Diagnostics

Zihao Guo 931278

2022-09-15

Implementation Diagnostics for the Cox Model

```
setwd("C:/Users/23690/Desktop/LivSim-Codes-master")
```

```
library(survival)
library("readxl")
library("ggplot2")
library("survminer")
library("ggfortify")
library(grid)
library(gridExtra)
post_trans <- read_excel("SimData_Transplant_Ver2.0.xls")
post_trans
```

```
## # A tibble: 4,000 x 15
##    Gender Blood~1 Inpt_~2 Donor~3 Donor~4 Donor~5 Waitt~6 Age_Tx Donor~7 Trans~8
##    <chr>  <chr>   <chr>   <chr>   <chr>   <chr>     <dbl>  <dbl>   <dbl>   <dbl>
##  1 Male   O       inpt    Male    B       No            1   54.3    45.4    24.7
##  2 Male   A       home    Male    O       No          157   59.8    60.5    28.9
##  3 Female A       home    Male    A       No           49   63.2    31.0    26.9
##  4 Male   O       inpt    Female  B       No           81   28.8    41.1    27.4
##  5 Male   O       home    Female  O       No          620   55.5    49.1    24.5
##  6 Male   O       home    Male    O       No          185   73.7    29.9    27.1
##  7 Female B       inpt    Male    B       No          254   44.6    50.2    32.9
##  8 Male   A       home    Female  A       No           36   63.1    22.9    28.2
##  9 Male   A       ventil~ Female  O       No            1   41.0    19.8    43.9
## 10 Male   A       inpt    Male    A       No           12   49.0    28.7    26.5
## # ... with 3,990 more rows, 5 more variables: Donor_BMI <dbl>, MELD <dbl>,
## #   MELDNA <dbl>, Time <dbl>, Cens <dbl>, and abbreviated variable names
## #   1: Bloodtype, 2: Inpt_attx, 3: Donor_Sex, 4: Donor_Bloodtype, 5: Donor_DCD,
## #   6: Waittime, 7: Donor_Age, 8: TransplantBMI
```

data preprocessing

First, to data preprocessing

```
encode_post_trans <- post_trans
encode_post_trans$Gender <- as.numeric(factor(post_trans$Gender))
encode_post_trans$Bloodtype <- as.numeric(factor(post_trans$Bloodtype))
encode_post_trans$Inpt_attx <- as.numeric(factor(post_trans$Inpt_attx))
encode_post_trans$Donor_Sex <- as.numeric(factor(post_trans$Donor_Sex))
encode_post_trans$Donor_Bloodtype <- as.numeric(factor(post_trans$Donor_Bloodtype))
encode_post_trans$Donor_DCD <- as.numeric(factor(post_trans$Donor_DCD))
encode_post_trans
```

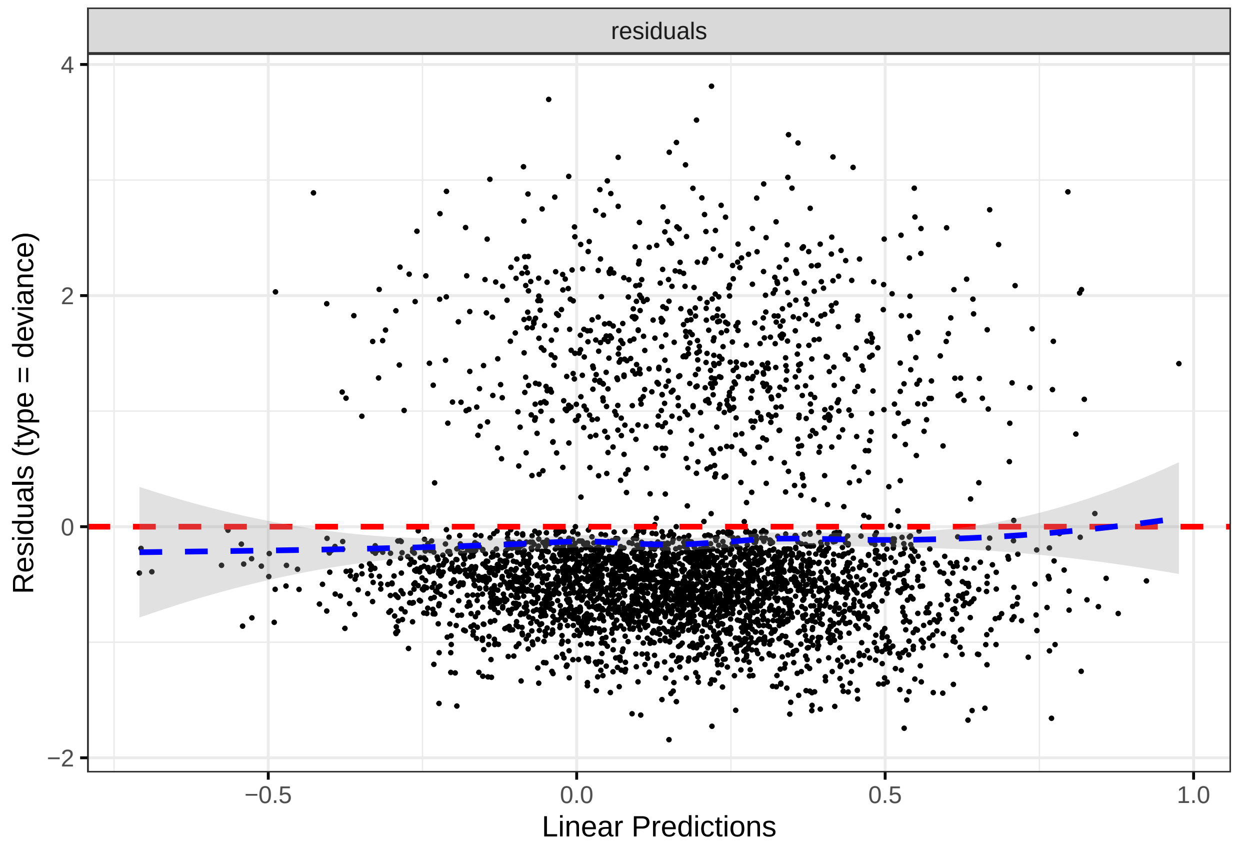
```
## # A tibble: 4,000 x 15
##    Gender Blood~1 Inpt_~2 Donor~3 Donor~4 Donor~5 Waitt~6 Age_Tx Donor~7 Trans~8
##     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>  <dbl>   <dbl>   <dbl>
##  1      2       4       3       2       3       1       1   54.3    45.4    24.7
##  2      2       1       1       2       4       1     157   59.8    60.5    28.9
##  3      1       1       1       2       1       1      49   63.2    31.0    26.9
##  4      2       4       3       1       3       1      81   28.8    41.1    27.4
##  5      2       4       1       1       4       1     620   55.5    49.1    24.5
##  6      2       4       1       2       4       1     185   73.7    29.9    27.1
##  7      1       3       3       2       3       1     254   44.6    50.2    32.9
##  8      2       1       1       1       1       1      36   63.1    22.9    28.2
##  9      2       1       4       1       4       1       1   41.0    19.8    43.9
## 10      2       1       3       2       1       1      12   49.0    28.7    26.5
## # ... with 3,990 more rows, 5 more variables: Donor_BMI <dbl>, MELD <dbl>,
## #   MELDNA <dbl>, Time <dbl>, Cens <dbl>, and abbreviated variable names
## #   1: Bloodtype, 2: Inpt_attx, 3: Donor_Sex, 4: Donor_Bloodtype, 5: Donor_DCD,
## #   6: Waittime, 7: Donor_Age, 8: TransplantBMI
```

Proportional Hazards

```
testFit1 <- coxph(Surv(Time, Cens) ~ Gender + Bloodtype + Inpt_attx + Donor_Sex + Donor_Bloodtype + Dono
my_res <- residuals(testFit1, "martingale")
```

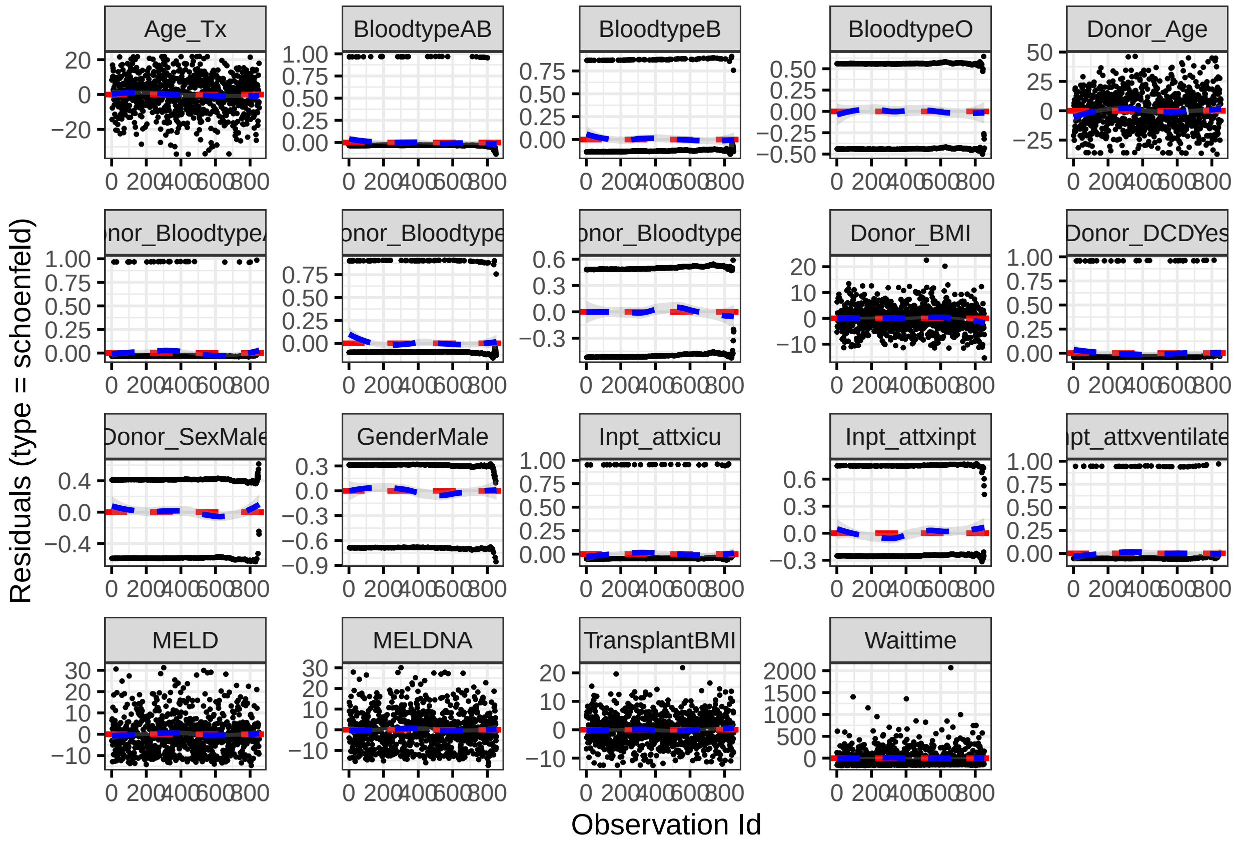
```
par(mfrow=c(2,2))
ggcoxdiagnostics(testFit1, type = "deviance", point.size = 0.3)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



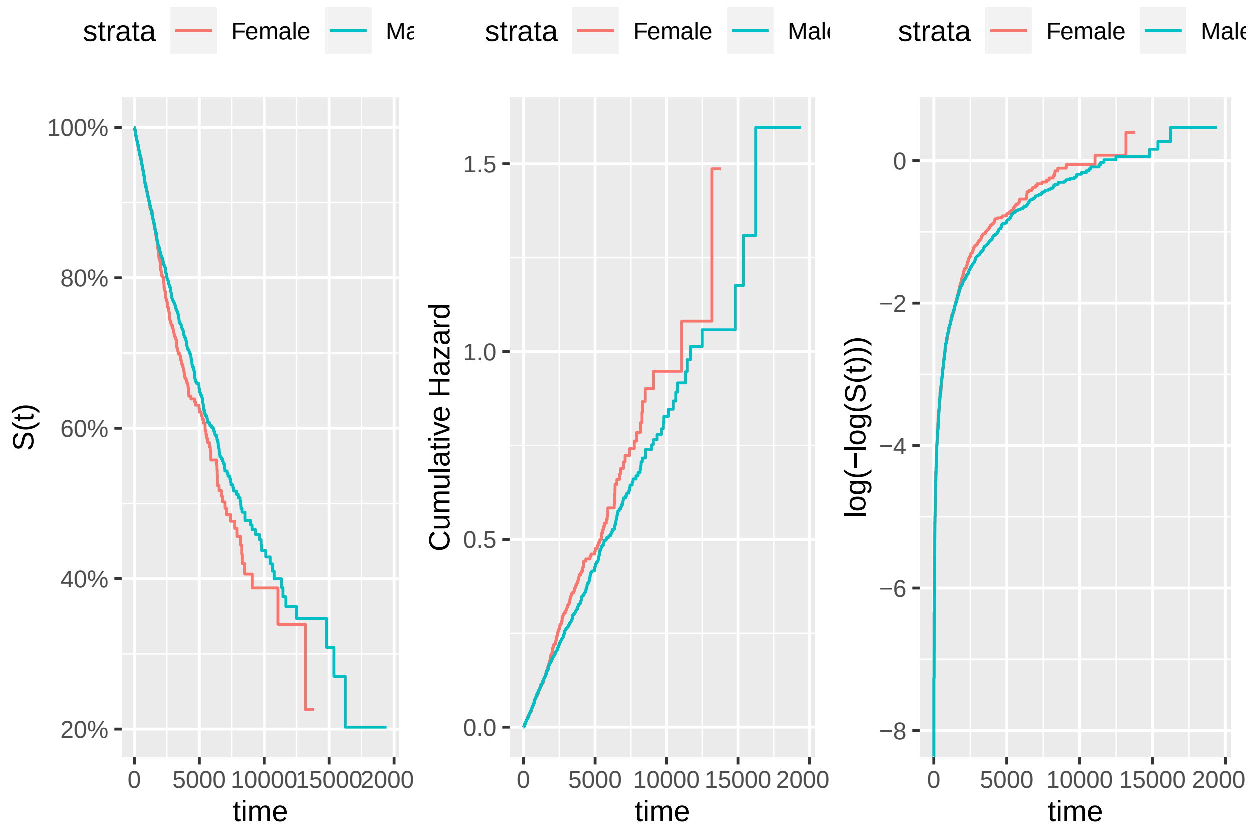
```
ggcoxdiagnostics(testFit1, type = "schoenfeld", point.size = 0.3)
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

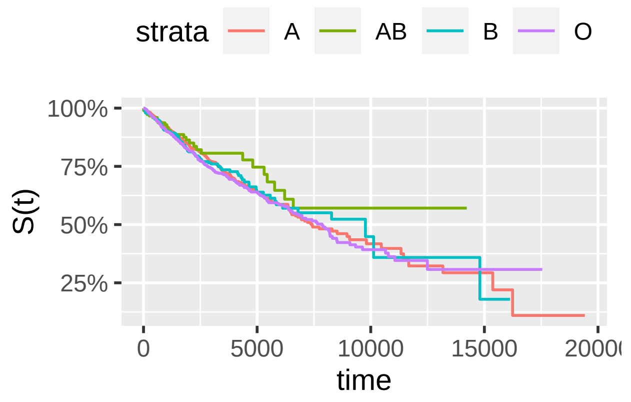


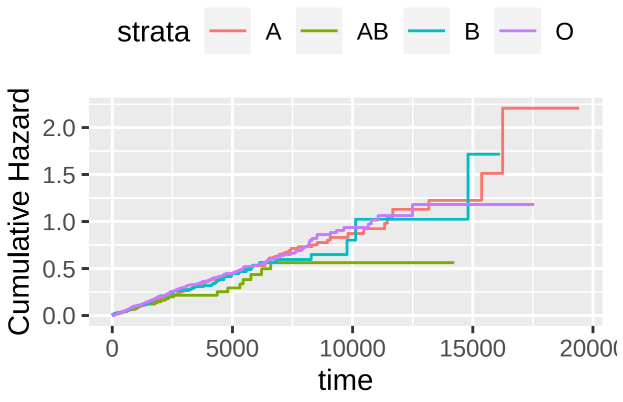
The curves for the two levels of Gender have little discrepancy on all three plots, indicating that the proportional effect of Gender is insignificant. The curves for the two levels of Occupation, however, display clear deviations from each other, and meet the standards in Hess (1995). This also echos the fact that Occupation is very significant in the fitted Cox model.

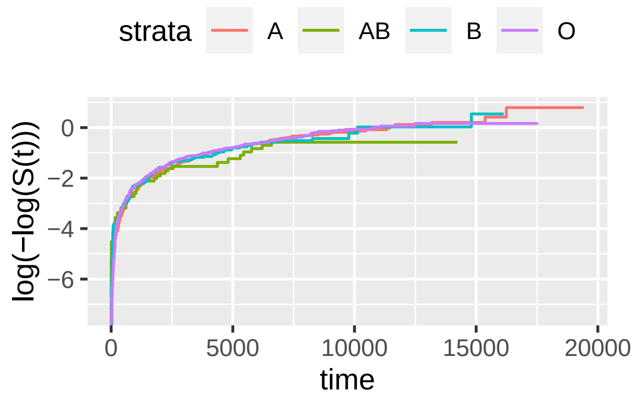
```r
## for Gender
sFit1 <- survfit(Surv(Time, Cens) ~ Gender, data = post_trans)
sF1Plot1 <- ggplot2::autoplot(sFit1, censor = FALSE, conf.int = FALSE) +
theme(legend.position = "top") + ylab("S(t)")
sF1Plot2 <- ggplot2::autoplot(sFit1, fun = "cumhaz", censor = FALSE, conf.int = FALSE) +
theme(legend.position = "top") + ylab("Cumulative Hazard")
sF1Plot3 <- ggplot2::autoplot(sFit1, fun = "cloglog", censor = FALSE, conf.int = FALSE) +
theme(legend.position = "top") + ylab("log(-log(S(t)))")
grid.arrange(sF1Plot1, sF1Plot2, sF1Plot3, ncol = 3)
```



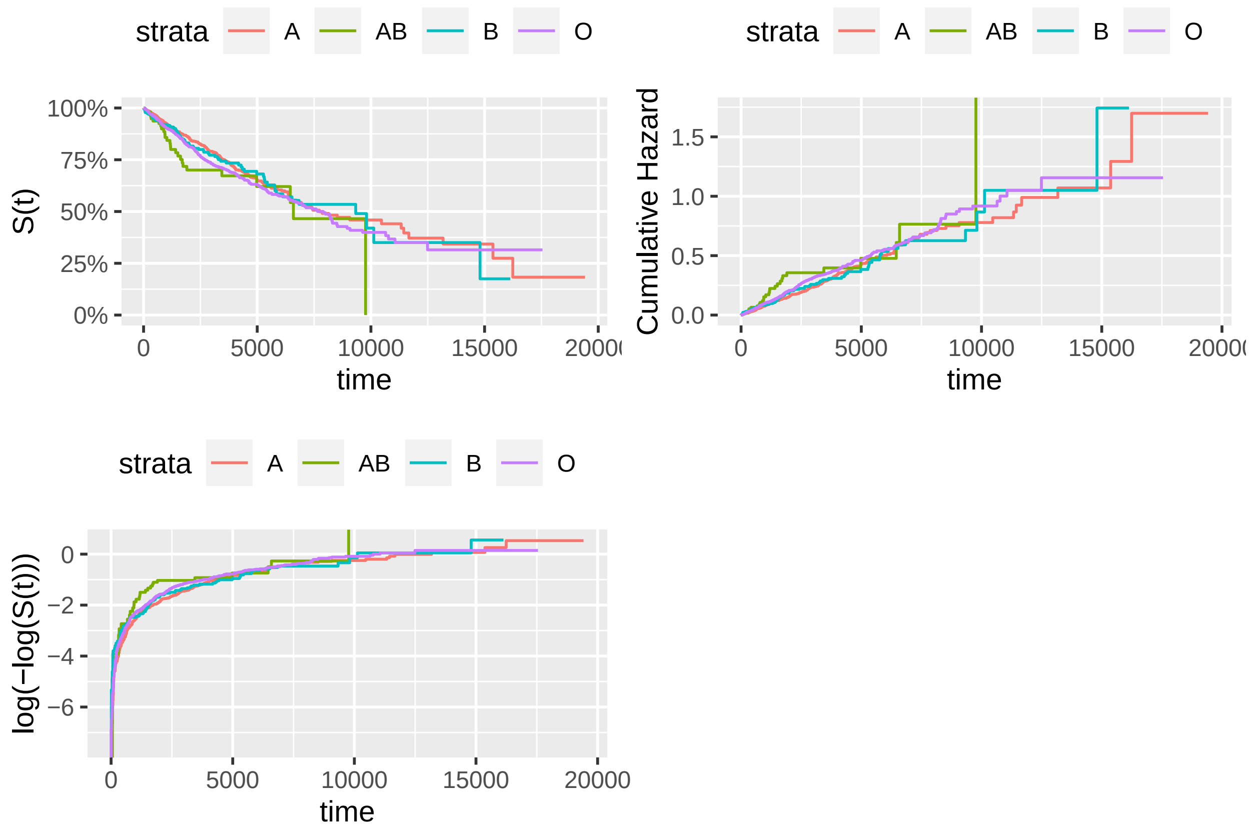
```
## for BloodType
sFit2 <- survfit(Surv(Time, Cens) ~ Bloodtype, data = post_trans)
sF2Plot1 <- ggplot2::autoplot(sFit2, censor = FALSE, conf.int = FALSE) +
theme(legend.position = "top") + ylab("S(t)")
sF2Plot2 <- ggplot2::autoplot(sFit2, fun = "cumhaz", censor = FALSE, conf.int = FALSE) +
theme(legend.position = "top") + ylab("Cumulative Hazard")
sF2Plot3 <- ggplot2::autoplot(sFit2, fun = "cloglog", censor = FALSE, conf.int = FALSE) +
theme(legend.position = "top") + ylab("log(-log(S(t)))")
grid.arrange(sF2Plot1, sF2Plot2, sF2Plot3, ncol = 2)
```



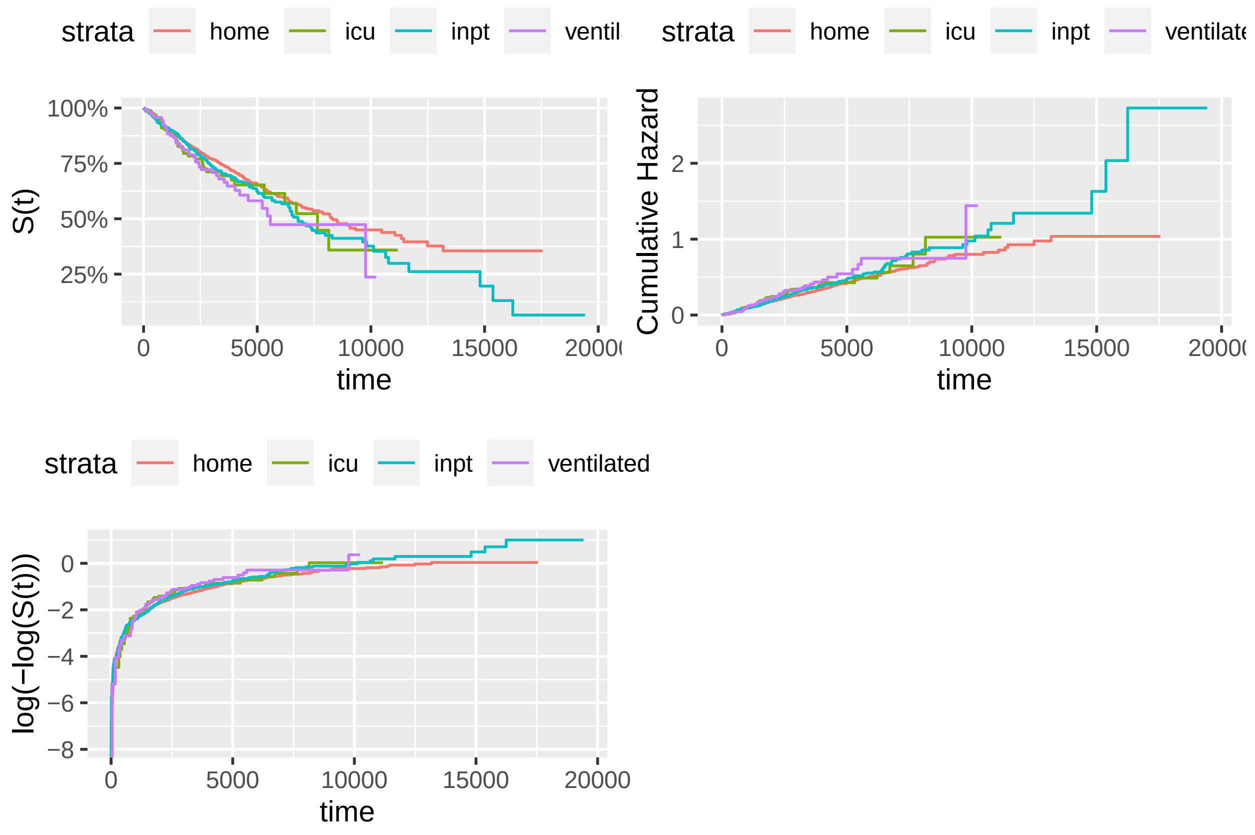




```r
# for Donor_Bloodtype
sFit3 <- survfit(Surv(Time, Cens) ~ Donor_Bloodtype, data = post_trans)
sF3Plot1 <- ggplot2::autoplot(sFit3, censor = FALSE, conf.int = FALSE) +
theme(legend.position = "top") + ylab("S(t)")
sF3Plot2 <- ggplot2::autoplot(sFit3, fun = "cumhaz", censor = FALSE, conf.int = FALSE) +
theme(legend.position = "top") + ylab("Cumulative Hazard")
sF3Plot3 <- ggplot2::autoplot(sFit3, fun = "cloglog", censor = FALSE, conf.int = FALSE) +
theme(legend.position = "top") + ylab("log(-log(S(t)))")
grid.arrange(sF3Plot1, sF3Plot2, sF3Plot3, ncol = 2)
```



```r
# for Donor_Bloodtype
sFit4 <- survfit(Surv(Time, Cens) ~ Inpt_attx, data = post_trans)
sF4Plot1 <- ggplot2::autoplot(sFit4, censor = FALSE, conf.int = FALSE) +
theme(legend.position = "top") + ylab("S(t)")
sF4Plot2 <- ggplot2::autoplot(sFit4, fun = "cumhaz", censor = FALSE, conf.int = FALSE) +
theme(legend.position = "top") + ylab("Cumulative Hazard")
sF4Plot3 <- ggplot2::autoplot(sFit4, fun = "cloglog", censor = FALSE, conf.int = FALSE) +
theme(legend.position = "top") + ylab("log(-log(S(t)))")
grid.arrange(sF4Plot1, sF4Plot2, sF4Plot3, ncol = 2)
```

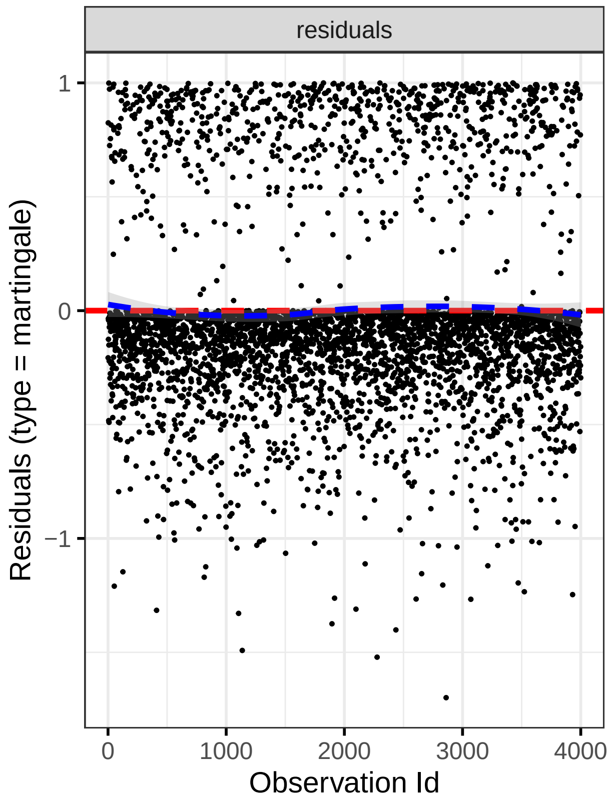


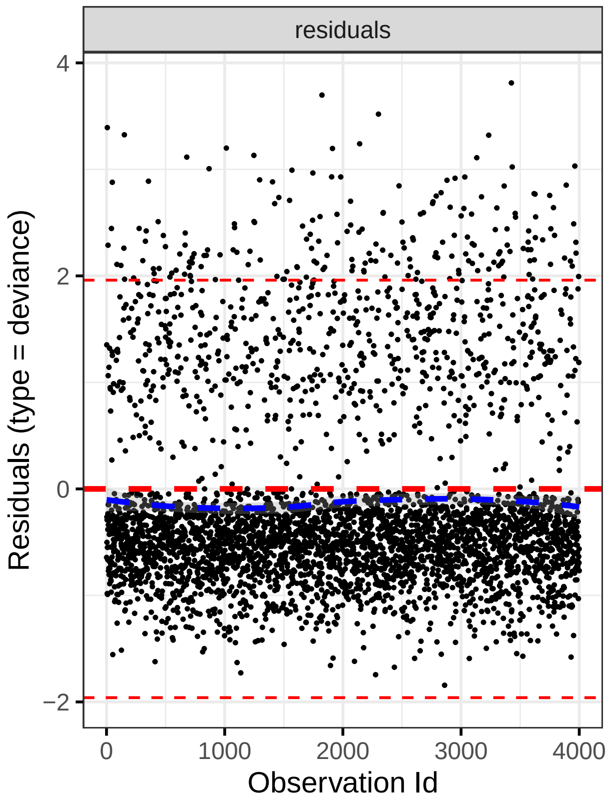
Outliers

We plot the martingale and deviance residuals to check for possible outliers as discussed in Terry M. Therneau,Grambsch, and Fleming (1990). The scale for x-axis can be linear predictions, time, or observation ID. We choose the observation ID here. The deviance residuals identified a group of potential outliers.

```
devPlot1 <- ggcoxdiagnostics(testFit1, type = "martingale", ox.scale = "observation.id",
point.size = 0.3) + scale_color_grey()
devPlot2 <- ggcoxdiagnostics(testFit1, type = "deviance", ox.scale = "observation.id",
point.size = 0.3) + scale_color_grey() + geom_hline(lty = 2, col = "red",
yintercept = c(1.96, -1.96))
marrangeGrob(list(devPlot1, devPlot2), ncol = 2, nrow = 1, top = "")
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

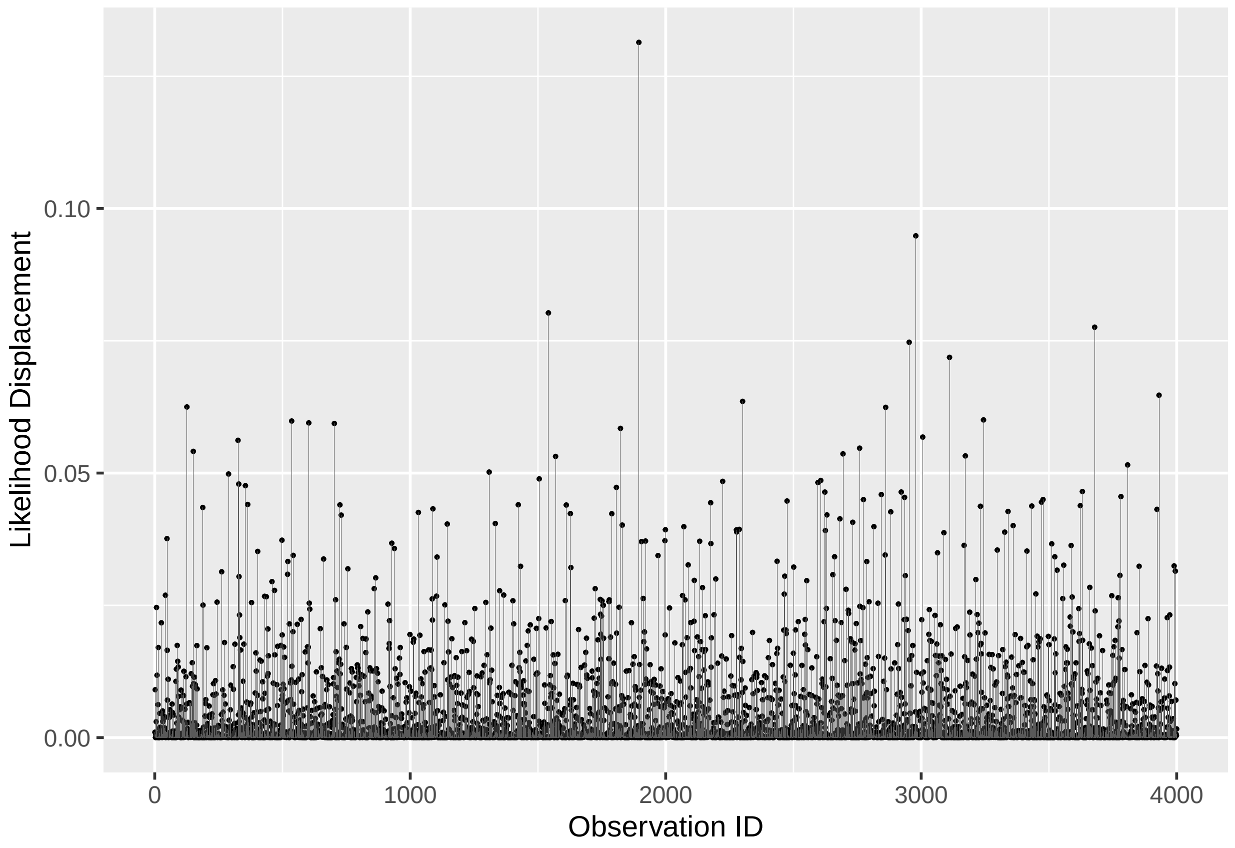




Influential Observations

We also plot the likelihood displacement (Pettitt and Daud 1989) of each observation. The five observations identified by dfbetas residual also have the largest likelihood displacement.

```
s.res <- residuals(testFit1, type = "score")
ld <- diag(s.res %*% testFit1$var %*% t(s.res))
lhd <- data.frame(ObservationID = 1:nrow(post_trans), LD = ld)
qplot(x = ObservationID, y = LD, data = lhd, geom = "point", ylab = "Likelihood Displacement",
xlab = "Observation ID", size = I(0.3)) + geom_bar(stat = "identity")
```



Likelihood Displacement
0.10
0.05
0.00
0
1000
2000
3000
4000
Observation ID