Survival Analysis with Tidy Models

R in Pharma Recap 2021

Jenny Leopoldina Smith

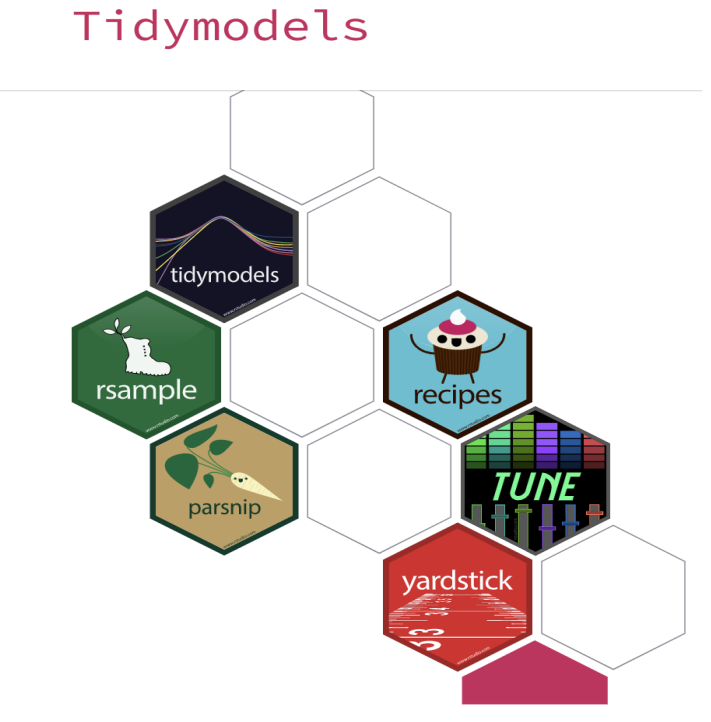
Fred Hutchinson Cancer Research Center

2021-12-08

Introduction

- Introduction of Time-to-event and Cox Proportional Hazards Regression Modeling with Tidy Models framework

- Description of what I found interesting during the talk from RStudio
    - Max Kuhn and Hannah Frick
- Beginning to attempt to incorporate Tidy Models in my own work
    - pediatric acute myeloid leukemia (AML)



What is the Tidyverse?

- From the documentation:
  - The tidyverse is an opinionated collection of R packages designed for data science.
- The tidyverse syntax relys on the use of a %>% pipe, which allows for:
  - modularity
  - readablility

```
```{r}
library(dplyr)
mean_mass_by_homeworld <- starwars %>%
  mutate(bmi = mass / ((height / 100)  ^ 2)) %>%
  select(name:mass, bmi, homeworld) %>%
  group_by(homeworld) %>%
  summarize(mean_mass=mean(mass)) %>%
  ungroup() %>%
  arrange(desc(mass))
```
```

In comparison

```r
#base R
starwars$bmi <- bmi = mass / ((height / 100)  ^ 2)
columns_sel <- c(1:3, grep("bmi|homeworld", colnames(starwars)))
starwars <- starwars[,columns_sel]
mean_mass_by_homeworld <- tapply(starwars$mass,
                                 starwars$homeworld,
                                 FUN=mean)
reordered <- order(mean_mass_by_homeworld, decreasing=TRUE)
mean_mass_by_homeworld <- mean_mass_by_homeworld[reordered]
```

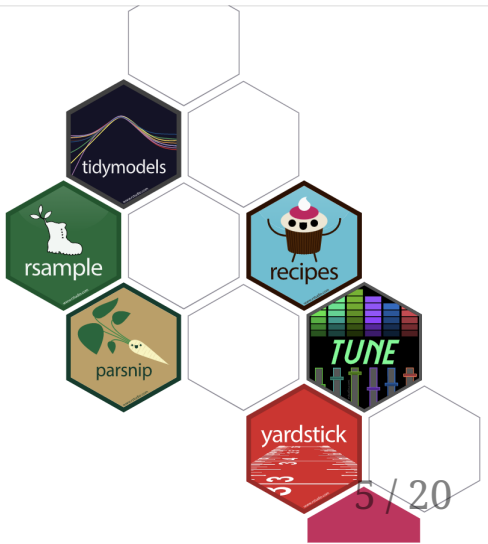
[1] Tidyverse Landing Page

Tidy Models

- What are tidy models?
  - The `tidymodels` framework is a collection of packages for modeling and machine learning using tidyverse principles.
- As with tidyverse
  - it includes numerous packages
  - you can pick and choose to install to complete your specific analysis
  - except of course for core packages

- Why Tidy models?
  - Consistent interface for modeling functions
  - resampling, assessing performance, and hyperparameter tuning.
  - Performance metrics (in the yardstick package)
  - Model tuning (with the tune package)

`Tidymodels`



Survival Analysis

- `censored` package along with `parsnip` package
  - `parsnip` is one of the core R packages in the Tidy Models framework
- First, survival and time-to-event analysis uses censored data.
  - **right censored**: Patient enrolled on study and diagnosed on 12/20/2020 and is still alive at 12/06/2021
  - **left censored**: Patient was on trial on 12/20/2020, but don't know when they were diagnosed, and was relapsed (event) by 12/06/2021
  - **interval censoring**: Patient was on trial on 12/20/2020, but don't know when they were diagnosed,and was not relapsed by 12/06/2021

[1] Parsnip Landing Page

Cox Proportional Hazards Regression

- A regression model commonly used in biomedical research
- The association between the survival time of patients and one or more predictor variables.
- It provides a means to estimate the risk (hazard) of the predictor variables between 2 or more groups.
    - For example, Patient Group A has **mutation A** and Patient group B has **no mutation**
    - Cox PH model can help associate if **mutation A** has higher incidence of events (relapse, death, etc) compared to **no mutation**
    - Results: Presence of **mutation A** is associated with a 2x increased risk of death from the time of diagnosis (hazard ratio = 2.0, 95% CI = 1.5-2.2, p-value=0.05)

Cox Proportional Hazards Regression: libraries

- CPH can be used to relate many risk factors and variables simultaneously to survival time.
- Here, I will present a univariate CPH model for simplicity.

```
library(tidymodels)
library(censored)
library(survival)
str(aml)  #dataframe provided withthe survival package
```

```
'data.frame':    23 obs. of  3 variables:
 $ time  : num  9 13 13 18 23 28 31 34 45 48 ...
 $ status: num  1 1 0 1 1 0 1 1 0 1 ...
 $ x     : Factor w/ 2 levels "Maintained","Nonmaintained": 1 1 1 1 1 1 1 1 1
```

Cox Proportional Hazards Regression

- Note `Surv(time, status)` is from the survival package and calculate Kaplan-Meier estimates which take into account censoring (right, left, interval)

```
tidymodels_prefer()  # to prevent common clashes

cph_fit <- proportional_hazards(engine = "survival") %>%
    fit(Surv(time, status) ~ x, data = aml)

cph_fit
```

```
parsnip model object

Call:
survival::coxph(formula = Surv(time, status) ~ x, data = data,
    model = TRUE, x = TRUE)

                coef exp(coef) se(coef)     z      p
xNonmaintained 0.9155    2.4981   0.5119 1.788 0.0737

Likelihood ratio test=3.38  on 1 df, p=0.06581
n= 23, number of events= 18
```

Cox Proportional Hazards Regression

```
# cox proportional Hazards
cph_fit2 <- coxph(Surv(time, status) ~ x, data = aml)

cph_fit2
```

```
Call:
coxph(formula = Surv(time, status) ~ x, data = aml)

                coef exp(coef) se(coef)     z      p
xNonmaintained 0.9155    2.4981   0.5119 1.788 0.0737

Likelihood ratio test=3.38  on 1 df, p=0.06581
n= 23, number of events= 18
```

Cox Proportional Hazards Regression

```
summary(cph_fit2)
```

```
Call:
coxph(formula = Surv(time, status) ~ x, data = aml)

  n= 23, number of events= 18

                 coef exp(coef) se(coef)     z Pr(>|z|)
xNonmaintained 0.9155    2.4981   0.5119 1.788   0.0737 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

               exp(coef) exp(-coef) lower .95 upper .95
xNonmaintained     2.498     0.4003    0.9159     6.813

Concordance= 0.619  (se = 0.063 )
Likelihood ratio test= 3.38  on 1 df,   p=0.07
Wald test            = 3.2  on 1 df,   p=0.07
Score (logrank) test = 3.42  on 1 df,   p=0.06
```

Cox Proportional Hazards Regression

Versus

```
tidy(cph_fit)
```

```
# A tibble: 1 × 5
  term            estimate std.error statistic p.value
  <chr>              <dbl>     <dbl>     <dbl>   <dbl>
1 xNonmaintained    0.916     0.512      1.79  0.0737
```

Predict with New Data

- First, simulate some patient outcome data

```
set.seed(1)
new_data <- tibble(Patient = paste0("p", 1:10), time = sample(3:200,
    status = sample(c(1, 0), size = 10, replace = T, prob = c(0.3, 0.
        "Nonmaintained"), size = 10, replace = T) %>%
        factor(., levels = levels(aml$x)))

head(new_data)
```

```
# A tibble: 6 × 4
  Patient  time status x
  <chr>   <int>  <dbl> <fct>
1 p1         70      0 Maintained
2 p2        169      1 Nonmaintained
3 p3        131      1 Maintained
4 p4        164      0 Maintained
5 p5         45      1 Nonmaintained
6 p6         16      1 Nonmaintained
```

Predict with New Data

- Predict the survival probabilities (type = "survival") at given times (times argument)

```
times <- seq(0, 72, by = 6)

prediction_df <- predict(cph_fit, new_data, type = "survival", time =
    bind_cols(new_data) %>%
    unnest(cols = c(.pred)) %>%
    group_by(Patient) %>%
    mutate(n_pred = 1:n()) %>%
    ungroup()

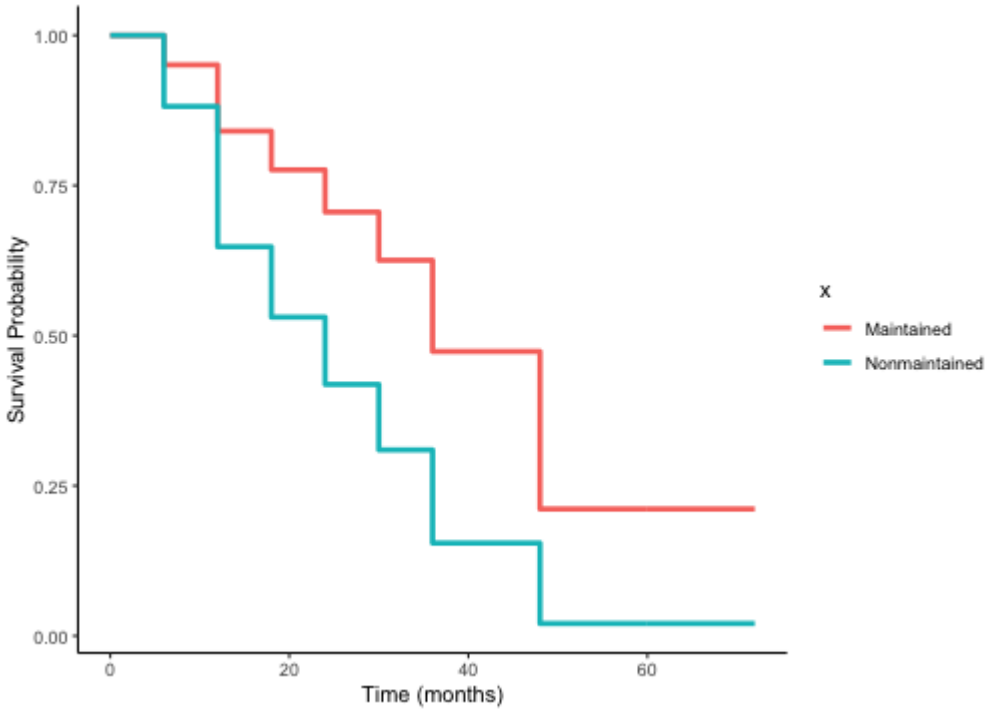
head(prediction_df)
```

```
# A tibble: 6 × 7
  .time .pred_survival Patient  time status x           n_pred
  <dbl>          <dbl> <chr>   <int>  <dbl> <fct>       <int>
1     0          1     p1         70      0 Maintained      1
2     6          0.951 p1         70      0 Maintained      2
3    12          0.841 p1         70      0 Maintained      3
4    18          0.776 p1         70      0 Maintained      4
5    24          0.706 p1         70      0 Maintained      5
6    30          0.625 p1         70      0 Maintained      6
```

Predict with New Data

- Visualize the data with `geom_step` to create a Kaplan-Meier curve.

```
prediction_df %>%
    ggplot(aes(x = .time, y = .pred_survival, group = x, col = x)) +
    labs(x = "Time (months)", y = "Survival Probability") + theme_cla
```



More Advanced Models

- regularized cox proportional hazards models
- compare to `glmnet` R package
  - much more difficult interface in `glmnet`

```
```{r}
cph_glmnet_strata_fit <-
  proportional_hazards(penalty = 0.1, mixture = 0.75) %>%
  set_engine("glmnet") %>%
  fit(Surv(age, adopted) ~ . + strata(sex), data = dogs)
```
```

More Advanced Models

- for example, using decisions trees
  - boost_tree()
  - decision_tree()
  - rand_forest()

```
```{r}
bag_fit <-
  bag_tree() %>% #defines an ensemble of decision trees.
  set_mode("censored regression") %>%
  set_engine("rpart", times = 50) %>%
  fit(Surv(age, adopted) ~ ., data = dogs)
```
```

Upcoming Support in Survival Analysis

- A role for censoring indicator columns and a step `step_surv()` in recipes.
- A new ROC metric for survival models in `yardstick`.
- An adaption of workflows and `tune` is to follow after that.

[1] Parsnip blog

References

- Tidyverse
- Tidy Models
- Tidy models bookdown
- Vignette with `censored` package
- R in pharma Survival Analysis presentation
- R in pharma youtube

Questions?

Please feel free to contact me: *JennyL.Smith12 [at] gmail.com*

[1] Slides created with the R package **xaringan**.