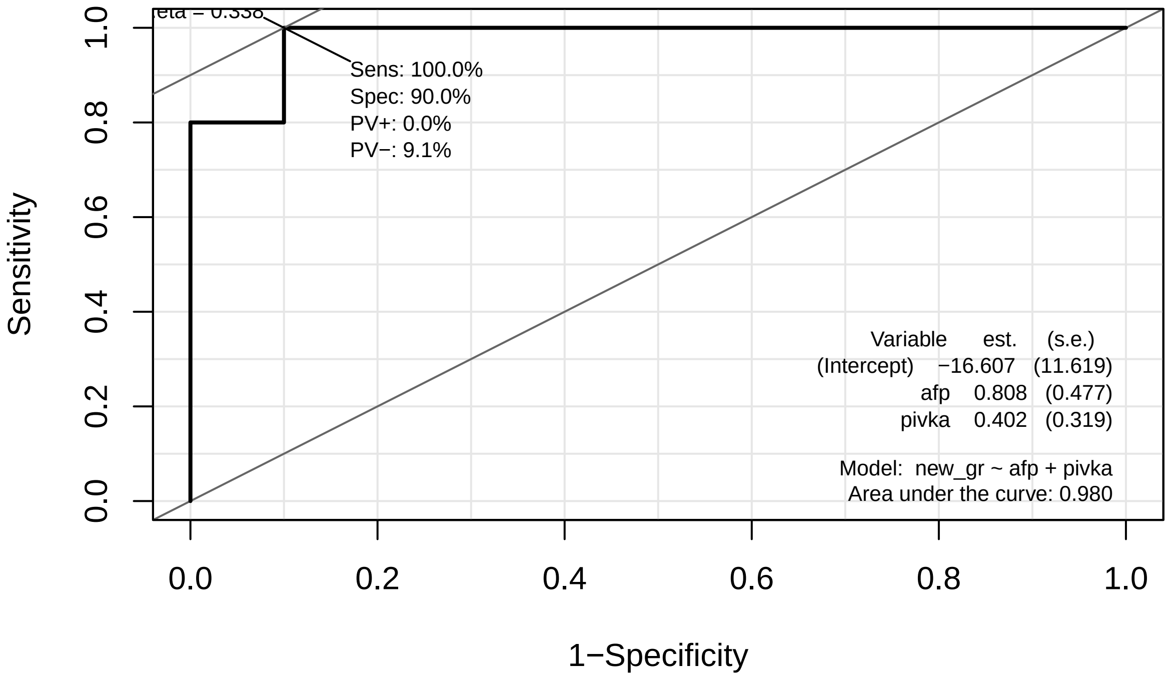


```
ROC(form=new_gr~pivka, data=roc.ex, plot='ROC')
```



1) 2개 진단검사를 포함한 ROC

```
ROC(form=new_gr~afp+pivka, data=roc.ex, plot='ROC')
```



4. 생존 분석

4.1 Time to event 분석

2) 생존함수와 위험함수

```
library(survival)
library(lubridate)
suv.dt<-read_csv('C:\\Users\\phl02\\Desktop\\P\\bio\\ch6\\Ch6_survival.csv')
suv.dt
```

```
## # A tibble: 20 x 6
##       id gender    lc start_date hcc_date     hcc
##    <dbl> <chr>  <dbl> <date>     <date>     <dbl>
##  1     1 M          1 2007-01-05 2014-07-18     1
##  2     2 F          0 2007-01-10 2016-08-25     0
##  3     3 M          1 2007-01-11 2017-06-21     1
##  4     4 M          0 2007-01-12 2012-12-17     0
##  5     5 M          1 2007-01-18 2009-07-15     1
##  6     6 M          1 2007-01-18 2013-01-11     0
##  7     7 M          1 2007-01-26 2015-01-09     1
##  8     8 F          0 2007-01-31 2017-08-24     0
##  9     9 F          1 2007-01-31 2009-03-20     0
## 10    10 M          0 2007-02-01 2017-09-12     1
## 11    11 M          0 2007-02-01 2010-09-30     1
## 12    12 M          0 2007-02-01 2010-04-23     1
## 13    13 M          0 2007-02-01 2017-08-03     0
## 14    14 M          1 2007-02-02 2011-07-07     1
## 15    15 M          1 2007-02-08 2012-11-21     1
## 16    16 M          1 2007-02-08 2011-01-21     0
## 17    17 F          1 2007-02-09 2010-07-28     0
## 18    18 F          1 2007-02-14 2009-06-17     1
## 19    19 M          0 2007-02-15 2017-08-25     0
## 20    20 F          1 2007-02-15 2012-01-31     1
```

3) 추적관찰기간 계산

```
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)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     779    1319    2140    2303    3590    3876
```

day 단위를 year 단위로

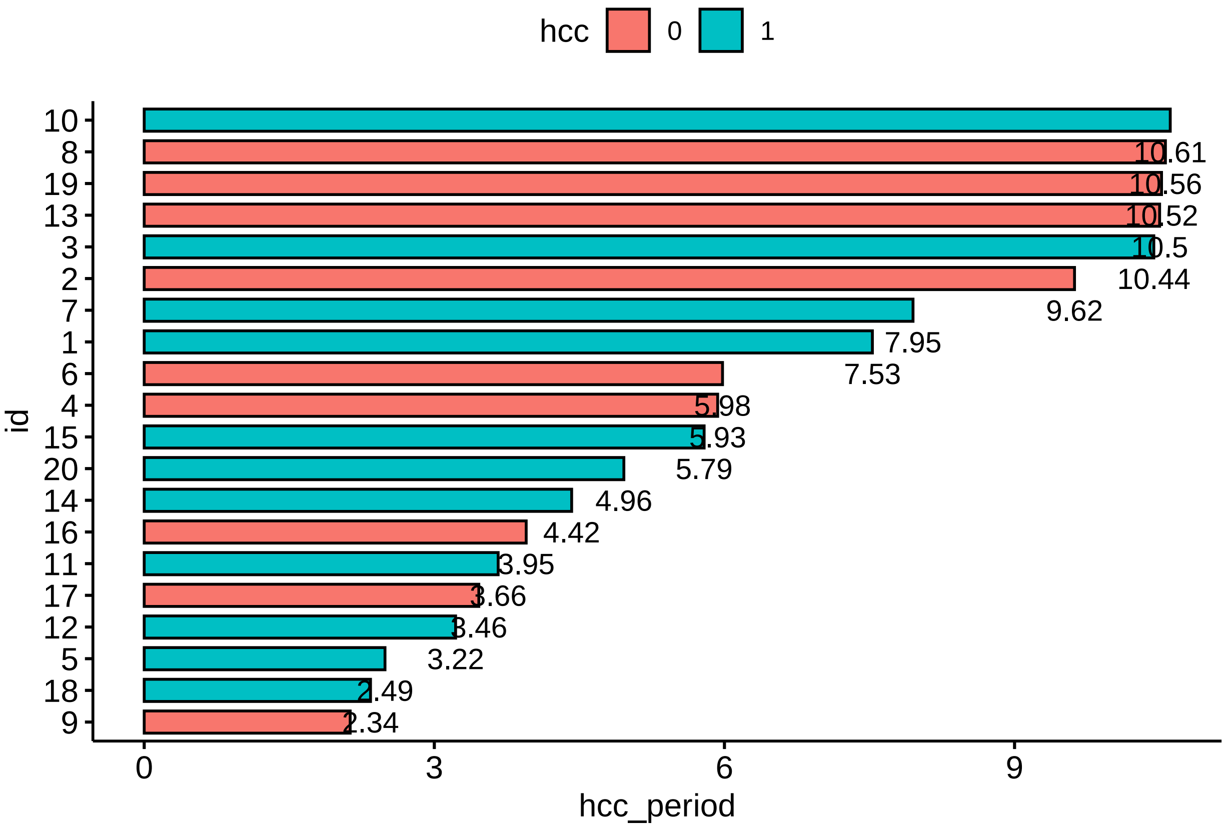
```
suv.dt$hcc_period <- suv.dt$hcc_period / 365.25
summary(suv.dt$hcc_period)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.133   3.611   5.858   6.304   9.828  10.612
```

한 번에 코드로

```
suv.dt <- suv.dt %>%
  mutate(hcc_period = hcc_date - start_date) %>%
  mutate(hcc_period = as.numeric(hcc_period)/365.25)
```

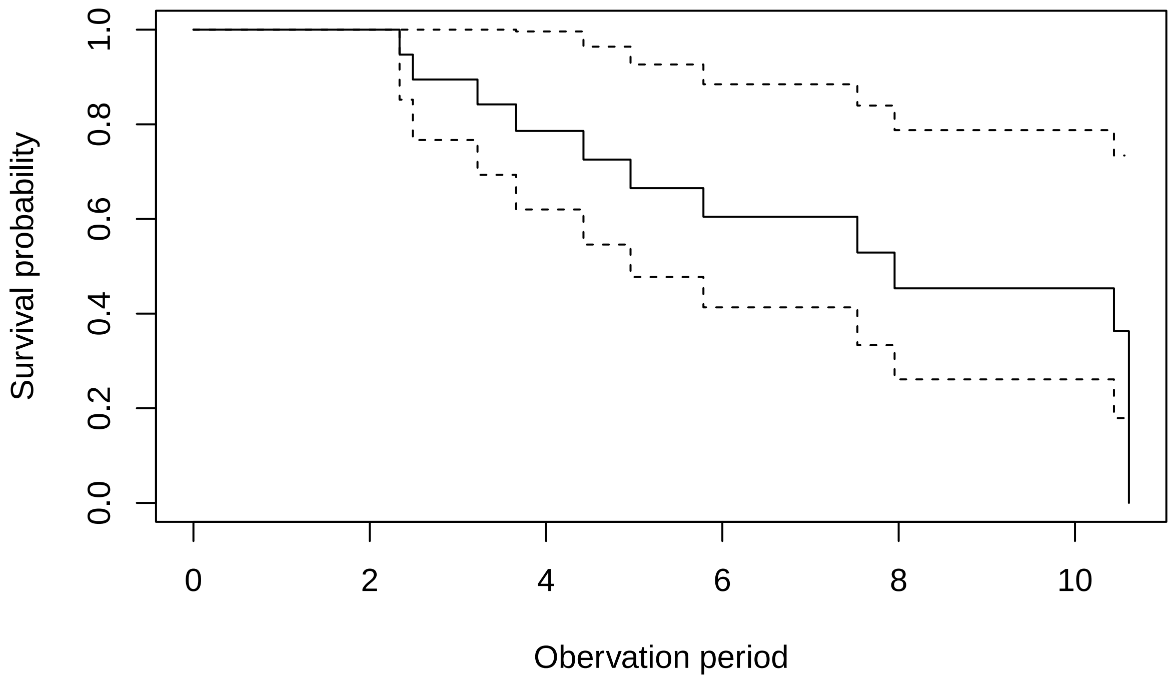
```r
library(ggpubr)

suv.dt1 <- suv.dt %>%
  mutate(hcc_period = hcc_date - start_date) %>%
  mutate(hcc_period = as.numeric(hcc_period)/365.25)%>%
  mutate(hcc_period=round(hcc_period,2))%>%
  mutate(hcc = factor(hcc))

ggbarplot(suv.dt1,y='hcc_period',x='id',fill='hcc',sort.val = 'asc',
          sort.by.groups = F, label = TRUE,lab.pos = "in",orientation = "horiz")
```



4.2 Kaplan-Meier 곡선

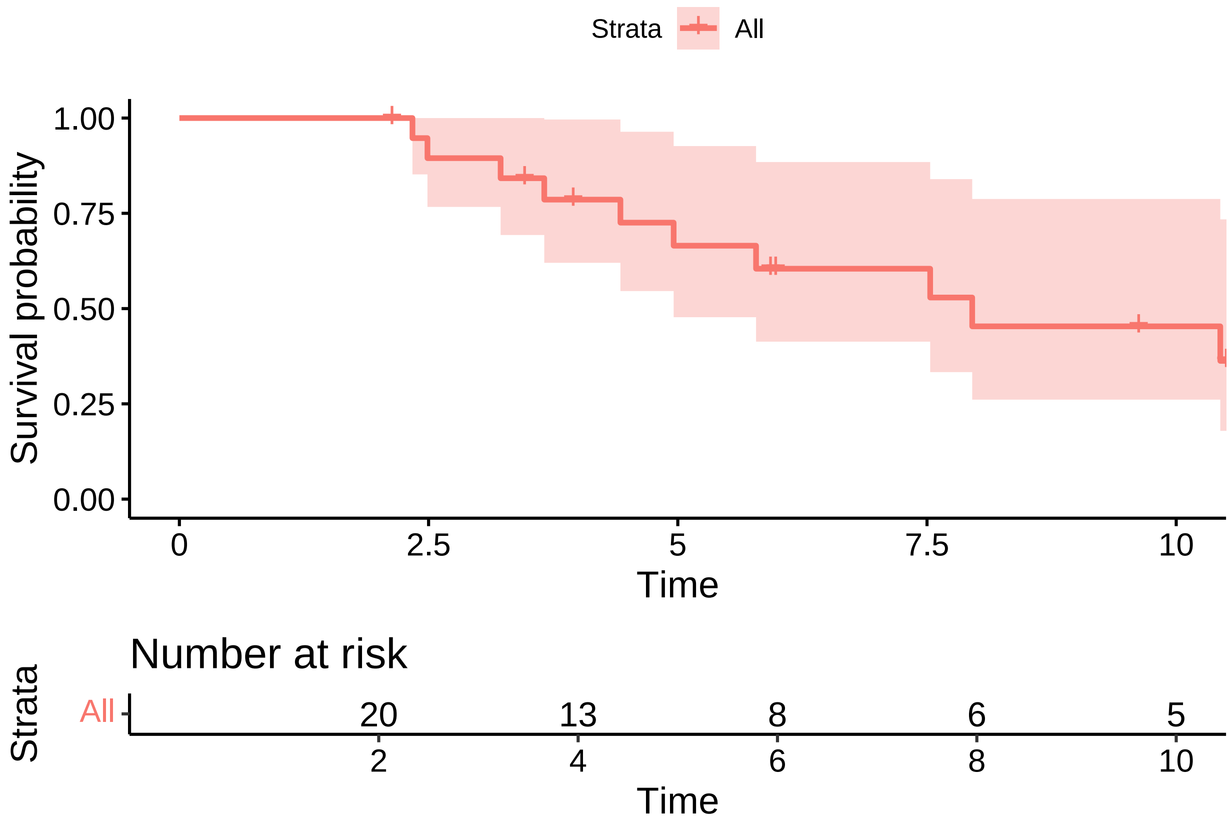
1) 생존함수 객체 만들기

```r
f1<-survfit(Surv(hcc_period, hcc)~1, data=suv.dt)
plot(f1,
     xlab='Obervation period',
     ylab='Survival probability')
```



2) survminer패키지

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



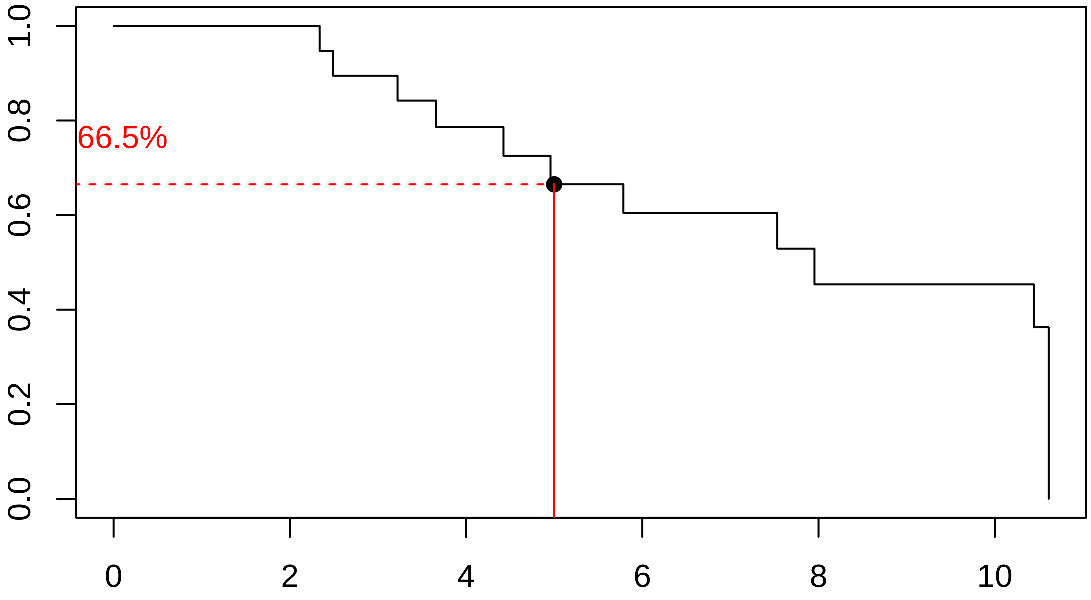
4.3 5년 생존율 계산

```
summary(f1,times=5)
```

```
## Call: survfit(formula = Surv(hcc_period, hcc) ~ 1, data = suv.dt)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     5     11       6    0.665   0.113        0.477        0.926
```

그래프로 확인

```
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')
```



4.4 Median survival 계산

```
median(suv.dt$hcc_period)
```

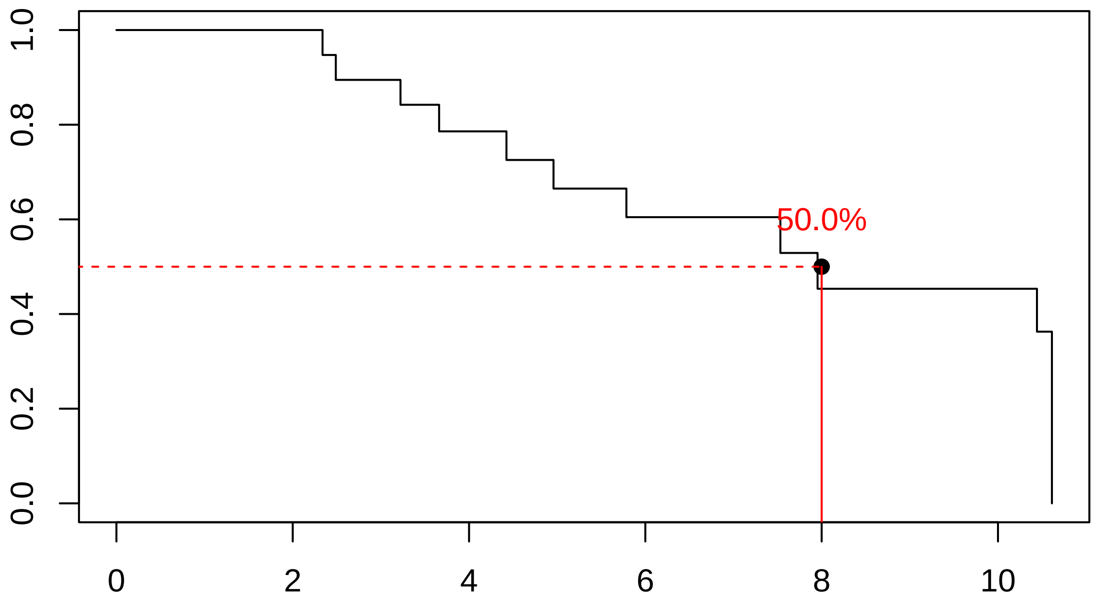
```
## [1] 5.857632
```

중도절단을 고려하지 않았기 때문에 단순한 중위 생존기간을 계산하면 틀림

```
f1
```

```
## Call: survfit(formula = Surv(hcc_period, hcc) ~ 1, data = suv.dt)
##
##        n events median 0.95LCL 0.95UCL
## [1,] 20     11   7.95    4.96      NA
```

```
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')
```



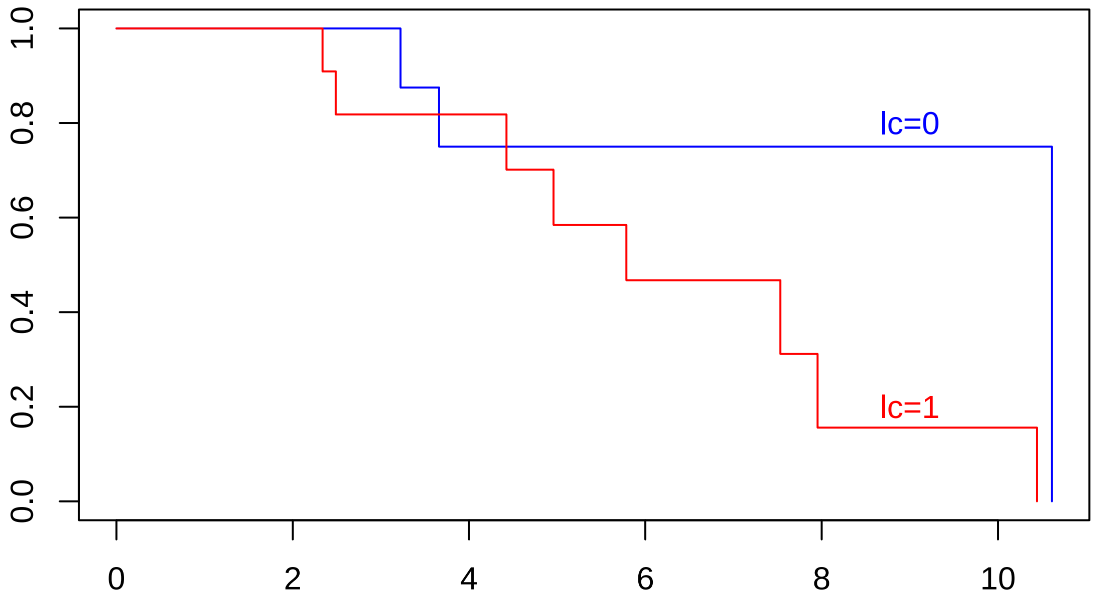
4.5 두그룹에서 생존 함수 비교

- log-rank test

```r
survdiff(Surv(hcc_period, hcc)~lc, data=suv.dt)
```

```
## Call:
## survdiff(formula = Surv(hcc_period, hcc) ~ lc, data = suv.dt)
##
##        N Observed Expected (O-E)^2/E (O-E)^2/V
## lc=0  8        3     6.45      1.84       5.1
## lc=1 12        8     4.55      2.61       5.1
##
##  Chisq= 5.1  on 1 degrees of freedom, p= 0.02
```

```r
f2<-survfit(Surv(hcc_period, hcc)~lc, data=suv.dt)
plot(f2, 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')
```



4.6 Survminer 패키지

```
suv.dt1<-read_csv('C:\\Users\\phl02\\Desktop\\P\\bio\\ch6\\Ch6_survival1.csv')
dim(suv.dt1)
```

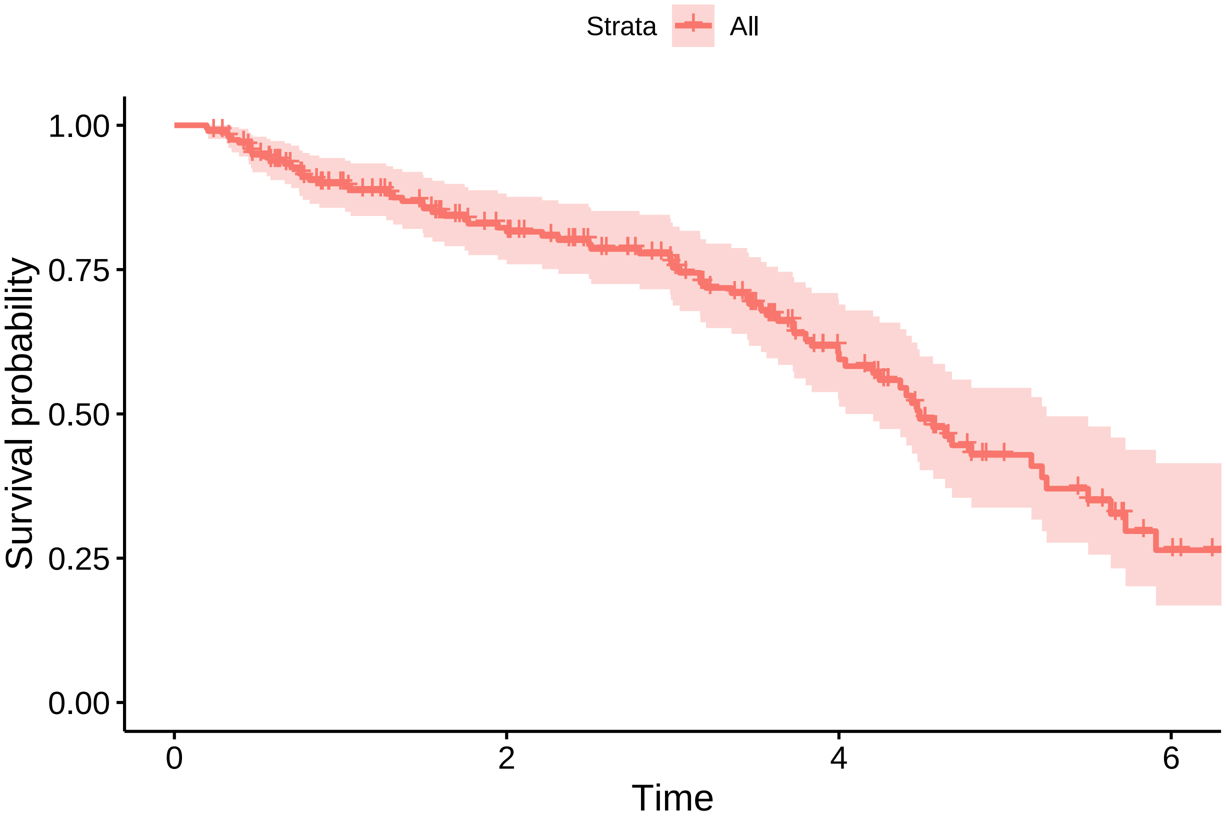
```
## [1] 200  10
```

```
head(suv.dt1)
```

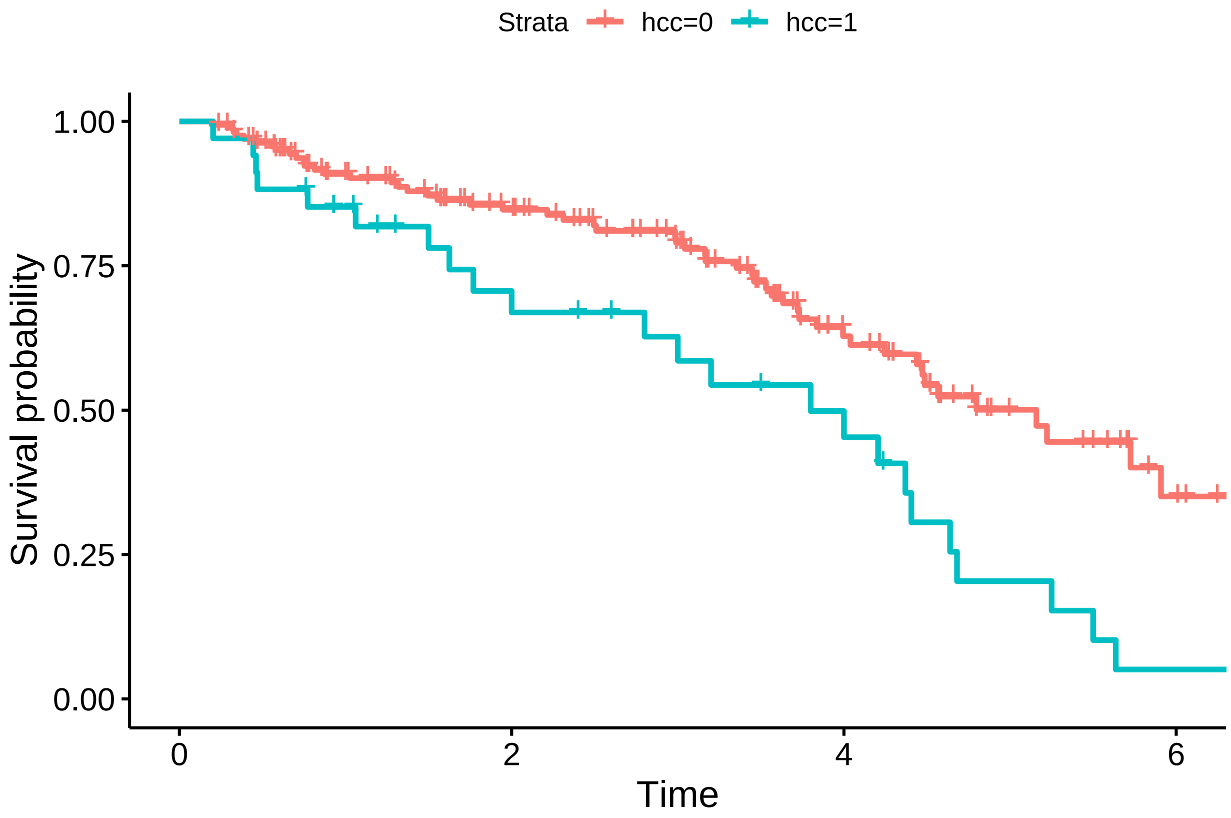
```
## # A tibble: 6 x 10
##      id gender   age    lc    dm hbeag death death_yr   hcc hcc_yr
##   <dbl> <chr>  <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl> <dbl>  <dbl>
## 1   340 F         68     1     0     1     1    3.2       1 0.0889
## 2    82 M         60     1     1     0     0    2.4       1 0.0972
## 3   434 M         50     1     0     1     0    2.6       1 0.142
## 4   373 M         50     0     1     0     1    4         1 0.161
## 5   430 M         50     1     0     1     1    0.772     1 0.172
## 6   416 M         65     1     0     0     1    6.44      1 0.25
```

1) 가장 기본 km곡선 그리기

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

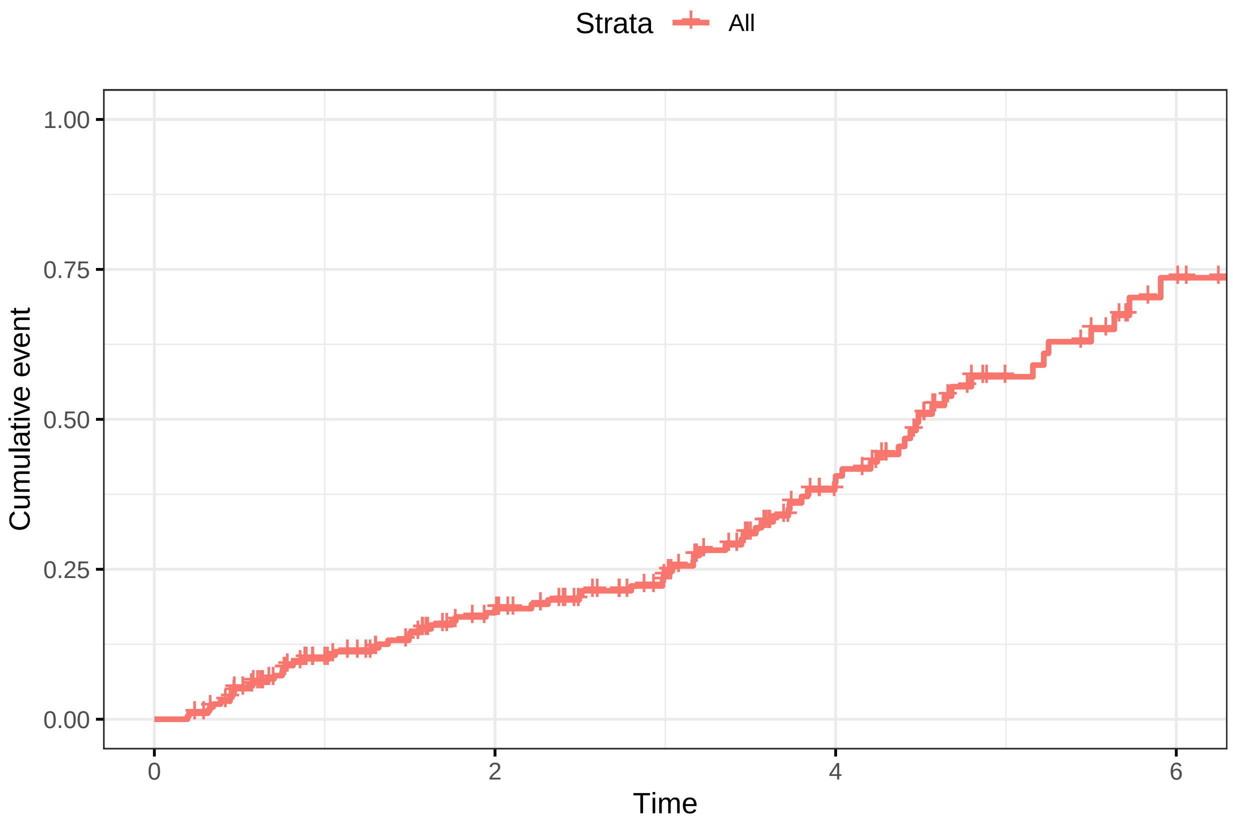


```r
f1.hcc<-survfit(Surv(death_yr, death)~hcc, data=suv.dt1)
ggsurvplot(f1.hcc)
```

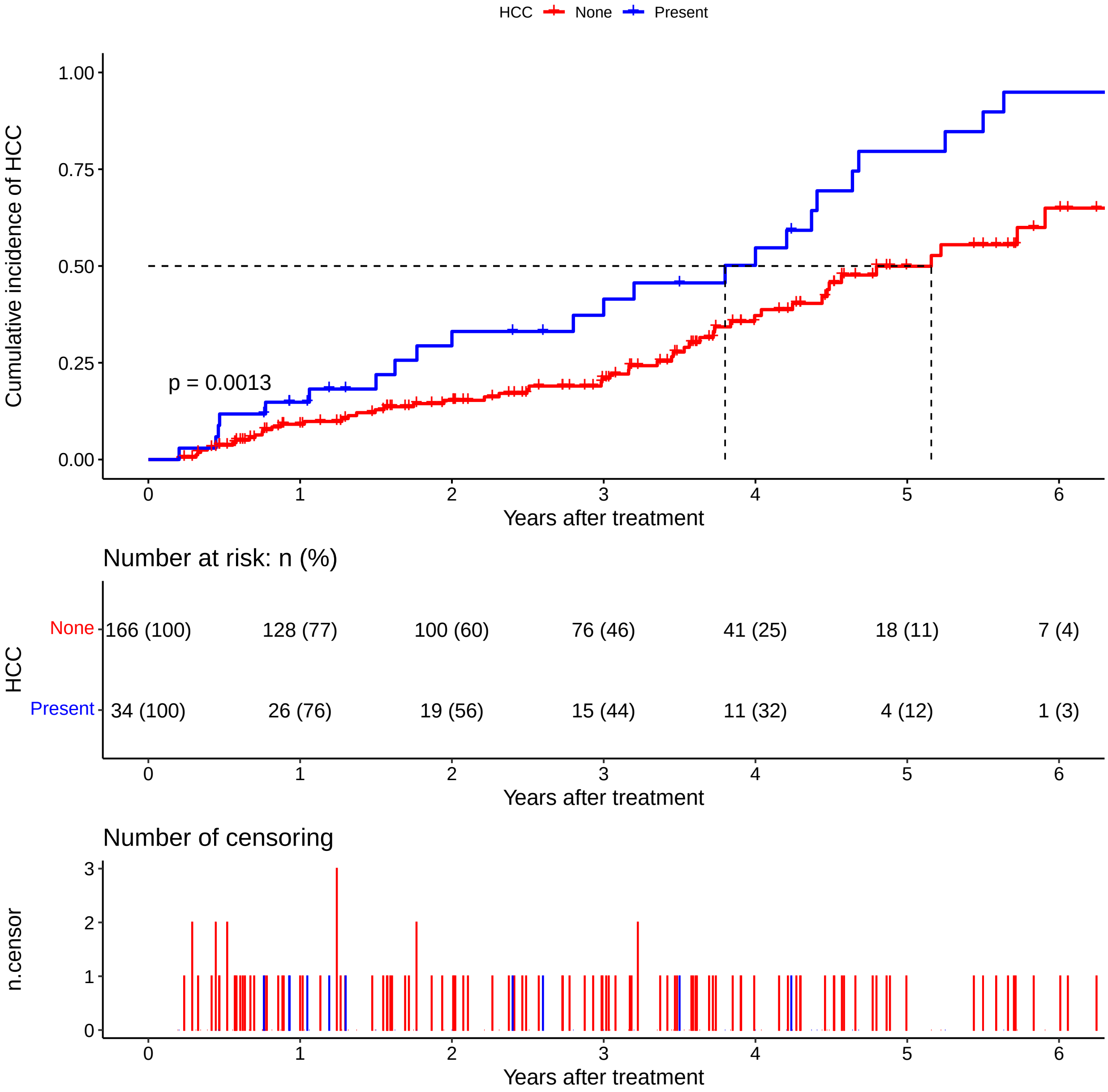


2) 누적 발생률

```
f2<-survfit(Surv(death_yr, death)~1, data=suv.dt1)
ggsurvplot(f2,
           conf.int = FALSE,
           fun = 'event',
           ylim=c(0,1),
           ggtheme=theme_bw())
```



```r
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)
```



HCC	0	1	2	3	4	5	6
None	166 (100)	128 (77)	100 (60)	76 (46)	41 (25)	18 (11)	7 (4)
Present	34 (100)	26 (76)	19 (56)	15 (44)	11 (32)	4 (12)	1 (3)

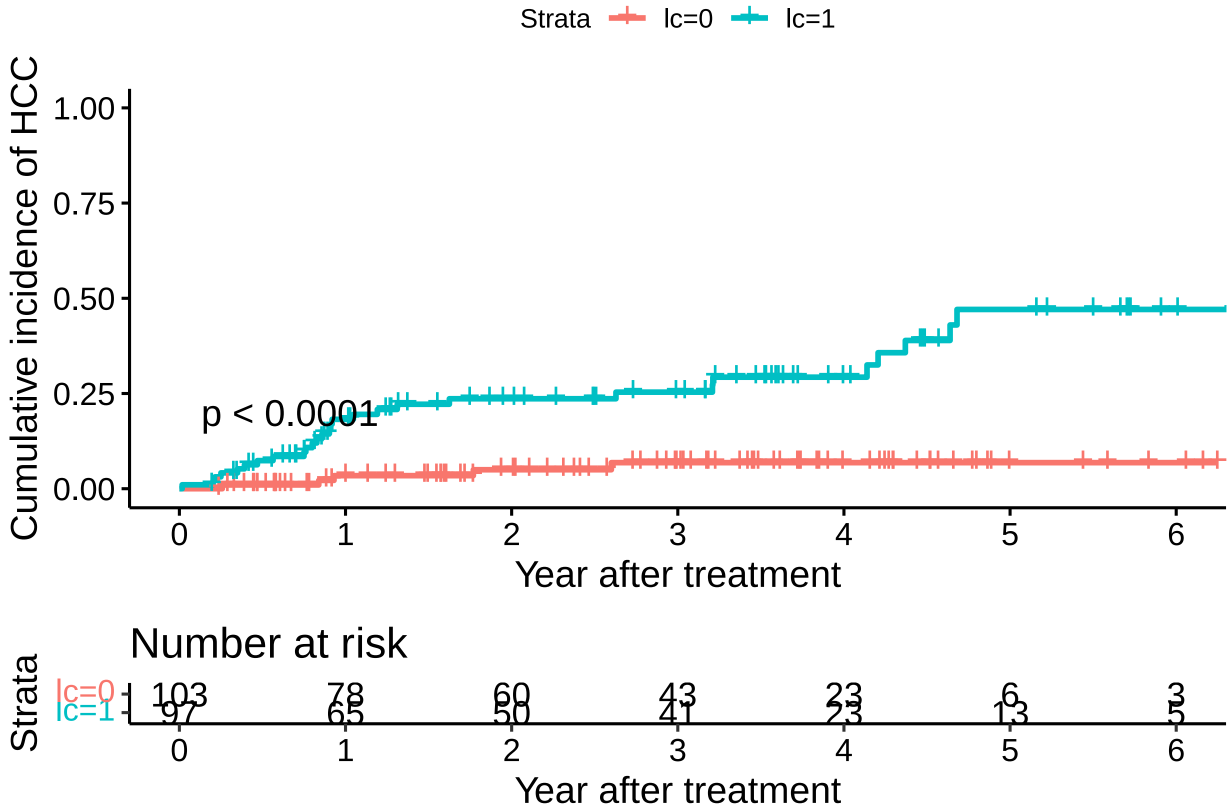
4.7 Cox 비례위험모형

1) cox model

```
library(moonBook)
suv.dt2 <-read_csv('C:\\Users\\phl02\\Desktop\\P\\bio\\ch6\\Ch6_survival2.csv')

f1.lc<-coxph(Surv(hcc_yr, hcc)~lc, data=suv.dt2)
extractHR(f1.lc)
```

```
##      HR  lcl   ucl p
## lc 6.16 2.38 15.97 0
```

```
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))
```



4.8 Cox 모형을 이용하여 단변량, 다변량 분석

1) moonBook 패키지- 단변량

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

```
##
##  mycph : perform coxph of individual expecting variables
##
##  Call: TS ~ gender + age + lc + dm + hbeag, data= suv.dt2

##            HR  lcl   ucl     p
## genderM 0.59 0.29  1.20 0.146
## age     1.08 1.05  1.12 0.000
## lc      6.16 2.38 15.97 0.000
## dm      0.73 0.22  2.38 0.598
## hbeag   1.12 0.56  2.24 0.745
```

2) gtsummary 패키지 - 단변량

```
library(gtsummary)
suv.dt2 %>%
  select(-id, -TS, -death, -death_yr) %>%
  tbl_uvregression(method=coxph,
                   y=Surv(hcc_yr, hcc),
                   exponentiate=TRUE)
```

Characteristic	N	HR	95% CI	p-value
gender	200			
F		—	—	
M		0.59	0.29, 1.20	0.15
age	200	1.08	1.05, 1.12	<0.001
lc	200	6.16	2.38, 16.0	<0.001
dm	200	0.73	0.22, 2.38	0.6
hbeag	200	1.12	0.56, 2.24	0.7

3) 다변량 분석 결과 제시

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

```
##           HR  lcl   ucl     p
## age     1.07 1.03  1.12 0.002
## genderM 1.61 0.68  3.83 0.277
## lc      3.97 1.45 10.86 0.007
## dm      0.78 0.23  2.59 0.679
## hbeag   1.48 0.71  3.10 0.295
```

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

```
## Start:  AIC=293.56
## Surv(hcc_yr, hcc) ~ age + gender + lc + dm + hbeag
##
##          Df    AIC
## - dm      1 291.75
## - hbeag   1 292.64
## - gender  1 292.78
## <none>      293.56
## - lc      1 300.48
## - age     1 302.08
##
## Step:  AIC=291.75
## Surv(hcc_yr, hcc) ~ age + gender + lc + hbeag
##
##          Df    AIC
## - hbeag   1 290.71
## - gender  1 290.89
## <none>      291.75
## - lc      1 298.53
## - age     1 300.31
##
## Step:  AIC=290.71
## Surv(hcc_yr, hcc) ~ age + gender + lc
##
##          Df    AIC
## - gender  1 289.37
## <none>      290.71
## - lc      1 297.11
## - age     1 298.49
##
## Step:  AIC=289.37
## Surv(hcc_yr, hcc) ~ age + lc
##
##        Df    AIC
## <none>    289.37
## - lc    1 295.23
## - age   1 296.64
```

```
extractHR(f1.final)
```

```
##       HR  lcl   ucl     p
## age 1.06 1.02  1.10 0.002
## lc  3.66 1.34 10.01 0.012
```

4) gtsummary 패키지- 다변량

```
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
```

Characteristic	N	HR	95% CI	p-value
age	200	1.08	1.05, 1.12	<0.001
gender	200			
F		—	—	
M		0.59	0.29, 1.20	0.15
lc	200	6.16	2.38, 16.0	<0.001
dm	200	0.73	0.22, 2.38	0.6
hbeag	200	1.12	0.56, 2.24	0.7

```
cox.multi<-coxph(Surv(hcc_yr, hcc)~age+lc, data=suv.dt2) %>%
  tbl_regression(exponentiate=TRUE)
cox.multi
```

Characteristic	HR	95% CI	p-value
age	1.06	1.02, 1.10	0.002
lc	3.66	1.34, 10.0	0.012

```
cox.table<-tbl_merge(
  tbls = list(cox.uni, cox.multi),
  tab_spanner = c("**Univariate analysis**","**Multivariable analysis**")
)
cox.table
```

Characteristic	N	HR	95% CI	p-value	HR	95% CI	p-value
age	200	1.08	1.05, 1.12	<0.001	1.06	1.02, 1.10	0.002
gender	200						
F		—	—				
M		0.59	0.29, 1.20	0.15			
lc	200	6.16	2.38, 16.0	<0.001	3.66	1.34, 10.0	0.012
dm	200	0.73	0.22, 2.38	0.6			
hbeag	200	1.12	0.56, 2.24	0.7			

4.9 Forest plot 그리기

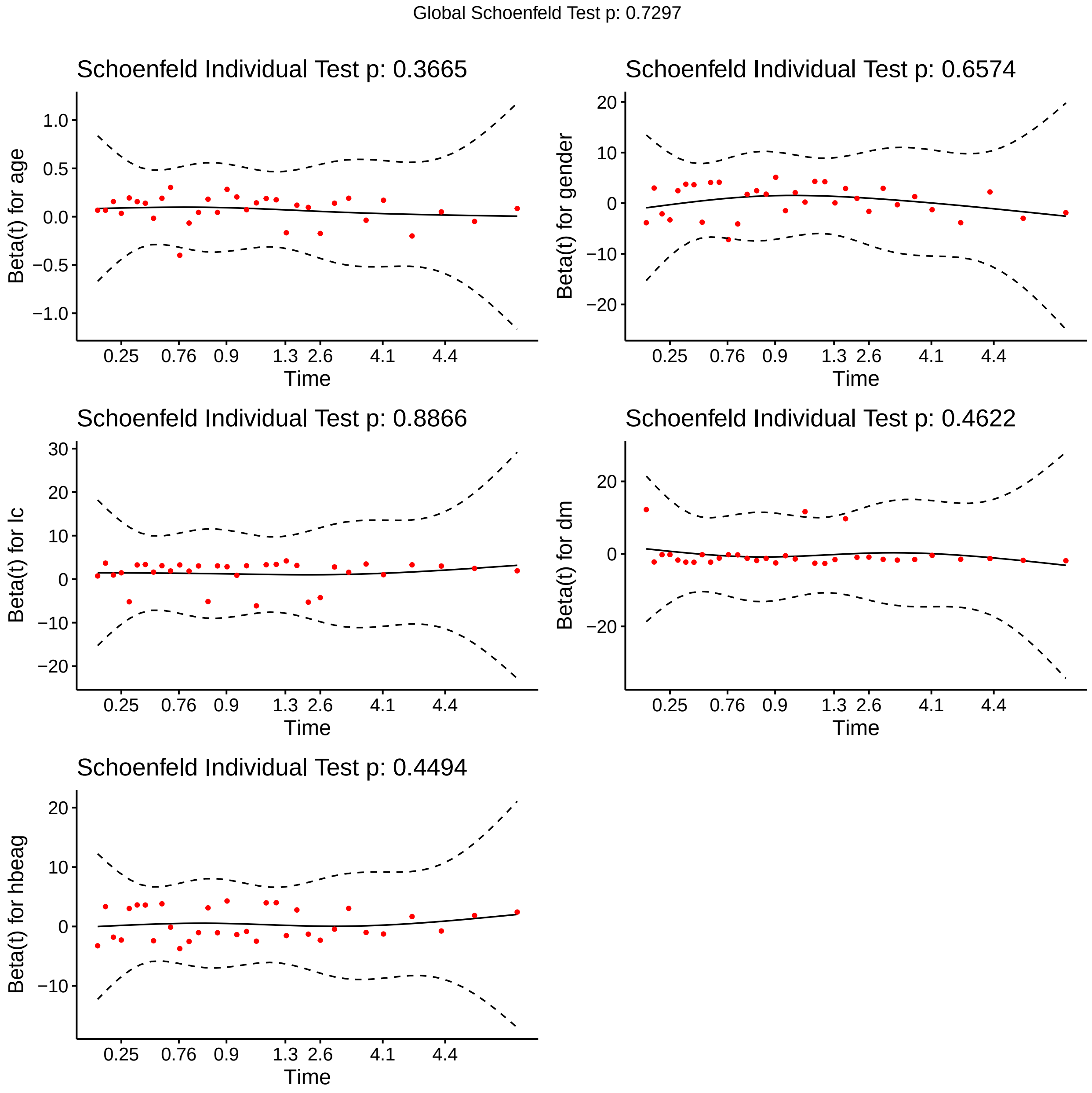
```
library(forestmodel)
f1.cox<-coxph(Surv(hcc_yr, hcc==1)~age+gender+lc+dm+hbeag, data=suv.dt2)
ggforest(f1.cox)
```

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)
```

```
##         chisq df    p
## age    0.8153  1 0.37
## gender 0.1968  1 0.66
## lc     0.0203  1 0.89
## dm     0.5405  1 0.46
## hbeag  0.5723  1 0.45
## GLOBAL 2.8071  5 0.73
```

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

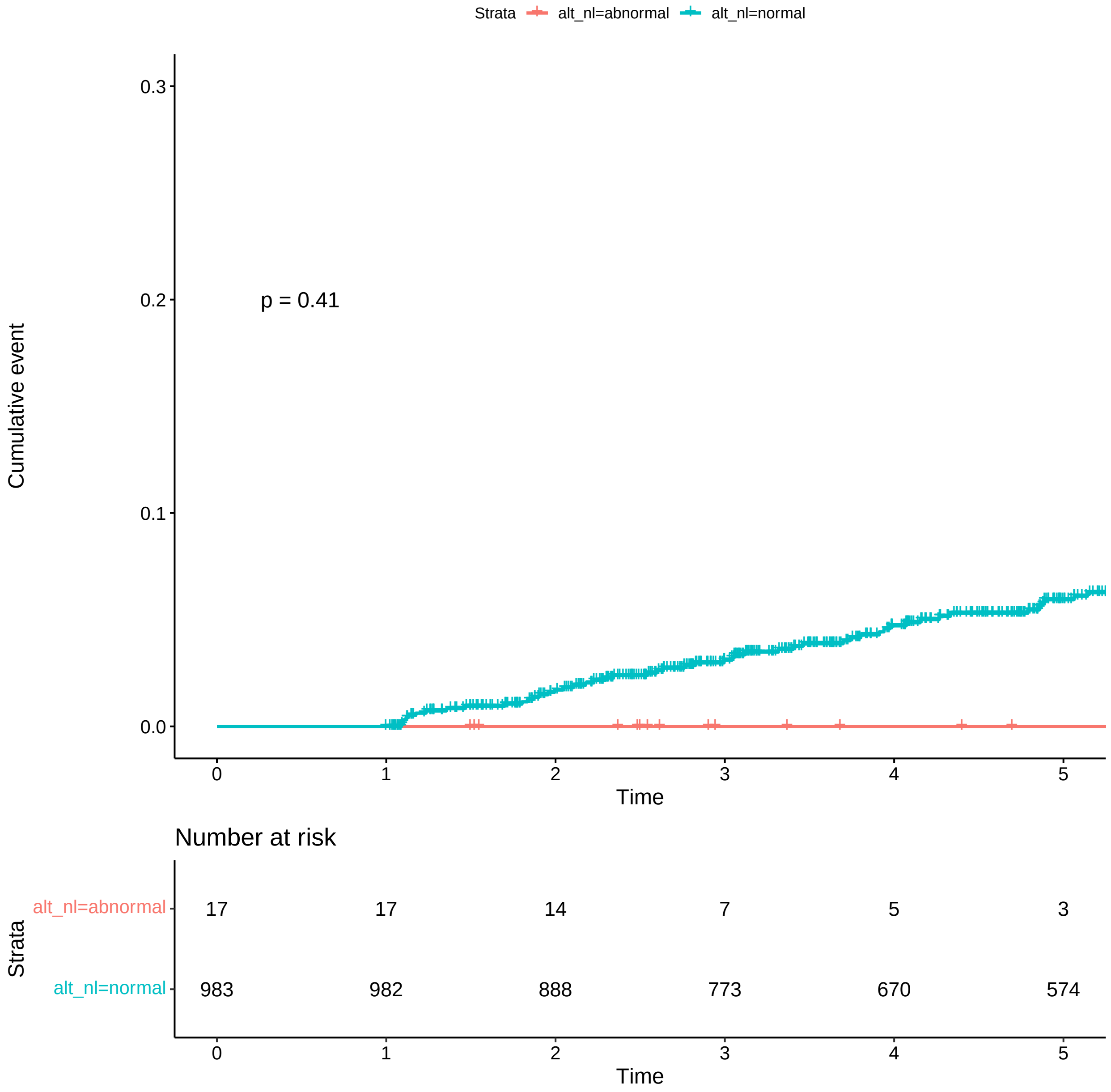


5 Time dependent Cox model

```
dt.time<-read_csv('C:\\Users\\phl02\\Desktop\\P\\bio\\ch6\\Ch6_survival3.csv')
head(dt.time)
```

```
## # A tibble: 6 x 9
##      id   age sex     alt    lc hcc_yr   hcc alt_nl   alt_duration
##   <dbl> <dbl> <chr> <dbl> <dbl>  <dbl> <dbl> <chr>           <dbl>
## 1     1  59.5 F        86     1   5.54     0 abnormal        5.54
## 2     2  37.8 M       261     0  10.8      0 normal          0.249
## 3     3  69.5 F        43     1   3.63     0 normal          0.230
## 4     4  39.7 F        97     0   5.63     0 normal          0.249
## 5     5  30.6 M       172     0   3.91     1 normal          1.02
## 6     6  38.3 M        56     0  10.7      1 normal          0.246
```

```
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)
```



Strata	0	1	2	3	4	5
alt_nl=abnormal	17	17	14	7	5	3
alt_nl=normal	983	982	888	773	670	574

```r
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')])
```

```
## # A tibble: 6 x 5
##      id hcc_yr   hcc alt_nl   alt_duration
##   <dbl>  <dbl> <dbl> <chr>           <dbl>
## 1     1   5.54     0 abnormal         5.54
## 2     2  10.8      0 normal           0.249
## 3     3   3.63     0 normal           0.230
## 4     4   5.63     0 normal           0.249
## 5     5   3.91     1 normal           1.02
## 6     6  10.7      1 normal           0.246
```

```r
head(dt.time1[,c('id','hcc_yr','hcc','alt_nl','alt_duration','tstart','tstop','HCC','ALT')],11)
```

```
##    id    hcc_yr hcc   alt_nl alt_duration    tstart      tstop HCC      ALT
## 1   1  5.535934   0 abnormal    5.5359343 0.0000000  5.5359343   0 abnormal
## 2   2 10.830938   0   normal    0.2491444 0.0000000  0.2491444   0 abnormal
## 3   2 10.830938   0   normal    0.2491444 0.2491444 10.8309377   0   normal
## 4   3  3.630390   0   normal    0.2299795 0.0000000  0.2299795   0 abnormal
## 5   3  3.630390   0   normal    0.2299795 0.2299795  3.6303901   0   normal
## 6   4  5.634497   0   normal    0.2491444 0.0000000  0.2491444   0 abnormal
## 7   4  5.634497   0   normal    0.2491444 0.2491444  5.6344969   0   normal
## 8   5  3.912389   1   normal    1.0212183 0.0000000  1.0212183   0 abnormal
## 9   5  3.912389   1   normal    1.0212183 1.0212183  3.9123888   1   normal
## 10  6 10.669405   1   normal    0.2464066 0.0000000  0.2464066   0 abnormal
## 11  6 10.669405   1   normal    0.2464066 0.2464066 10.6694045   1   normal
```

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

```
##            HR  lcl  ucl     p
## ALTnormal 0.3 0.15 0.63 0.001
```