Survival Analysis Notes

Cox Proportional Hazards Model

We observe event times along with covariates which may help to predict the event times. We define a *hazard function*, $h(t)$, which gives the probability per unit time of the event occuring at t, given that it has not already occured. The Cox model attempts to measure the effects of the covariates on the hazard as opposed to measuring $h(t)$ directly.

The model for the i^{th} subject is

$$h_i(t) = h_o(t)exp(\mathbf{X}_i(t)\beta)$$

$h_o(t)$ is the baseline hazard so each subject's baseline hazard gets modified by their coefficient values.

One thing to note is that the coefficient values are only estimated when an event occurs. Not 100% on why this is.

The Probability that the i^{th} subject expired at t_i is (if $R(i)$ is the set of all subjects at risk at time t)

$$exp(\mathbf{X}_i\beta)/\sum_{j\in R(i)} exp(\mathbf{X}_j\beta)$$

So the liklihood becomes

$$\prod_i exp(\mathbf{X}_i\beta)/\sum_{j\in R(i)} exp(\mathbf{X}_j\beta)$$

If subjectes leave the study before their event occurs (or if the study ends before they do) then they still count in $R(i)$ (The denominator) but don't exist in the numerator. This is called Right Censoring.

Because we are not estimating h_o (it cancels out of the above equations) this is called the *partial liklihood*.

Cumulative Hazard and Survival Functions

The probability of surviving until time t is called the *survival function*, $S(t)$. If $H(t) = \int_0^t h(x)dx$ then $S(t) = exp(-H(t))$. The baseline hazard can then be given as

$$\hat{H}_o(t) = \sum_{t_j \le t} exp(\hat{\alpha}_j)$$

where α_i is the intercept term for the i^{th} time.

Another way to think about this is the following equation:

$$exp(\alpha_j) = \int_{t_{j-1}^+}^{t_j} h_o(x)dx$$

Which basically says that the intercept at time j is is the baseline hazard function between the previous event and the current one. For a subject specific hazard, $\hat{H}(t)$, you just add the subjects covariates to the intercept.

The useful residuals are just $d_i - \hat{H}_i$

Example

Fit a model using the `bone` data set; `t` is the time, `d` is a binary indicator where 1 is death and 0 is censored, and `trt` is a factor with levels `allo` and `auto`.

t	d	trt
28	1	allo
32	1	allo
49	1	allo
84	1	allo
357	1	allo
933	0	allo

We are going to use the approach in the book of creating a record for every event time a subject is alive for.

```
## order by t
bone$id <- 1:nrow(bone)
bone_ordered <- bone[order(bone$t), ]
## event times
et <- unique(bone_ordered$t[bone_ordered$d == 1])
## Starts of risk sets
es <- match(et, bone_ordered$t)
n <- nrow(bone_ordered)
# Times for risk sets
t <- rep(et, 1 + n - es)
print(et)
```

```
##  [1]  28  32  42  49  53  57  63  81  84 140 176 252 357 524
```

```
print(1 + n - es)
```

```
##  [1] 23 22 21 20 19 18 17 16 15 14 13 11 10  8
```

```
str(t)
```

```
##  num [1:227] 28 28 28 28 28 28 28 28 28 28 ...
```

```
st <- cbind(0,
            bone_ordered[unlist(apply(matrix(es), 1, function(x, n) x:n, n = n)),])
## Signal Events
st[(st$t == t) & (st$d != 0), 1] <- 1
## Reset Event Time to risk set time
st$t <- t
names(st)[1] <- "z"

knitr::kable(head(st))
```

	z	t	d	trt	id
1	1	28	1	allo	1
2	0	28	1	allo	2
12	0	28	1	auto	12
3	0	28	1	allo	3
13	0	28	1	auto	13
14	0	28	1	auto	14

```
pb <- st
pb$tf <- factor(pb$t)
bone_gam <- glm(z ~ tf + trt - 1, poisson, pb)
```

`tf` is first so it is a contrast free term, so we estimate a seperate a_j for each event.

```
cum_hazard <- tapply(fitted(bone_gam), pb$id, sum)
## Martingale Residuals
m_resid <- bone$d - cum_hazard
summary(bone_gam)
```

```
##
## Call:
## glm(formula = z ~ tf + trt - 1, family = poisson, data = pb)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6346  -0.3786  -0.3387  -0.2571   2.2767
##
## Coefficients:
##         Estimate Std. Error z value Pr(>|z|)
## tf28     -3.5632     1.0742  -3.317  0.00091 ***
## tf32     -3.5345     1.0784  -3.277  0.00105 **
## tf42     -3.5049     1.0830  -3.236  0.00121 **
## tf49     -3.4421     1.0793  -3.189  0.00143 **
## tf53     -3.4096     1.0844  -3.144  0.00166 **
## tf57     -3.3404     1.0802  -3.092  0.00199 **
## tf63     -3.2661     1.0756  -3.037  0.00239 **
## tf81     -3.1857     1.0703  -2.976  0.00292 **
## tf84     -3.0984     1.0643  -2.911  0.00360 **
## tf140    -3.0522     1.0703  -2.852  0.00435 **
## tf176    -2.9517     1.0634  -2.776  0.00551 **
## tf252    -2.7142     1.0457  -2.596  0.00944 **
## tf357    -2.5703     1.0345  -2.485  0.01297 *
## tf524    -2.3072     1.0260  -2.249  0.02453 *
## trtauto   0.7046     0.5701   1.236  0.21648
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 426.000  on 227  degrees of freedom
## Residual deviance:  75.198  on 212  degrees of freedom
## AIC: 133.2
##
## Number of Fisher Scoring iterations: 6
```

```
drop1(bone_gam, test = "Chisq")
```

```
## Single term deletions
##
## Model:
## z ~ tf + trt - 1
##        Df Deviance   AIC     LRT Pr(>Chi)
## <none>      75.198 133.2
```

```
## tf     14  261.702 291.7 186.504   <2e-16 ***
## trt     1   76.799 132.8   1.601   0.2057
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So the auto treatment is not significant in the model. A Positive value for the trtauto variable means that the hazard is estimated to be higher for the auto group.

```
library(dplyr)
```

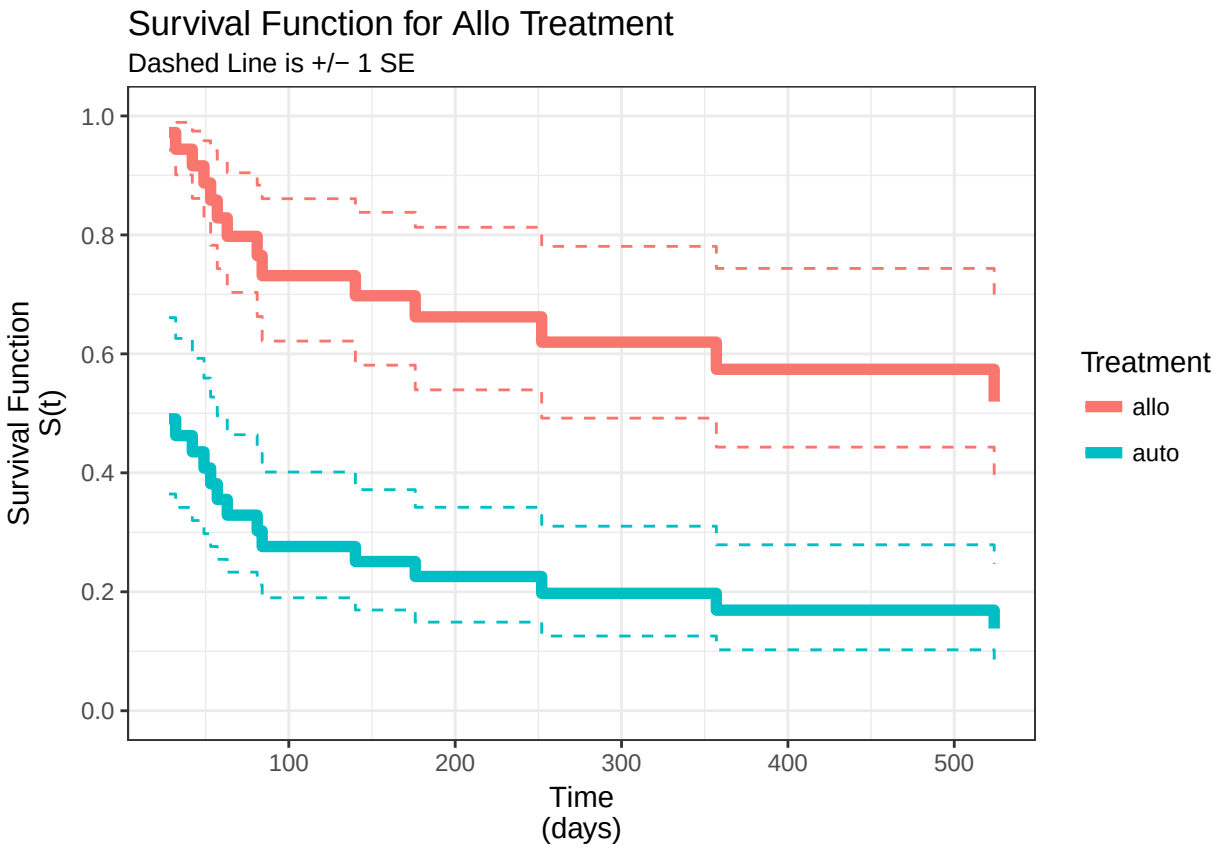
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
base_plot_df <- data_frame(te = sort(unique(bone$t[bone$d == 1])),
                           tf = factor(te),
                           trt = bone$trt[1]) %>%
  bind_rows(data_frame(te = sort(unique(bone$t[bone$d == 1])),
                           tf = factor(te),
                           trt = bone$trt[20]))

base_plot_df <- base_plot_df %>%
  mutate(raw_preds = as.numeric(predict(bone_gam, base_plot_df)),
         cum_hazard = cumsum(exp(raw_preds)),
         survive = exp(-cum_hazard))

X <- model.matrix(~tf + trt - 1, base_plot_df)
J <- apply(exp(base_plot_df$raw_preds) * X, 2, cumsum)
se_raw <- diag(J %*% vcov(bone_gam) %*% t(J))^.5

base_plot_df <- base_plot_df %>%
  mutate(se = exp(-cum_hazard + se_raw),
         se_minus = exp(-cum_hazard - se_raw))

ggplot(aes(x = te, y = survive, color = trt), data = base_plot_df) +
  geom_step(size = 2) +
  geom_step(aes(y = se), linetype = "dashed") +
  geom_step(aes(y = se_minus), linetype = "dashed") +
  xlab("Time\n(days)") +
  ylab("Survival Function\nS(t)") +
  ggtitle("Survival Function for Allo Treatment", "Dashed Line is +/- 1 SE") +
  scale_y_continuous(breaks = seq(0, 1, by = .2), limits = c(0, 1)) +
  scale_color_discrete(name = "Treatment") +
  theme(legend.position = "top") +
  theme_bw()
```

```
## Warning: Removed 1 rows containing missing values (geom_path).
```



Survival Analysis with MGCV

We have data for drug trials. `pbc` contains baseline measures for each patient and `pbcseq` contains some time-varying values.

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
## This is mgcv 1.8-17. For overview type 'help("mgcv-package")'.
```

id	time	status	trt	age	sex	ascites	hepato	spiders	edema	bili	chol	albumin	copper	alk.phos
1	400	2	1	58.76523	f	1	1	1	1.0	14.5	261	2.60	156	1718.0
2	4500	0	1	56.44627	f	0	1	1	0.0	1.1	302	4.14	54	7394.8
3	1012	2	1	70.07255	m	0	0	0	0.5	1.4	176	3.48	210	516.0
4	1925	2	1	54.74059	f	0	1	1	0.5	1.8	244	2.54	64	6121.8
5	1504	1	2	38.10541	f	0	1	1	0.0	3.4	279	3.53	143	671.0
6	2503	2	2	66.25873	f	0	1	0	0.0	0.8	248	3.98	50	944.0

id	futime	status	trt	age	sex	day	ascites	hepato	spiders	edema	bili	chol	albumin	alk.phos
1	400	2	1	58.76523	f	0	1	1	1	1	14.5	261	2.60	1718

id	futime	status	trt	age	sex	day	ascites	hepato	spiders	edema	bili	chol	albumin	alk.phos
1	400	2	1	58.76523	f	192	1	1	1	1	21.3	NA	2.94	1612
2	5169	0	1	56.44627	f	0	0	1	1	0	1.1	302	4.14	7395
2	5169	0	1	56.44627	f	182	0	1	1	0	0.8	NA	3.60	2107
2	5169	0	1	56.44627	f	365	0	1	1	0	1.0	NA	3.55	1711
2	5169	0	1	56.44627	f	768	0	1	1	0	1.9	NA	3.92	1365

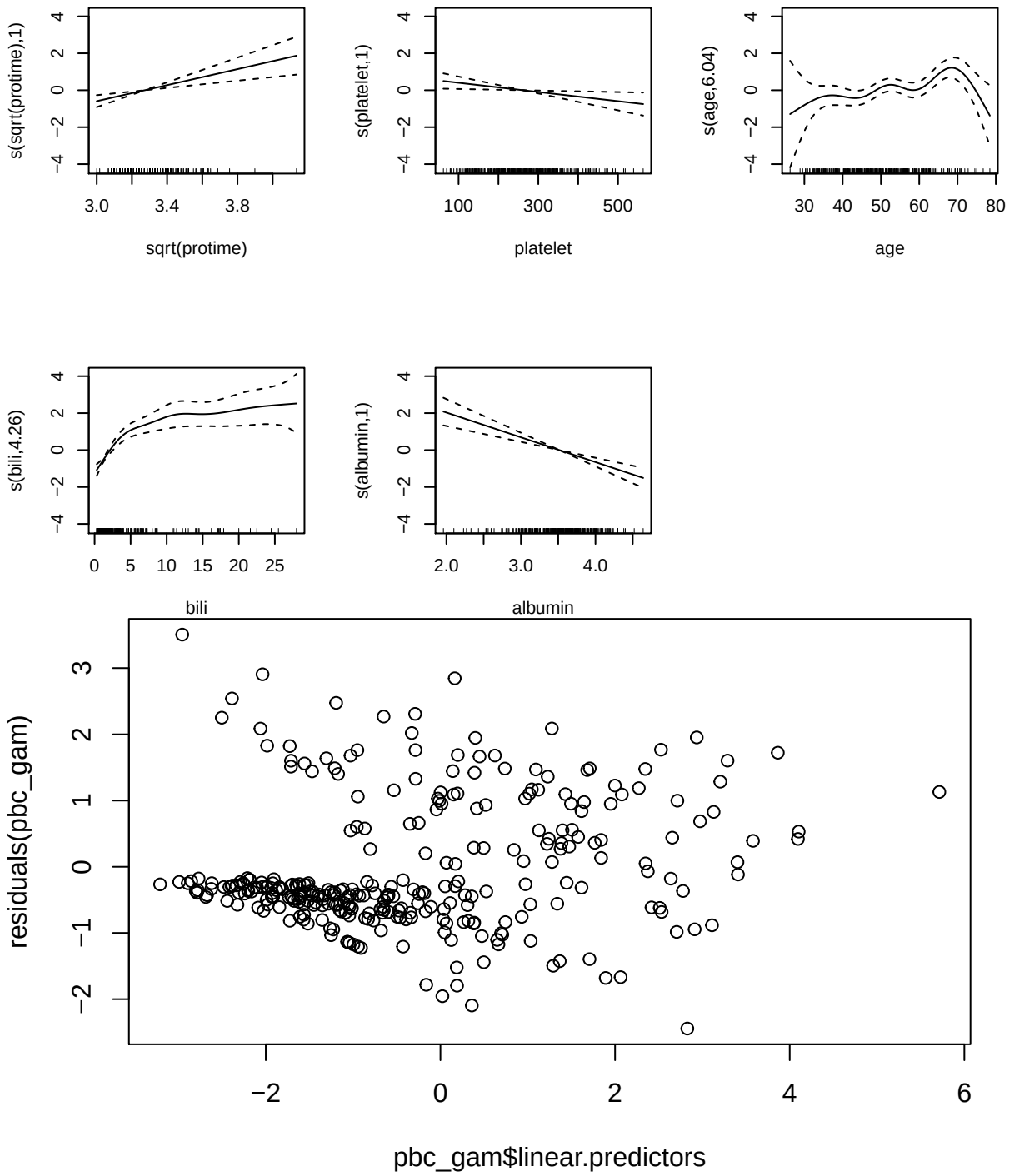
The disease takes 4 `stage` values, with 1 being subtle damage and 4 is cirrhosis. The weights vector provides the censoring information (0 for censoring, 1 for event)

```
pbc$status1 <- as.numeric(pbc$status == 2)
pbc$stage <- factor(pbc$stage)

pbc_gam <- gam(time ~ trt + sex + s(sqrt(protime)) + s(platelet) + s(age) + s(bili) + s(albumin),
          weights = status1,
          family = cox.ph,
          data = pbc)

anova(pbc_gam)
```

```
##
## Family: Cox PH
## Link function: identity
##
## Formula:
## time ~ trt + sex + s(sqrt(protime)) + s(platelet) + s(age) +
##     s(bili) + s(albumin)
##
## Parametric Terms:
##     df Chi.sq p-value
## trt  1  0.120  0.7294
## sex  1  3.439  0.0637
##
## Approximate significance of smooth terms:
##                     edf Ref.df Chi.sq  p-value
## s(sqrt(protime)) 1.000  1.001 13.337 0.000261
## s(platelet)      1.001  1.002  5.789 0.016141
## s(age)           6.043  7.172 29.417 0.000145
## s(bili)          4.264  5.223 89.545  < 2e-16
## s(albumin)       1.000  1.000 31.086 2.47e-08
```

So the treatment is not significant

```
plot(pbc_gam, pages = 1); plot(pbc_gam$linear.predictors, residuals(pbc_gam))
```



The wedge of points on the residual plot is from the censored observations and is expected in survival analysis residual plots.

```
np <- 300
newd <- data.frame(matrix(0, np, 0))
for(n in names(pbc)) newd[[n]] <- rep(pbc[[n]][25], np)
newd$time <- seq(0, 4500, length = np)
fv <- predict(pbc_gam, newdata = newd, type = "response", se = T)
newd$fit <- fv$fit
newd$se_fit <- fv$se.fit
```

```
newd$se <- with(newd, se_fit / fit)
newd$se_pred <- with(newd, exp(log(fit) + se))
newd$se_pred_minus <- with(newd, exp(log(fit) - se))

ggplot(aes(x = time, y = fit), data = newd) +
  geom_step() +
  geom_step(aes(y = se_pred), linetype = "dashed") +
  geom_step(aes(y = se_pred_minus), linetype = "dashed") +
  xlab("Time") +
  ylab("Survival Function\nS(t)") +
  ggtitle("Predicted Survival Function for Patient 25") +
  theme_bw()
```