Meta-analysis of survival models in the DataSHIELD platform

Soumya Banerjee, Tom Bishop and DataSHIELD technical team

23 April 2021

Contents

1 Summary

This is a document that outlines technical notes and a development plan for implementing survival models and meta-analyzing hazard ratios in the DataSHIELD platform.

2 Survival analysis in DataSHIELD

We outline the development and code for implementing survival models and meta-analysis of hazard ratios in DataSHIELD.

All code is available here:

- https://github.com/neelsoumya/dsBaseClient/tree/absolute_newbie_client
- https://github.com/neelsoumya/dsBase/tree/absolute_newbie
- https://github.com/neelsoumya/datashield_testing_basic/blob/master/development_plan.rmd
- https://github.com/neelsoumya/datashield_testing_basic/blob/master/development_plan.pdf

The computational steps are outlined below. The first step is connecting to the server and loading the survival data. We assume that the reader is familiar with these details.

3 Creating server-side variables for survival analysis

We now outline some steps for analysing survival data.

- make sure that the outcome variable is numeric

```
ds.asNumeric(x.name = "D$cens",
             newobj = "EVENT",
             datasources = connections)

ds.asNumeric(x.name = "D$survtime",
             newobj = "SURVTIME",
             datasources = connections)
```

- convert time id variable to a factor

```
ds.asFactor(input.var.name = "D$time.id",
            newobj = "TID",
            datasources = connections)
```

- create in the server-side the log(survtime) variable

```
ds.log(x = "D$survtime",
       newobj = "log.surv",
       datasources = connections)
```

- create start time variable

```
ds.asNumeric(x.name = "D$starttime",
             newobj = "STARTTIME",
             datasources = connections)

ds.asNumeric(x.name = "D$endtime",
             newobj = "ENDTIME",
             datasources = connections)
```

4 Create survival object and call coxphSLMA()

- use constructed Surv object in *coxph.SLMA()*

```
dsBaseClient::ds.Surv(time='STARTTIME', time2='ENDTIME',
                      event = 'EVENT', objectname='surv_object',
                      type='counting')

coxph_model_full <- dsBaseClient::ds.coxph.SLMA(formula = 'surv_object~D$age+D$female')
```

- use direct inline call to *survival::Surv()*

```
ds.coxph.SLMA(formula = 'survival::Surv(time=SURVTIME,event=EVENT)~D$age+D$female',
                dataName = 'D',
                datasources = connections)
```

- call with *survival::strata()*

```
coxph_model_strata <- dsBaseClient::ds.coxph.SLMA(formula = 'surv_object~D$age +
                          survival::strata(D$female)')

summary(coxph_model_strata)
```

5 Summary of survival objects

We can also summarize a server-side object of type *survival::Surv()* using a call to *ds.summary()*. This will provide a non-disclosive summary of the server-side object. An example call is shown below:

```
dsBaseClient::ds.summary(x = 'surv_object')
```

6 Diagnostics for Cox proportional hazards models

We have also created functions to test for the assumptions of Cox proportional hazards models.

```
dsBaseClient::ds.coxphSLMAassign(formula = ’surv_object~D$age+D$female’,
                            objectname = ’coxph_serverside’)

dsBaseClient::ds.cox.zphSLMA(fit = ’coxph_serverside’)

dsBaseClient::ds.coxphSummary(x = ’coxph_serverside’)
```

A diagnostic summary is shown below.

```
## surv_object~D$age+D$female

## NULL

## $study1
##          chisq df    p
## D$age    1.022  1 0.31
## D$female 0.364  1 0.55
## GLOBAL   1.239  2 0.54
##
## $study2
##              chisq df    p
## D$age    -1389.472  1 1.00
## D$female     0.591  1 0.44
## GLOBAL    -857.492  2 1.00
##
## $study3
##          chisq df       p
## D$age    15.27  1 9.3e-05
## D$female  8.04  1  0.0046
## GLOBAL   23.31  2 8.7e-06

## $study1
## Call:
## survival::coxph(formula = formula, data = dataTable, weights = weights,
##     ties = ties, singular.ok = singular.ok, model = model, x = x,
##     y = y)
##
##   n= 2060, number of events= 426
##
##                 coef exp(coef)  se(coef)      z Pr(>|z|)
## D$age      0.041609  1.042487  0.003498 11.894  < 2e-16 ***
## D$female1 -0.660002  0.516850  0.099481 -6.634 3.26e-11 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## D$age        1.0425     0.9592    1.0354    1.0497
## D$female1    0.5169     1.9348    0.4253    0.6281
```

```
##
## Concordance= 0.676  (se = 0.014 )
## Likelihood ratio test= 170.7  on 2 df,   p=<2e-16
## Wald test            = 168.2  on 2 df,   p=<2e-16
## Score (logrank) test = 166.3  on 2 df,   p=<2e-16
##
##
## $study2
## Call:
## survival::coxph(formula = formula, data = dataTable, weights = weights,
##     ties = ties, singular.ok = singular.ok, model = model, x = x,
##     y = y)
##
##   n= 1640, number of events= 300
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## D$age      0.04067   1.04151  0.00416  9.776  < 2e-16 ***
## D$female1 -0.62756   0.53389  0.11767 -5.333 9.66e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## D$age        1.0415     0.9601    1.0331    1.0500
## D$female1    0.5339     1.8730    0.4239    0.6724
##
## Concordance= 0.674  (se = 0.017 )
## Likelihood ratio test= 117.8  on 2 df,   p=<2e-16
## Wald test            = 115.2  on 2 df,   p=<2e-16
## Score (logrank) test = 116.4  on 2 df,   p=<2e-16
##
##
## $study3
## Call:
## survival::coxph(formula = formula, data = dataTable, weights = weights,
##     ties = ties, singular.ok = singular.ok, model = model, x = x,
##     y = y)
##
##   n= 2688, number of events= 578
##
##                coef exp(coef)  se(coef)      z Pr(>|z|)
## D$age      0.042145  1.043045  0.003086 13.655  < 2e-16 ***
## D$female1 -0.599238  0.549230  0.084305 -7.108 1.18e-12 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## D$age        1.0430     0.9587    1.0368    1.0494
## D$female1    0.5492     1.8207    0.4656    0.6479
##
## Concordance= 0.699  (se = 0.011 )
## Likelihood ratio test= 227.9  on 2 df,   p=<2e-16
## Wald test            = 228.4  on 2 df,   p=<2e-16
## Score (logrank) test = 229.4  on 2 df,   p=<2e-16
```

7 Meta-analyze hazard ratios

We now outline how the hazard ratios from the survival models are meta-analyzed. We use the *metafor* package for meta-analysis. We show the summary of an example meta-analysis and a forest plot below. The forest plot shows a basic example of meta-analyzed hazard ratios from a survival model (analyzed in DataSHIELD).

The log-hazard ratios and their standard erros from each study can be found after running *ds.coxphSLMA()*

The hazard ratios can then be meta-analyzed:

```
metafor::rma(log_hazard_ratio, sei = se_hazard_ratio, method = 'REML')
```

A summary of this meta-analyzed model is shown below.

```
##
## Random-Effects Model (k = 3; tau^2 estimator: REML)
##
##   logLik  deviance       AIC       BIC      AICc
##   9.3824  -18.7648  -14.7648  -17.3785   -2.7648
##
## tau^2 (estimated amount of total heterogeneity): 0 (SE = 0.0000)
## tau (square root of estimated tau^2 value):      0
## I^2 (total heterogeneity / total variability):   0.00%
## H^2 (total variability / sampling variability):  1.00
##
## Test for Heterogeneity:
## Q(df = 2) = 0.0880, p-val = 0.9569
##
## Model Results:
##
## estimate      se      zval    pval   ci.lb   ci.ub
##   1.0425  0.0020  515.4456  <.0001  1.0385  1.0465  ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We now show a forest plot with the meta-analyzed hazard ratios. The hazard ratios come from the DataSHIELD function *ds.coxphSLMA()*. The hazard ratios are meta-analyzed using the *metafor* package.

8 Plotting of privacy-preserving survival curves

We also plot privacy preserving survival curves.

```
dsBaseClient::ds.survfit(formula='surv_object~1', objectname='survfit_object')

dsBaseClient::ds.plotsurvfit(formula = 'survfit_object')
```

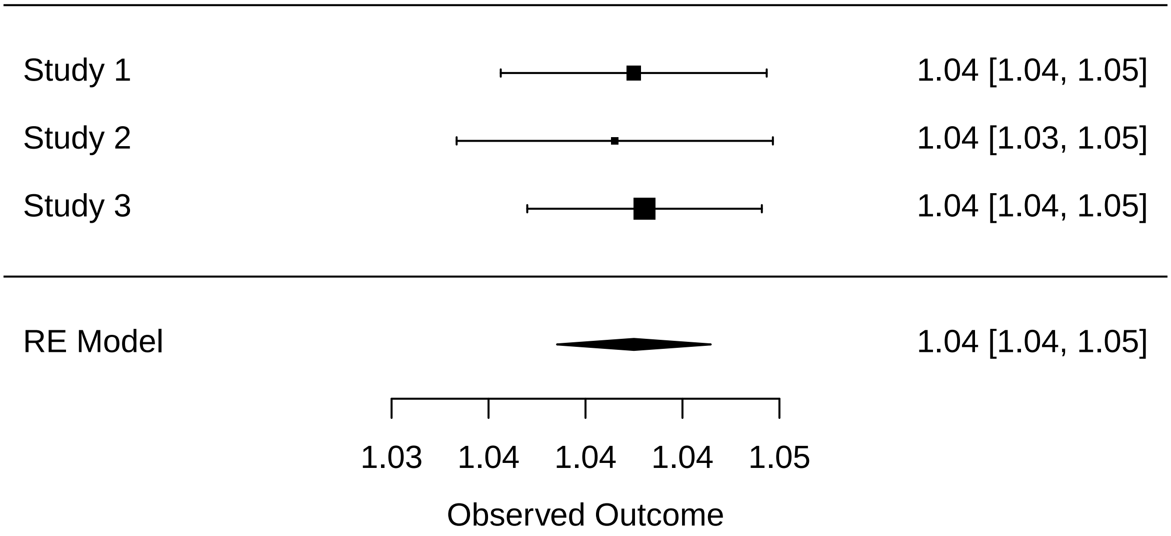
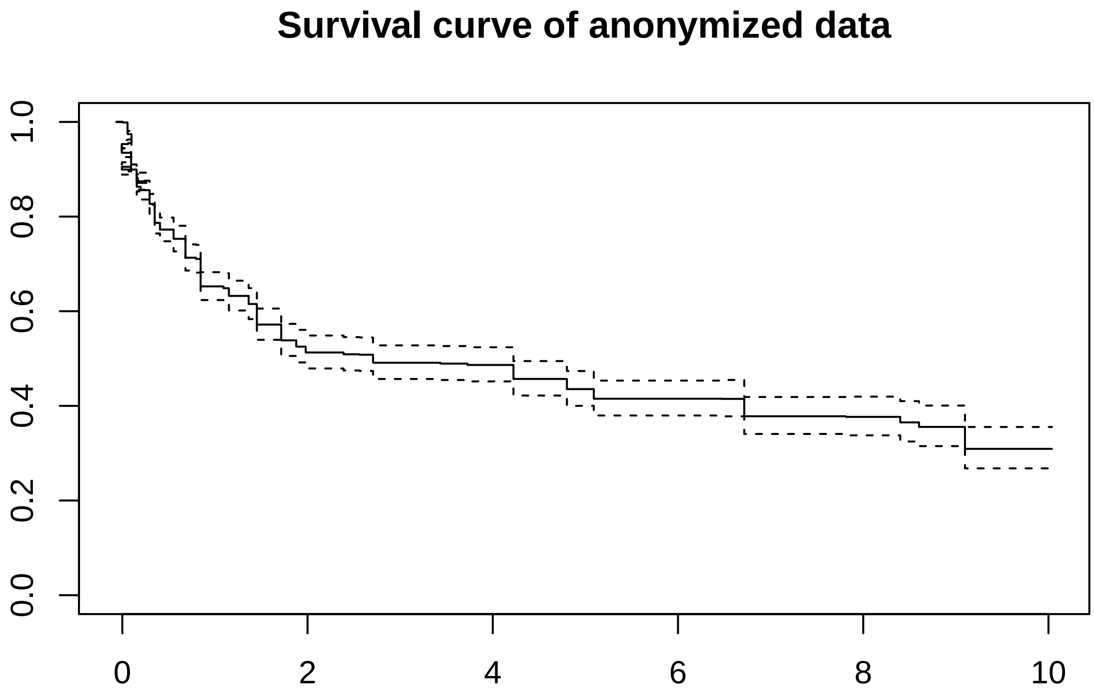


Figure 1: Example forest plot of meta-analyzed hazard ratios.


Survival curve of anonymized data

```
## $study1
## Call: survfit(formula = formula)
##
## records       n  events  median 0.95LCL 0.95UCL
## 2060.00  886.00  426.00    2.71    1.88    4.22
##
## $study2
## Call: survfit(formula = formula)
##
## records       n  events  median 0.95LCL 0.95UCL
## 1640.00  659.00  300.00    3.00    2.12    7.40
##
## $study3
## Call: survfit(formula = formula)
##
## records       n  events  median 0.95LCL 0.95UCL
## 2688.00 1167.00  578.00    1.92    1.86    2.41
```

9 Dealing with preserving privacy and disclosure checks

Disclosure checks are an integral part of DataSHIELD. We outline some of the checks we have built in to ensure privacy of individual level data. We also outline some issues that need to be discussed.

9.1 Disclosure related to oversaturated models

We disallow any Cox models where the number of parameters are greater than a fraction (set to 0.2) of the number of data points. The number of data points is the number of entries (for all patients) in the survival data.

9.2 Summary of Cox models with no individual data

We also present privacy preserving summaries of Cox models and survival objects. These functions reveal no individual level data and present only quantiles.

9.3 Dealing with strata() that could be potentially disclosive

An issue that needs further discussion is how to deal with strata in Cox models. It may be desirable to disallow Cox models with a strata that has less than a specified number of patients (for example 10).

9.4 Relevant work for ensuring survival models are privacy preserving

We briefly outline some relevant work for ensuring survival models are privacy preserving. One work reduces the dimensions of a survival model and the reduced feature space model is then shared amongst multiple parties.

- https://people.csail.mit.edu/romer/papers/PPCox_KDD08.pdf

10 Development plan for additional functions

We now outline additional functionalities that can be implemented. We envisage rolling these out in phased releases.

10.1 Plotting diagnostics for Cox proportional hazards models

- residuals like Schoenfeld residuals plotted over time to test for validity of assumptions in the Cox proportional hazards model
- equivalent of functions like survival::cox.zph()
  - dsBaseClient::ds.plot.cox.zphSLMA()
  - dsBase::plot.cox.zphDS()

10.2 Plotting survival curves that are non-disclosive and privacy preserving

The functions planned are:

- dsBaseClient::ds.survfit()
- dsBase::survfitDS()
- dsBaseClient::ds.plotsurvival()
- dsBaseClient::ds.plotsurvival(survfit_object, fun="cloglog")
- dsBase::plotsurvivalDS()
- use of ggplot() within these functions

10.3 Additional options in coxph()

- additional options in coxph() like control, subset

10.4 Time dependent covariates in Cox proportional hazards models

- facility for inline functions in call to coxph.SLMA()

```
coxph( Surv(time, status) ~ age + sex + tt(ph.karno),
       data = lung,
       tt   = function(x,t,...) x * log(t+20)
     )
```

- other niche examples for time dependent covariates in survival models
    - some examples are given in the following paper:

  https:www.ncbi.nlm.nih.gov/pmc/articles/PMC6015946

    - equivalent of survival::survSplit() as another approach for time dependent covariates

11 Other ideas

- privacy preserving plotting
- privacy preserving summary using quantile means
- technical definition of privacy and disclosure checks for plotting of survival models
    - for example, how does the ordering of events violate privacy?
    - is it disclosive only if combined with other covariates (stratified by age, gender, etc.)?

12 Acknowledgements

We acknowledge the help and support of the DataSHIELD technical team. We are especially grateful to Paul Burton, Demetris Avraam, Stuart Wheater and Patricia Ryser-Welch for fruitful discussions and feedback.

13 References

- https://github.com/datashield
- http://www.metafor-project.org
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6015946