7. Appendix

Jinglun Li; Fuyu Guo; Xinhao Li

11/5/2021

```
library(tidyverse)
library(survival)
library(ggpubr)
library(survminer)
library(glmnet)
library(car)
library(pROC)
```

```
#############################
# load data
dta <- read.csv("D:/BST_210_Heart_failure/heart_f.csv")

# select variables we will use
dta <- dplyr::select(dta,
                     "age", "sex", "anaemia",
                     "diabetes", "ejection_fraction", "smoking",
                     "platelets", "serum_creatinine", "serum_sodium",
                     "time", "DEATH_EVENT")

# rename the variables to make our work easier
names(dta) <- c("age", "sex", "anemia",
                     "dbt", "ef", "smoking",
                     "plat", "ser_crt", "ser_na",
                     "time", "death")
dta$ser_crt_ab <- if_else(dta$ser_crt <= 1.5, 0, 1)
# check sample size
dim(dta)
```

```
## [1] 299  12
```

```
# check if there is any missing value in variables
complete.cases(dta) %>% all()
```

```
## [1] TRUE
```

```
# no missing value
```

Linear, flexible/additive or other methods (LASSO, ridge)

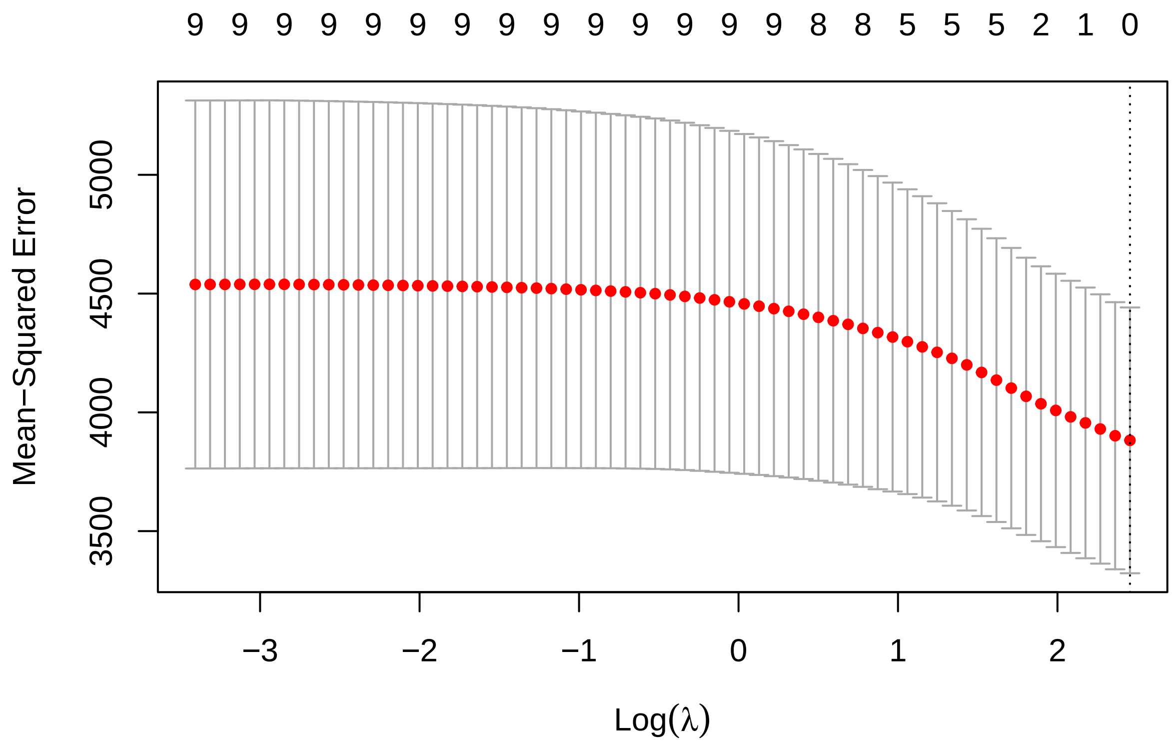
```
# choose patients who died by the end of the study
dta_line <- dplyr::filter(dta, death == 1)
# there are only 96 people died by the end

################################################
```

```r
# a crude analysis
fit_lin_1 <- lm(time~ser_crt, data = dta_line)
summary(fit_lin_1)
```

```
##
## Call:
## lm(formula = time ~ ser_crt, data = dta_line)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -66.93 -46.63 -26.23  31.36 170.12
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  69.6578    10.2774   6.778 1.06e-09 ***
## ser_crt       0.6687     4.3805   0.153    0.879
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.7 on 94 degrees of freedom
## Multiple R-squared:  0.0002478,  Adjusted R-squared:  -0.01039
## F-statistic: 0.0233 on 1 and 94 DF,  p-value: 0.879
```

```r
#################################################
# conduct a lasso regression to choose covariate sets for linear serum creatine as a continuous variabl
x <- dplyr::select(dta_line, -death, -time, -ser_crt_ab) %>%
        as.matrix()
y <- dta_line$time %>% as.vector()
cv <-  cv.glmnet(x, y, type.measure = "mse", nfolds = 4)
plot(cv)
```



The results of the lasso tells us that we should not include any covariates in the model. Only intercept is enough. This result is not helpful in guiding covariates selection, partly because of the small sample size and null association.

```
# we will adjust for age and sex based on the background knowledge
fit_lin_2 <- lm(time~ser_crt + age + sex, data = dta_line)
summary(fit_lin_2)
```

```
##
## Call:
## lm(formula = time ~ ser_crt + age + sex, data = dta_line)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -77.39 -42.70 -24.74  41.42 164.57
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 127.1234    32.9043   3.863 0.000208 ***
## ser_crt       1.1688     4.3668   0.268 0.789563
## age          -0.8914     0.4918  -1.813 0.073147 .
## sex          -0.3834    13.5115  -0.028 0.977425
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.24 on 92 degrees of freedom
## Multiple R-squared:  0.03597,    Adjusted R-squared:  0.004537
```

```
## F-statistic: 1.144 on 3 and 92 DF,  p-value: 0.3355
```

```r
###########################################################
#try use serum creatine as a binary variable

fit_lin_3 <- lm(time~ser_crt_ab, data = dta_line)
summary(fit_lin_3)
```

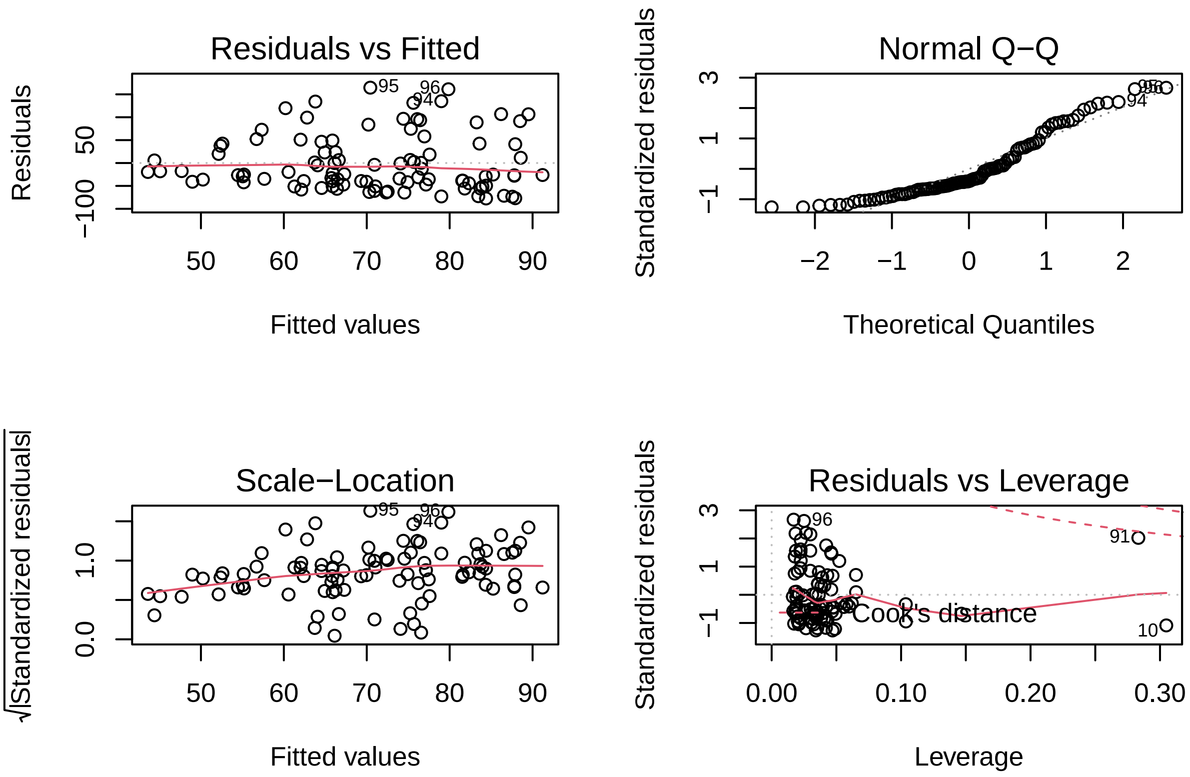
```
##
## Call:
## lm(formula = time ~ ser_crt_ab, data = dta_line)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -76.14 -49.57 -20.38  27.68 171.62
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   63.377      8.536   7.425 5.04e-11 ***
## ser_crt_ab    16.762     12.754   1.314    0.192
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.14 on 94 degrees of freedom
## Multiple R-squared:  0.01804,    Adjusted R-squared:  0.007598
## F-statistic: 1.727 on 1 and 94 DF,  p-value: 0.1919
```

```r
fit_lin_4 <- lm(time~ser_crt_ab + age + sex, data = dta_line)
summary(fit_lin_2)
```

```
##
## Call:
## lm(formula = time ~ ser_crt + age + sex, data = dta_line)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -77.39 -42.70 -24.74  41.42 164.57
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 127.1234    32.9043   3.863 0.000208 ***
## ser_crt       1.1688     4.3668   0.268 0.789563
## age          -0.8914     0.4918  -1.813 0.073147 .
## sex          -0.3834    13.5115  -0.028 0.977425
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.24 on 92 degrees of freedom
## Multiple R-squared:  0.03597,    Adjusted R-squared:  0.004537
## F-statistic: 1.144 on 3 and 92 DF,  p-value: 0.3355
```

```r
# we will do some model diagnostics with the most spares model
# residual analysis

par(mfrow = c(2,2))
plot(fit_lin_2)
```



- Variance is not equal.
- The residual is not normally distributed.
- Although there are three high influence values, they are still within the Cook's distance boundary. Thus, we conclude there is no high influence from outliers or leverage values in the model.

```
# adjusted R^2 in the model above
summary(fit_lin_2)
```

```
##
## Call:
## lm(formula = time ~ ser_crt + age + sex, data = dta_line)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -77.39 -42.70 -24.74  41.42 164.57
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 127.1234    32.9043   3.863 0.000208 ***
## ser_crt       1.1688     4.3668   0.268 0.789563
## age          -0.8914     0.4918  -1.813 0.073147 .
## sex          -0.3834    13.5115  -0.028 0.977425
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.24 on 92 degrees of freedom
## Multiple R-squared:  0.03597,    Adjusted R-squared:  0.004537
```

```
## F-statistic: 1.144 on 3 and 92 DF,  p-value: 0.3355
```

Logistic, multinomial, ordinal:

```r
# assess the death by the 30-days
dta$death_30 <- NA
dta$death_30[dta$death == 1& dta$time <= 30] <- "Yes"
dta$death_30[(dta$death == 1& dta$time > 30) |
               (dta$death == 0& dta$time > 30)] <- "No"
dta$death_30[dta$death == 0& dta$time <= 30] <- "Censored"
table(dta$death_30)
```

```
##
## Censored       No      Yes
##        5      259       35
```

```r
# there are 5 patients censored at the 30 days
# we will just drop them

dta_log <- dplyr::filter(dta, death_30 != "Censored")
dta_log$death_30 <- if_else(dta_log$death_30 == "Yes", 1, 0)
############################################################
# a crude logistic model
fit_log_1 <- glm(death_30~ser_crt, family = "binomial", data = dta_log)
summary(fit_log_1)
```

```
##
## Call:
## glm(formula = death_30 ~ ser_crt, family = "binomial", data = dta_log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5005  -0.4824  -0.4579  -0.4422   2.1788
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.5695     0.2829  -9.083  < 2e-16 ***
## ser_crt       0.3670     0.1300   2.822  0.00477 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 214.63  on 293  degrees of freedom
## Residual deviance: 206.92  on 292  degrees of freedom
## AIC: 210.92
##
## Number of Fisher Scoring iterations: 4
```

```r
fit_log_1$coefficients %>% exp
```

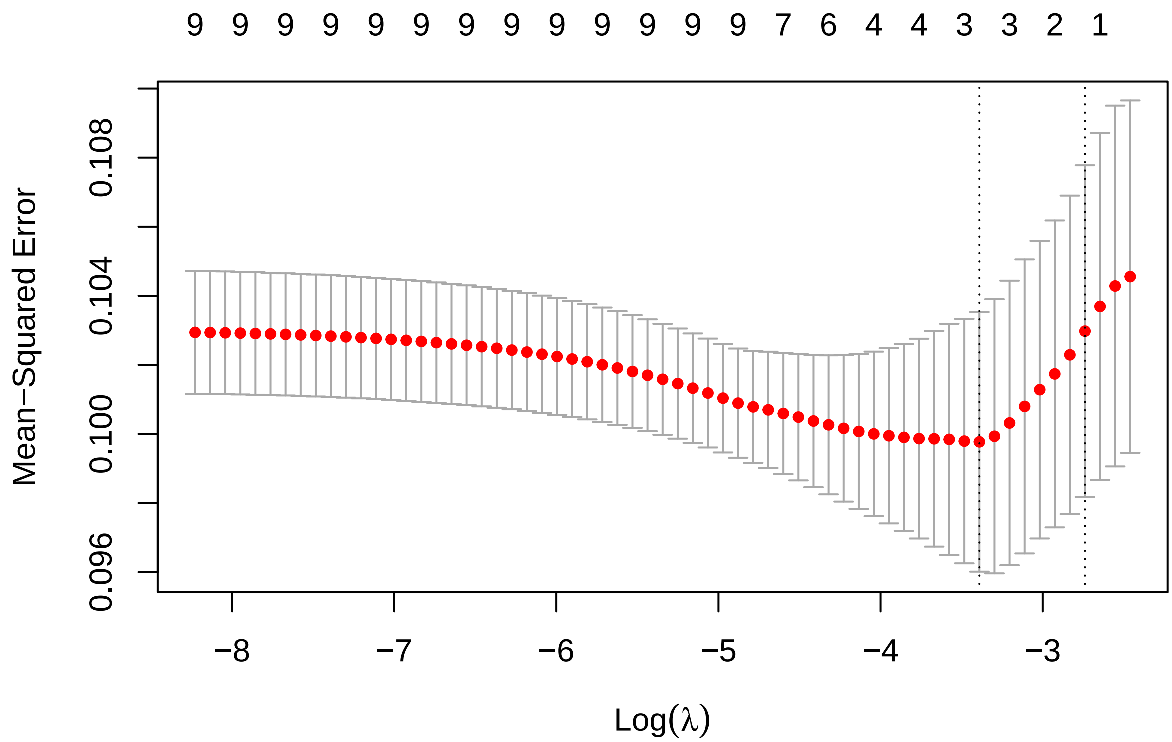
```
## (Intercept)     ser_crt
##  0.07657599  1.44340585
```

```r
confint(fit_log_1) %>% exp # check 95% CI
```

```
## Waiting for profiling to be done...

##                   2.5 %    97.5 %
## (Intercept) 0.04271699 0.1305243
## ser_crt     1.12006891 1.8922657
```

```
# use lasso to help determine the covariate sets
x <- dplyr::select(dta_log, -death, -time, -death_30, -ser_crt_ab) %>%
        as.matrix()
y <- dta_log$death_30 %>% as.vector()
cv <-  cv.glmnet(x, y, type.measure = "mse", nfolds = 4)
plot(cv)
```




```
cv$lambda.min
```

```
## [1] 0.03369666
```

```
cv$glmnet.fit
```

```
##
## Call:  glmnet(x = x, y = y)
##
##    Df  %Dev   Lambda
## 1   0  0.00 0.085430
## 2   1  1.18 0.077840
## 3   1  2.16 0.070930
## 4   1  2.98 0.064630
## 5   1  3.65 0.058890
## 6   2  4.61 0.053650
```

```
## 7   3  5.58 0.048890
## 8   3  6.50 0.044550
## 9   3  7.27 0.040590
## 10  3  7.91 0.036980
## 11  3  8.43 0.033700
## 12  3  8.87 0.030700
## 13  4  9.28 0.027980
## 14  4  9.69 0.025490
## 15  4 10.03 0.023230
## 16  4 10.31 0.021160
## 17  4 10.55 0.019280
## 18  4 10.74 0.017570
## 19  5 10.92 0.016010
## 20  6 11.09 0.014590
## 21  6 11.26 0.013290
## 22  6 11.39 0.012110
## 23  6 11.50 0.011030
## 24  7 11.60 0.010050
## 25  7 11.69 0.009161
## 26  9 11.80 0.008347
## 27  9 11.89 0.007605
## 28  9 11.98 0.006930
## 29  9 12.04 0.006314
## 30  9 12.10 0.005753
## 31  9 12.15 0.005242
## 32  9 12.18 0.004776
## 33  9 12.22 0.004352
## 34  9 12.24 0.003965
## 35  9 12.27 0.003613
## 36  9 12.28 0.003292
## 37  9 12.30 0.003000
## 38  9 12.31 0.002733
## 39  9 12.32 0.002490
## 40  9 12.33 0.002269
## 41  9 12.34 0.002068
## 42  9 12.34 0.001884
## 43  9 12.35 0.001717
## 44  9 12.35 0.001564
## 45  9 12.36 0.001425
## 46  9 12.36 0.001299
## 47  9 12.36 0.001183
## 48  9 12.36 0.001078
## 49  9 12.37 0.000982
## 50  9 12.37 0.000895
## 51  9 12.37 0.000816
## 52  9 12.37 0.000743
## 53  9 12.37 0.000677
## 54  9 12.37 0.000617
## 55  9 12.37 0.000562
## 56  9 12.37 0.000512
## 57  9 12.37 0.000467
## 58  9 12.37 0.000425
## 59  9 12.37 0.000387
## 60  9 12.37 0.000353
```

```
## 61   9 12.37 0.000322
## 62   9 12.37 0.000293
## 63   9 12.37 0.000267
```

```
coef(cv, s = "lambda.min")
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept)  0.299914132
## age          0.004033112
## sex          .
## anemia       .
## dbt          .
## ef           .
## smoking      .
## plat         .
## ser_crt      0.018167964
## ser_na      -0.003308001
```

```
best_cov <- c("age", "anemia", "ef", "plat", "ser_crt", "ser_na")
coef(cv, s = 0.010050)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)  7.850451e-01
## age          5.613910e-03
## sex          3.373783e-04
## anemia       3.544657e-02
## dbt          .
## ef          -5.075034e-04
## smoking      .
## plat        -3.718755e-08
## ser_crt      3.344088e-02
## ser_na      -7.619702e-03
```

```
include_cov <- c("sex", "age", "anemia", "ef", "plat", "ser_crt", "ser_na")
coef(cv, s = 0.016010)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)  0.6699599884
## age          0.0052104089
## sex          .
## anemia       0.0242794930
## dbt          .
## ef          -0.0000444681
## smoking      .
## plat         .
## ser_crt      0.0296350188
## ser_na      -0.0067224731
```

```
exclude_cov <- c("age", "anemia", "ef",  "ser_crt", "ser_na")
```

According to the lasso results, we will perform three logitstic regressions and compare their results.

```
log_fun <- function(set){
        x_matrix <- dta_log[set,]
```

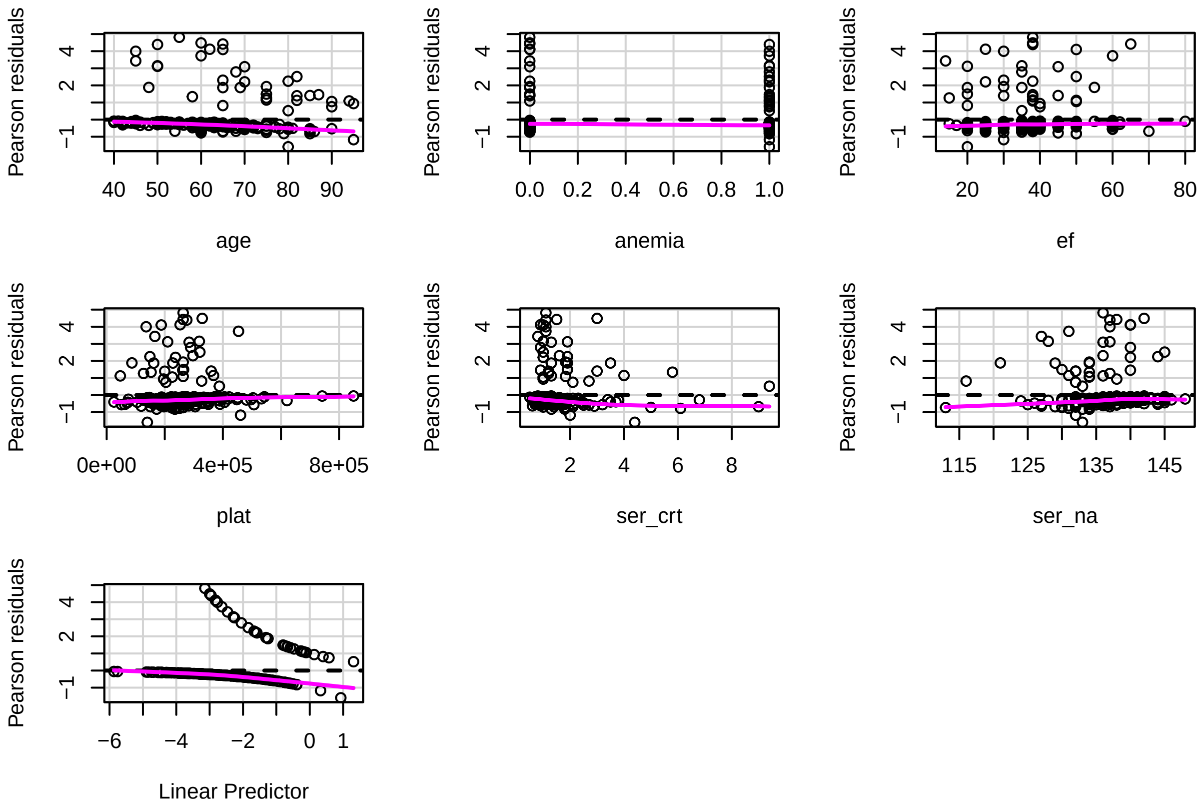
```
        y <- dta_log$death_30
        fit <- glm(y~x, family = "binomial")
        OR <- coef(fit)["xser_crt"] %>% exp()
        CI <- confint(fit)["xser_crt",] %>% exp()
        aic <- fit$aic
        return(list(OR, CI, aic))
}
lapply(list(best_cov, exclude_cov, include_cov), log_fun)
```

```
## Waiting for profiling to be done...
## Waiting for profiling to be done...
## Waiting for profiling to be done...
```

```
## [[1]]
## [[1]][[1]]
## xser_crt
## 1.324913
##
## [[1]][[2]]
##     2.5 %    97.5 %
## 0.9826464 1.7762613
##
## [[1]][[3]]
## [1] 198.6465
##
##
## [[2]]
## [[2]][[1]]
## xser_crt
## 1.324913
##
## [[2]][[2]]
##     2.5 %    97.5 %
## 0.9826464 1.7762613
##
## [[2]][[3]]
## [1] 198.6465
##
##
## [[3]]
## [[3]][[1]]
## xser_crt
## 1.324913
##
## [[3]][[2]]
##     2.5 %    97.5 %
## 0.9826464 1.7762613
##
## [[3]][[3]]
## [1] 198.6465
```

Diagnostics for the model

```
log_fit <- glm(death_30~age+anemia+ef+plat+ser_crt+ser_na, family = "binomial", data = dta_log)
residualPlots(log_fit)
```




```
##           Test stat Pr(>|Test stat|)
## age          2.6482           0.1037
## anemia       0.0000           1.0000
## ef           0.3016           0.5829
## plat         0.1064           0.7443
## ser_crt      0.1761           0.6748
## ser_na       0.7059           0.4008
```

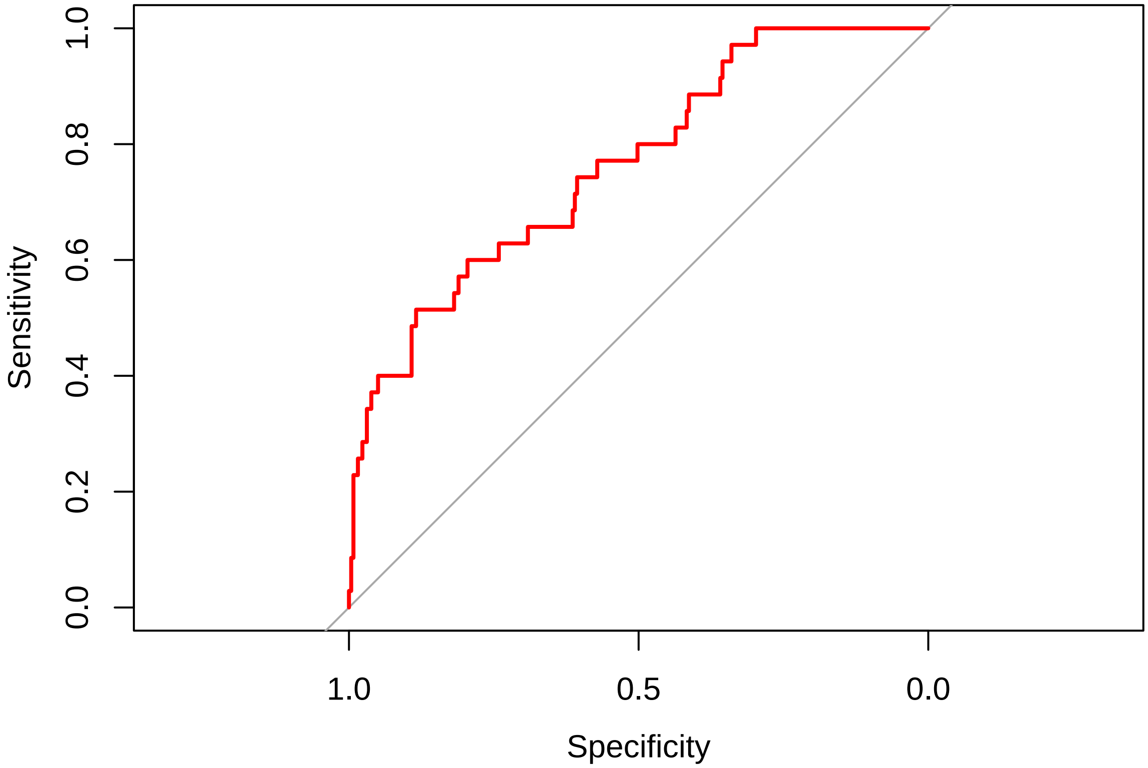
Discrimination

```
library(pROC)
predprob <- predict(log_fit,type=c("response"))
roccurve <- roc(dta_log$death_30 ~ predprob)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(roccurve,col="red")
```




```
auc(roccurve)
```

```
## Area under the curve: 0.7629
```

Poisson Regression

We will collapse the individual observations into total groups.

```
dta$age_65 <- if_else(dta$age >= 65, 1, 0)
dta$ef_group <- case_when(dta$ef <= 30 ~ "low",
                          dta$ef >30 & dta$ef < 45 ~ "normal",
                          dta$ef >=45 ~ "high")

dta$plat_group <- if_else(dta$plat >= median(dta$plat), "high", "low")
dta$ser_na_group <- if_else(dta$ser_na >= median(dta$ser_na), "high", "low")

dta_pos <- dta %>% group_by(age_65, sex, ef_group, plat_group, anemia, ser_na_group, ser_crt_ab) %>%
        summarise(death = sum(death),
                  time = sum(time))
```

```
## `summarise()` has grouped output by 'age_65', 'sex', 'ef_group', 'plat_group', 'anemia', 'ser_na_grou
```

```
names(dta_pos)
```

```
## [1] "age_65"       "sex"          "ef_group"     "plat_group"   "anemia"
## [6] "ser_na_group" "ser_crt_ab"   "death"        "time"
```

```
pois_fit <- glm(death~ser_crt_ab  + sex+ age_65+ef_group+plat_group+anemia + ser_na_group, offset = log
dim(dta_pos)
```

```
## [1] 126   9
```

```
pois_fit %>% summary()
```

```
##
## Call:
## glm(formula = death ~ ser_crt_ab + sex + age_65 + ef_group +
##     plat_group + anemia + ser_na_group, family = poisson(), data = dta_pos,
##     offset = log(time))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1568  -0.9484  -0.4188   0.7748   2.5915
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -6.9959     0.3348 -20.898  < 2e-16 ***
## ser_crt_ab         0.9566     0.2210   4.328  1.5e-05 ***
## sex               -0.1124     0.2156  -0.521 0.602233
## age_65             0.7562     0.2058   3.675 0.000238 ***
## ef_grouplow        0.6838     0.2849   2.400 0.016377 *
## ef_groupnormal    -0.2848     0.3050  -0.934 0.350550
## plat_grouplow     -0.1605     0.2080  -0.771 0.440472
## anemia             0.5014     0.2121   2.364 0.018092 *
## ser_na_grouplow    0.2712     0.2311   1.174 0.240547
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 230.62  on 125  degrees of freedom
## Residual deviance: 159.13  on 117  degrees of freedom
## AIC: 329.94
##
## Number of Fisher Scoring iterations: 6
```

```
coef(pois_fit) %>% exp()
```

```
##     (Intercept)      ser_crt_ab             sex          age_65     ef_grouplow
##    0.0009156486    2.6028034324    0.8937042734    2.1301637981    1.9813948239
##  ef_groupnormal   plat_grouplow          anemia ser_na_grouplow
##    0.7521940404    0.8517420535    1.6511077859    1.3114984728
```

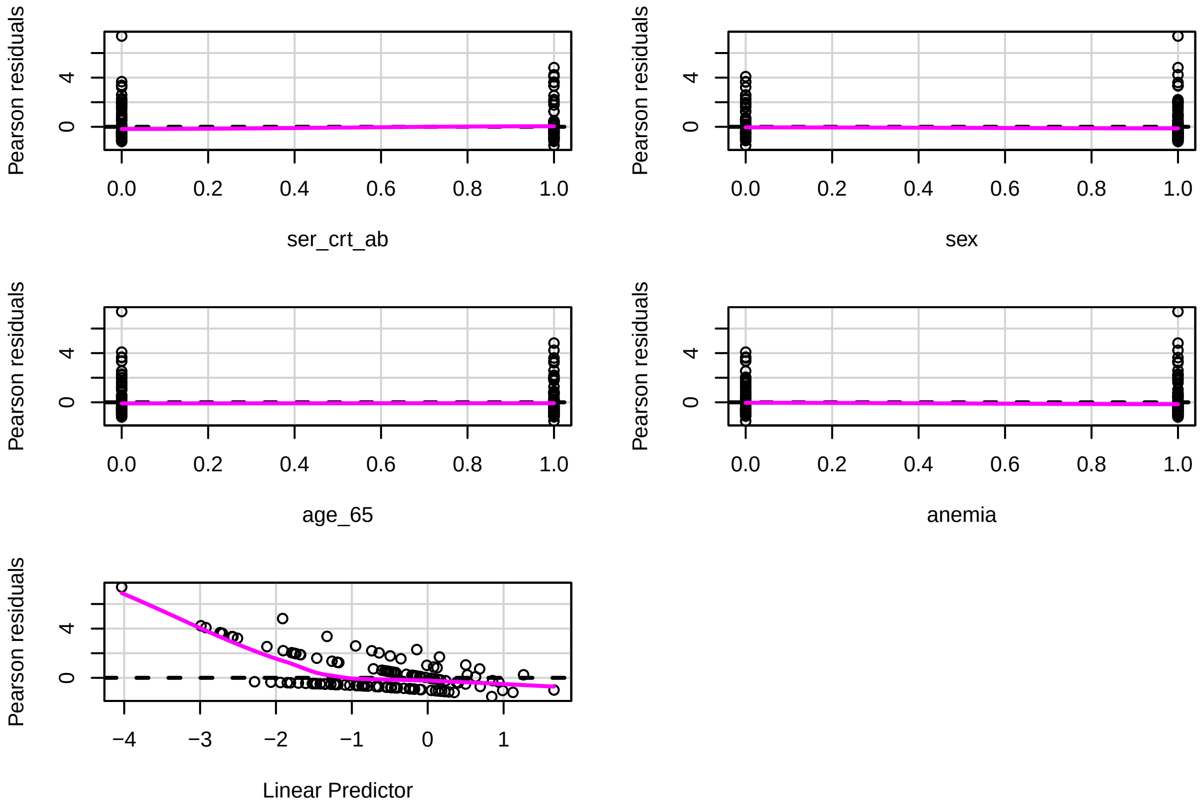
```
confint(pois_fit) %>% exp()
```

```
## Waiting for profiling to be done...
```

```
##                        2.5 %      97.5 %
## (Intercept)     0.0004611462 0.001716743
## ser_crt_ab      1.6823610137 4.009301799
## sex             0.5897161610 1.377366846
## age_65          1.4227862654 3.196016045
## ef_grouplow     1.1520367551 3.539387224
## ef_groupnormal  0.4157020757 1.386028096
## plat_grouplow   0.5658328669 1.282171634
```

```
## anemia           1.0869053000 2.502671572
## ser_na_grouplow 0.8357405639 2.072758933
```

```
residualPlots(pois_fit)
```




```
##              Test stat Pr(>|Test stat|)
## ser_crt_ab           0                1
## sex                  0                1
## age_65               0                1
## anemia               0                1
```

For a quasi-Poisson

```
quai_pois_fit <- glm(death~ser_crt_ab  + sex+ age_65+ef_group+plat_group+anemia + ser_na_group, offset =
quai_pois_fit %>% summary()
```

```
##
## Call:
## glm(formula = death ~ ser_crt_ab + sex + age_65 + ef_group +
##     plat_group + anemia + ser_na_group, family = quasipoisson(),
##     data = dta_pos, offset = log(time))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1568  -0.9484  -0.4188   0.7748   2.5915
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)      -6.9959     0.5435 -12.872  < 2e-16 ***
## ser_crt_ab        0.9566     0.3588   2.666  0.00876 **
## sex              -0.1124     0.3501  -0.321  0.74877
## age_65            0.7562     0.3341   2.263  0.02546 *
## ef_grouplow       0.6838     0.4625   1.478  0.14197
## ef_groupnormal   -0.2848     0.4953  -0.575  0.56641
## plat_grouplow    -0.1605     0.3377  -0.475  0.63559
## anemia            0.5014     0.3444   1.456  0.14810
## ser_na_grouplow   0.2712     0.3751   0.723  0.47121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 2.636011)
##
##     Null deviance: 230.62  on 125  degrees of freedom
## Residual deviance: 159.13  on 117  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

```
coef(quai_pois_fit) %>% exp()
```

```
##     (Intercept)      ser_crt_ab             sex          age_65     ef_grouplow
##    0.0009156486    2.6028034324    0.8937042734    2.1301637981    1.9813948239
##  ef_groupnormal   plat_grouplow          anemia ser_na_grouplow
##    0.7521940404    0.8517420535    1.6511077859    1.3114984728
```

```
confint(quai_pois_fit) %>% exp()
```

```
## Waiting for profiling to be done...
```

```
##                        2.5 %      97.5 %
## (Intercept)     0.0002910743 0.002474161
## ser_crt_ab      1.2749822700 5.251777025
## sex             0.4571383160 1.826980824
## age_65          1.1035485975 4.135106963
## ef_grouplow     0.8316029813 5.224560757
## ef_groupnormal  0.2866752941 2.071872562
## plat_grouplow   0.4369758856 1.660372661
## anemia          0.8337866477 3.250599757
## ser_na_grouplow 0.6312388771 2.777590426
```