12 ExpoSurv module

12.1 Application domain

ExpoSurvival module is designed to conduct the survival analysis of the censored data. It mainly aims to evaluate the associations between exposure factors and health outcome, as well as predicting the survival probability. Users can easily get the modeling results and their visualization plots with high quality by following the detailed instructions in each step.

12.2 Theory

In "SurvAsso()" function, Cox proportional hazards regression model is used to calculate the hazard ratio (HR). In "SurvPred()" function, various machine learners are used to conduct the prediction analysis, including coxph, coxboost, xgboost, ranger, cforest, ctree, nelson, etc., from mlr3 R package.

12.3 Work pipeline

Install packages

To use Exposurv module, users can install the "exposomex" package, or "exposurv" and "expotidy" packages separately.

```
# The following two packages should be installed in advance
# devtools::install_github("ExposomeX/exsurv", force = TRUE)
# devtools::install_github("ExposomeX/extidy", force = TRUE)

# library(exsurv)
# library(extidy)
library(tidyverse)

# devtools::install_github("ExposomeX/exposomex", force = TRUE)
library(exposomex)
```

Initialize

At first, you need to initialize the calculation environment using initialization function: "InitSurv()".

```
res = InitSurv()
res
```

```
## <eSet>
##   Public:
##     AddCommand: function (x)
##     AddLog: function (x)
##     clone: function (deep = FALSE)
##     ExcecutionLog: Complete initializing the Exoverse module.2022-12-14 14: ...
##     Expo: list
##     ExpoDel: list
##     FileDirOut: /home/ubuntu/@changxin/R_Exposome_1.0/output_145843FDEPNS
##     Func: list
##     PID: 145843FDEPNS
##     RCommandLog: eSet <- InitExpoData(PID = Any ID your like, FileDirOut  ...
```

Here, we can see that the returned value "res" is an R6 object. It contains an unique program ID of res$PID (e.g., "100737GJMWJA"), which is random generated by the system. Users need use it in the following step for further data process.

Upload data

Then, you need to upload data for analysis by runing "LoadSurv()" function. The function includes four parameters:
1. PID: chr. Program ID. It must be the same with the PID generated by InitSurv.
2. UseExample: chr. Method of uploading data. If "default", user should upload their own data files, or use "example#1" provided by this module.
3. DataPath: chr. Input data file directory, e.g. "D:/test/eg_Surv_data.xlsx".
4. VocaPath: chr. Input vocabulary file directory, e.g. "D:/test/eg_Surv_voca.xlsx".

Noted that there are several data format requirments for the data and vocabulary file. For data file,the first three columns should be named as "SampleID", "SubjectID", and "Group", respectively. For the "Group" variable, only two values can be used, i.e. "train" and "test". If there is no data for test, all values should be set as "train". For outcome variables, their initials must be set as "Y" and serialized by adding Arabic numerals if needed, e.g., Y1, Y2, Y3. In this module, the survival time (Y1) and status (Y2) must be provided.For exposure variables, their initials must be set as "X" and serialized by adding Arabic numerals if needed, e.g., X1, X2, X3. For covariate variables, their initials must be set as "C" and serialized by adding Arabic numerals if needed, e.g., C1, C2, C3. It should be noted the covariates are not required if users don't have. For vocabulary file, the first two columns must be named as "SerialNo" and "FullName", respectively. The list of SerialNo of outcomes, exposure, and covariates should be the same with the column names of "Data file". The list of the FullName is prepared as users' like.

Here, we use "example#1" data for demonstration:

```r
res1 <- LoadSurv(PID = res$PID, UseExample = "example#1", DataPath = NULL,
   VocaPath = NULL)
res1$Expo$Data
```

```
## # A tibble: 394 x 13
##    SampleID Subjec~1 Group    Y1    Y2    C1    C2    C3    X1    X2    X3    X4
##       <dbl>    <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1        1        1 train   306     2 209.     74     1     1    90   100 1175
##  2        2        2 train   455     2 213.     68     1     0    90    90 1225
##  3        3        3 train  1010     1 105.     56     1     0    90    90  485.
##  4        4        4 train   210     2 196.     57     1     1    90    60 1150
##  5        5        5 train   883     2 108.     60     1     0   100    90  504.
##  6        6        6 train  1022     1 103.     74     1     1    50    80  513
##  7        7        7 train   310     2  85.1    68     2     2    70    60  384
##  8        8        8 train   361     2 108.     71     2     2    60    80  538
##  9        9        9 train   218     2 149.     53     1     1    70    80  825
## 10       10       10 train   166     2  72.7    61     1     2    70    70  271
## # ... with 384 more rows, 1 more variable: X5 <dbl>, and abbreviated variable
## #   name 1: SubjectID
```

```r
res1$Expo$Voca
```

```
## # A tibble: 10 x 11
##    SerialNo Serial~1 FullN~2 Group~3 Lod   CasRN Inchi~4 KeggC~5 Uniprot Disea~6
##    <chr>    <chr>    <chr>   <chr>   <lgl> <lgl> <lgl>   <lgl>   <lgl>   <lgl>
##  1 Y1       Y1       time    outcome NA    NA    NA      NA      NA      NA
##  2 Y2       Y2       status  outcome NA    NA    NA      NA      NA      NA
##  3 C1       C1       conf    chemic~ NA    NA    NA      NA      NA      NA
##  4 C2       C2       age     chemic~ NA    NA    NA      NA      NA      NA
##  5 C3       C3       sex     chemic~ NA    NA    NA      NA      NA      NA
##  6 X1       X1       ph.ecog chemic~ NA    NA    NA      NA      NA      NA
##  7 X2       X2       ph.kar~ chemic~ NA    NA    NA      NA      NA      NA
##  8 X3       X3       pat.ka~ chemic~ NA    NA    NA      NA      NA      NA
##  9 X4       X4       meal.c~ chemic~ NA    NA    NA      NA      NA      NA
## 10 X5       X5       wt.loss chemic~ NA    NA    NA      NA      NA      NA
## # ... with 1 more variable: PhenotypeID <lgl>, and abbreviated variable names
## #   1: SerialNo_Raw, 2: FullName, 3: GroupName, 4: Inchikey, 5: KeggChemEntry,
## #   6: DiseaseID
```

Tidy data

See ExpoTidy module for more information. You can skip some steps of the "Tidy" part, if you make sure that the data can satisfy the modeling requirement. We recommend that you run 'TransDummy()' to prevent errors like the one below:

If X1 in your data is a categorical variable and you have an X11 variable, then name conflicts will occur during modeling. Because model result will turn X1 into X11, X12, X12…

Find Covariates

Users can run "FindCovaSurv()" to find out which covariates should be included in the model before modeling. The function includes eight parameters:
1. PID: chr. Program ID. It must be the same with the PID generated by InitSurv.
2. OutPath: chr. Output file directory, e.g. "D:/test". If "default", the current working directory will be set.
3. TimeY: chr. Outcome variable of survival time used for modelling. Only one variable can be entered.
4. EventY: chr. Outcome variable of status used for modelling. Only one variable can be entered.
5. VarsC_Prior: chr. Potential covariates needing further statistical test. The default value is all covariate variables listed in the data file.
6. VarsC_Fixed: chr. Covariate variables fixed in the model by users.
7. Method: chr. Methods for screening the covariates, including two options, i.e. "single.factor" and "two.stage".
8. Thr: num. Threshold of the P-value for screening the covariates. It isranging 0.05-0.25. The default value is 0.1.

```r
res2 = FindCovaSurv(PID = res$PID,
                    TimeY = "Y1",
                    EventY= 'Y2',
                    VarsC_Prior = "default",
                    VarsC_Fixed = NULL,
                    Method = "single.factor",
                    Thr = 0.1)

res2$FindCovaSurv
```

```
## # A tibble: 2 x 1
##   covariates
##   <chr>
## 1 C2
## 2 C3
```

Association

Now, users can calculate the hazard ratio(HR) of exposures variables by running "SurvAsso()" function. The function includes eight parameters:
1. PID: chr. Program ID. It must be the same with the PID generated by InitSurv.
2. OutPath: chr. Output file directory, e.g. "D:/test". If "default", the current working directory will be set.
3. TimeY: chr. Outcome variable of survival time used for modelling. Only one variable can be entered.
4. EventY: chr. Outcome variable of status used for modelling. Only one variable can be entered.
5. VarsX: chr. Exposure variable used for modeling. The default option is "all.x" (All exposure variables are

included). Users can also choose available variables. It should be noted that there is fixed format for the entering characters separated with comma and without space, e.g., “X1,X2,X3”
6. VarsN: chr. Choose the single factor or multiple factor model. Available options include “single.factor” and “multiple.factor”
7. VarsSel: lgl. T (or TRUE) and F (or FALSE). Whether to select the significant variable for the final model. Available options includes T and F.
8. IncCova: lgl. T (or TRUE) and F (or FALSE). Whether to include the covariate selected in the function “FindCovaSurv”.

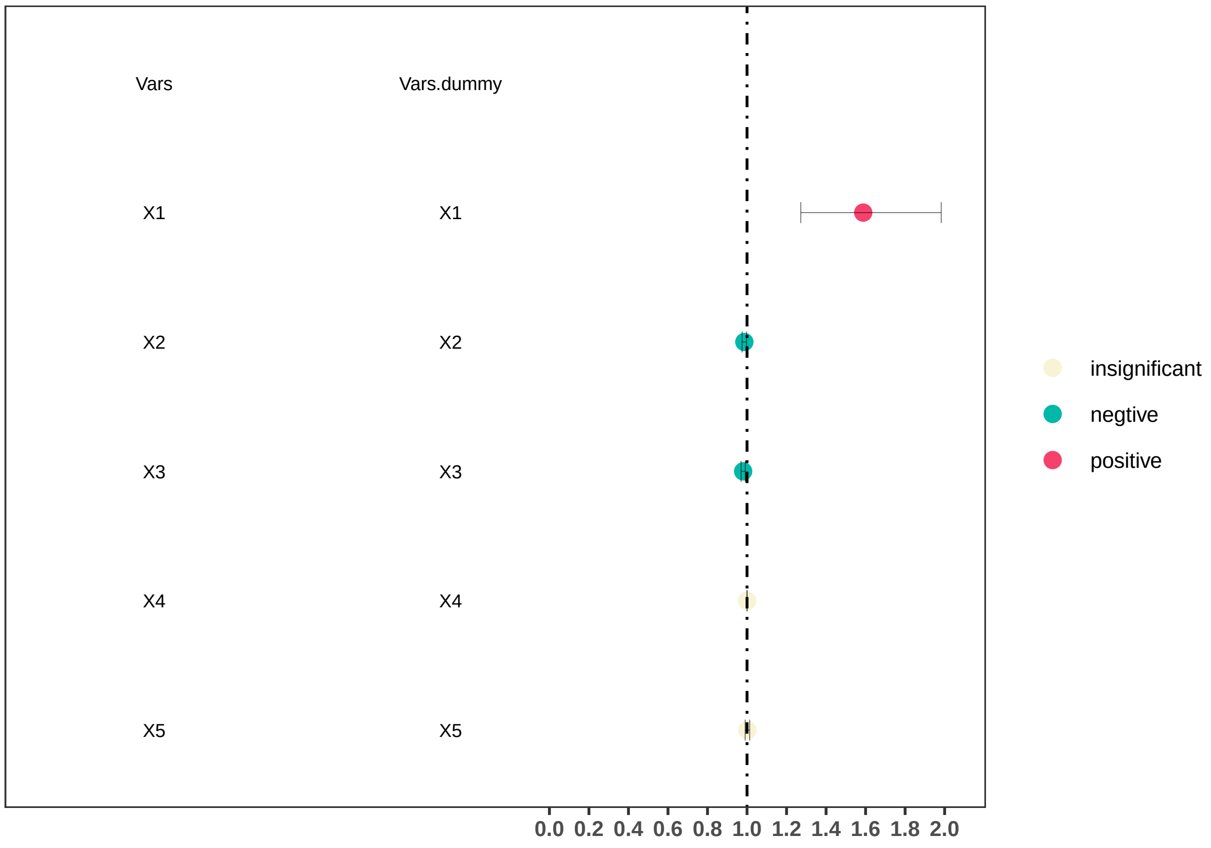
```
res3 = SurvAsso(PID = res$PID,
                TimeY = "Y1",
                EventY= 'Y2',
                VarsX='all.x',
                VarsN="single.factor",
                VarsSel=T,
                IncCova = T)
res3$coxph_res
```

```
## # A tibble: 5 x 8
##   Vars  Vars.dummy    HR  CI_L  CI_H   p.value cindex logloss
##   <chr> <chr>      <dbl> <dbl> <dbl>     <dbl>  <dbl>   <dbl>
## 1 X1    X1         1.59  1.27  1.98  0.0000442  0.637    4.17
## 2 X2    X2         0.986 0.975 0.998 0.0201     0.640    4.21
## 3 X3    X3         0.980 0.970 0.991 0.000327   0.643    4.16
## 4 X4    X4         1.00  1.00  1.00  0.502      0.607    4.23
## 5 X5    X5         1.00  0.990 1.01  0.741      0.601    4.24
```

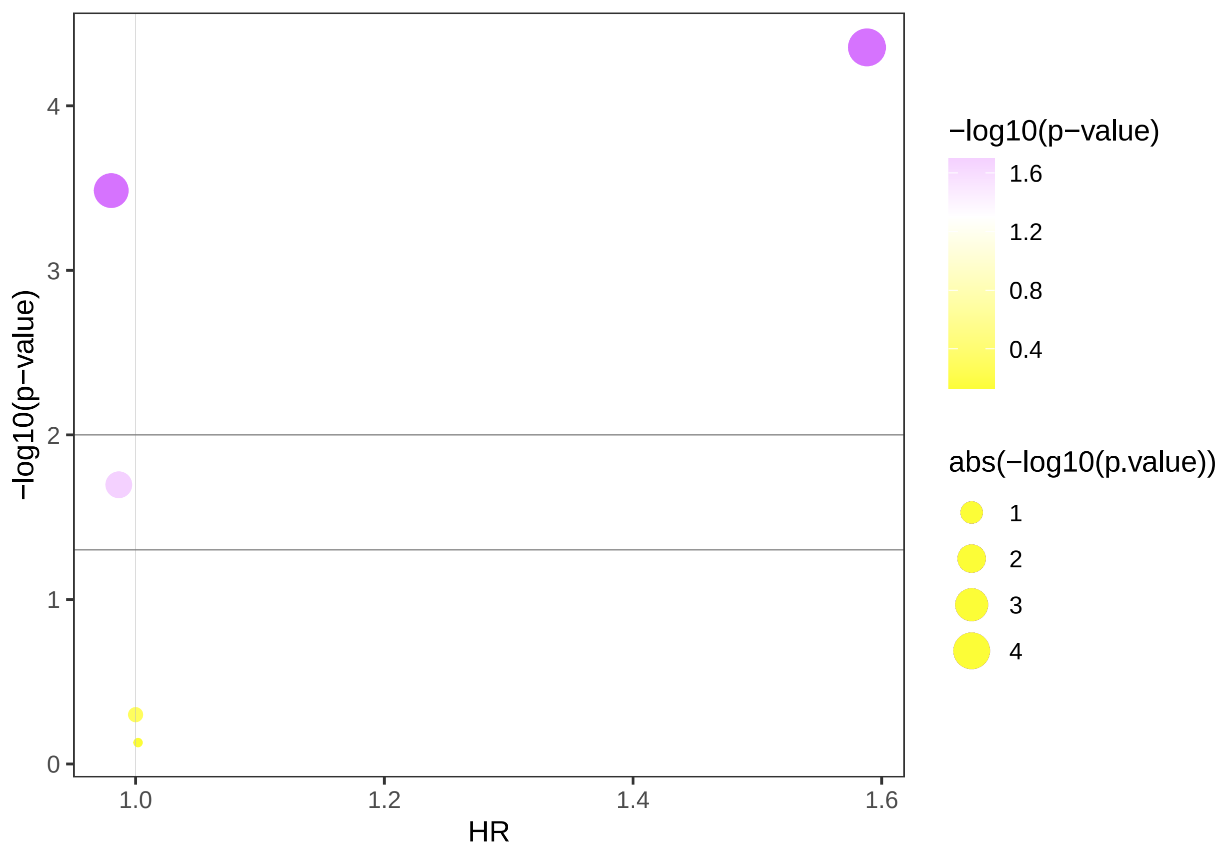
Viz Association

After running association model, users can visualize the results by running “VizSurvAsso()” function. The function includes eight parameters:
1. PID: chr. Program ID. It must be the same with the PID generated by InitSurv.
2. OutPath: chr. Output file directory, e.g. “D:/test”. If “default”, the current working directory will be set.
3. VarsN: chr. Choose the single factor or multiple factor model. Available options include “single.factor” and “multiple.factor”. It must be same as the ‘VarsN’ in Association function.
4. Layout: chr. Visualization layout. Available options include “forest” and “volcano”.
5. Brightness: chr. Visualization brightness. Available options include “light” and “dark”.
6. Palette: chr. Visualization palette. Available options include “default1”, “default2” and several journal preference styles (i.e., “cell”, “nature”, “science”, “lancet”, “nejm”, and “jama”).
7. ColorFor: chr. Volcano plot dot color. Available options include “p.value” and “hr”.
8. SizeFor: chr. Volcano plot dot size. Available options include “p.value” and “hr”.

```
res5 = VizSurvAsso(PID = res$PID,
                   VarsN="single.factor",
                   Layout="forest",
                   Brightness= "light",
                   Palette = "default1")
res5$ForestPlot
```



```
res6 = VizSurvAsso(PID = res$PID,
                   VarsN="single.factor",
                   Layout="volcano",
                   Brightness= "light",
                   Palette = "default1",
                   ColorFor= "p.value",
                   SizeFor= "p.value")
res6$VolcanoPlot
```



Prediction

In addition, users can predict the survival curve by "SurvPred()" function. The function includes ten parameters:
1. PID: chr. Program ID. It must be the same with the PID generated by InitSurv.
2. OutPath: chr. Output file directory, e.g. "D:/test". If "default", the current working directory will be set.
3. TimeY: chr. Outcome variable of survival time used for modelling. Only one variable can be entered.
4. EventY: chr. Outcome variable of status used for modelling. Only one variable can be entered.
5. VarsX: chr. Exposure variable used for modeling. The default option is "all.x" (All exposure variables are included). Users can also choose available variables. It should be noted that there is fixed format for the entering characters separated with comma and without space, e.g., "X1,X2,X3" 6. IncCova: lgl. Whether to include the covariate selected in the function of "FindCovaSurv". Available options include T (or TRUE) and F (or FALSE).
7. RsmpMethod: chr. Three resampling methods options for internal validation, including "cv" (i.e.,Cross validation), "bootstrap", and "holdout".
8. Folds: num. Folds of Cross-validation resampling. It is ranging 2-10.
9. Ratio: num. Ratio of Bootstrap resampling. It is ranging 0.4-0.9.
10. Repeats: num. Number of Bootstrap resampling. It is ranging 2-20.

```
res7 = SurvPred(PID = res$PID,
                TimeY = "Y1",
                EventY = 'Y2',
                VarsX ='all.x',
                IncCova = T,
                RsmpMethod ="cv",
                Folds = 3)
```

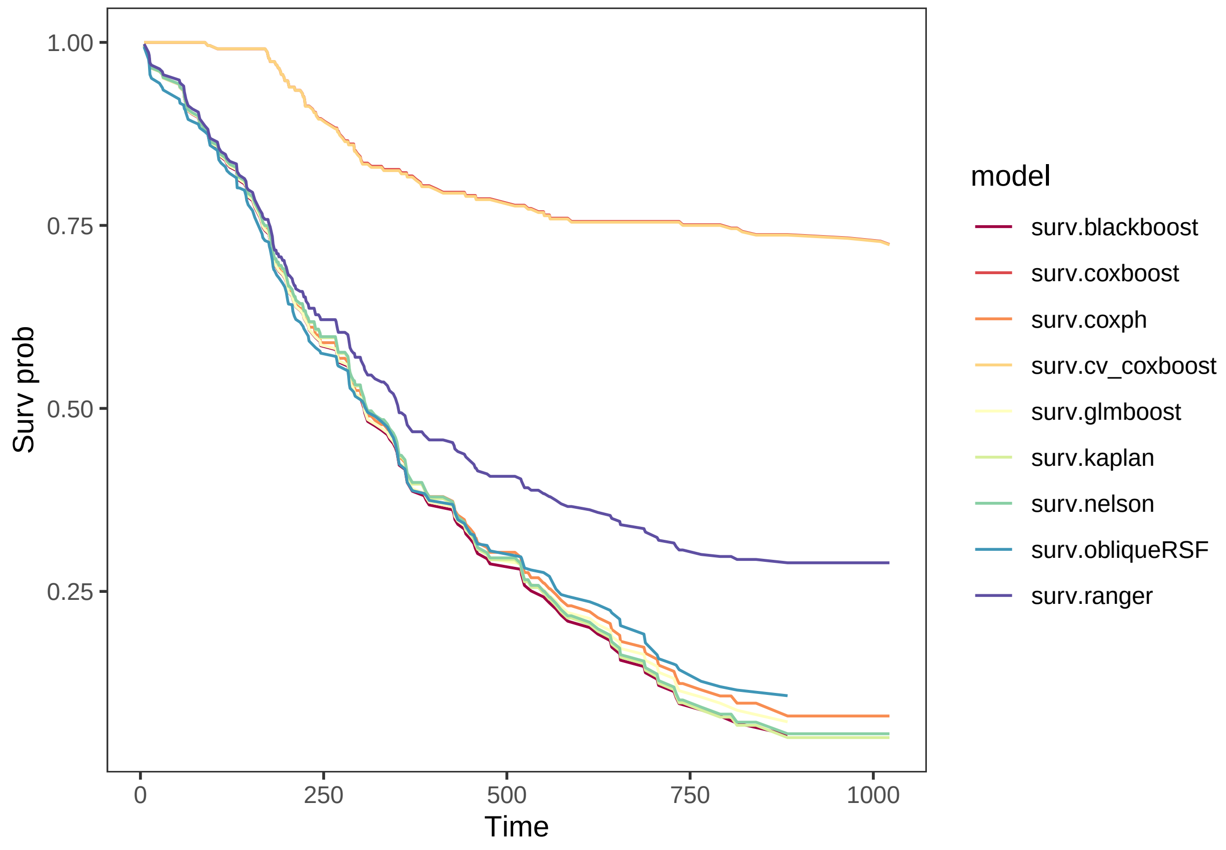
```
res7$bmrout
```

```
## # A tibble: 14 x 7
##    learner_id       cindex_rsmp_train_~1 cinde~2 cinde~3 cinde~4 cinde~5 cinde~6
##    <chr>                           <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 surv.blackboost                 0.737   0.598 0.00614 0.0302    0.709   0.718
##  2 surv.cforest                    0.723   0.602 0.00688 0.00907   0.723   0.729
##  3 surv.coxboost                   0.338   0.388 0.00483 0.0213    0.343   0.332
##  4 surv.coxph                      0.663   0.609 0.00802 0.0226    0.651   0.660
##  5 surv.ctree                      0.605   0.520 0.00468 0.0135    0.622   0.617
##  6 surv.cv_coxboost                0.391   0.418 0.0384  0.0477    0.357   0.346
##  7 surv.gbm                        0.701   0.580 0.00248 0.0142    0.679   0.684
##  8 surv.glmboost                   0.669   0.612 0.00591 0.0145    0.660   0.671
##  9 surv.kaplan                     0.5     0.5   0       0         0.5     0.5
## 10 surv.nelson                     0.5     0.5   0       0         0.5     0.5
## 11 surv.obliqueRSF                 0.703   0.580 0.0175  0.0226    0.711   0.715
## 12 surv.parametric                 0.663   0.609 0.00956 0.0214    0.651   0.660
## 13 surv.ranger                     0.862   0.569 0.00498 0.0382    0.857   0.866
## 14 surv.xgboost                    0.432   0.467 0.0386  0.0547    0.415   0.395
## # ... with abbreviated variable names 1: cindex_rsmp_train_mean,
## #   2: cindex_rsmp_test_mean, 3: cindex_rsmp_train_sd, 4: cindex_rsmp_test_sd,
## #   5: cindex_train, 6: cindex_test
```

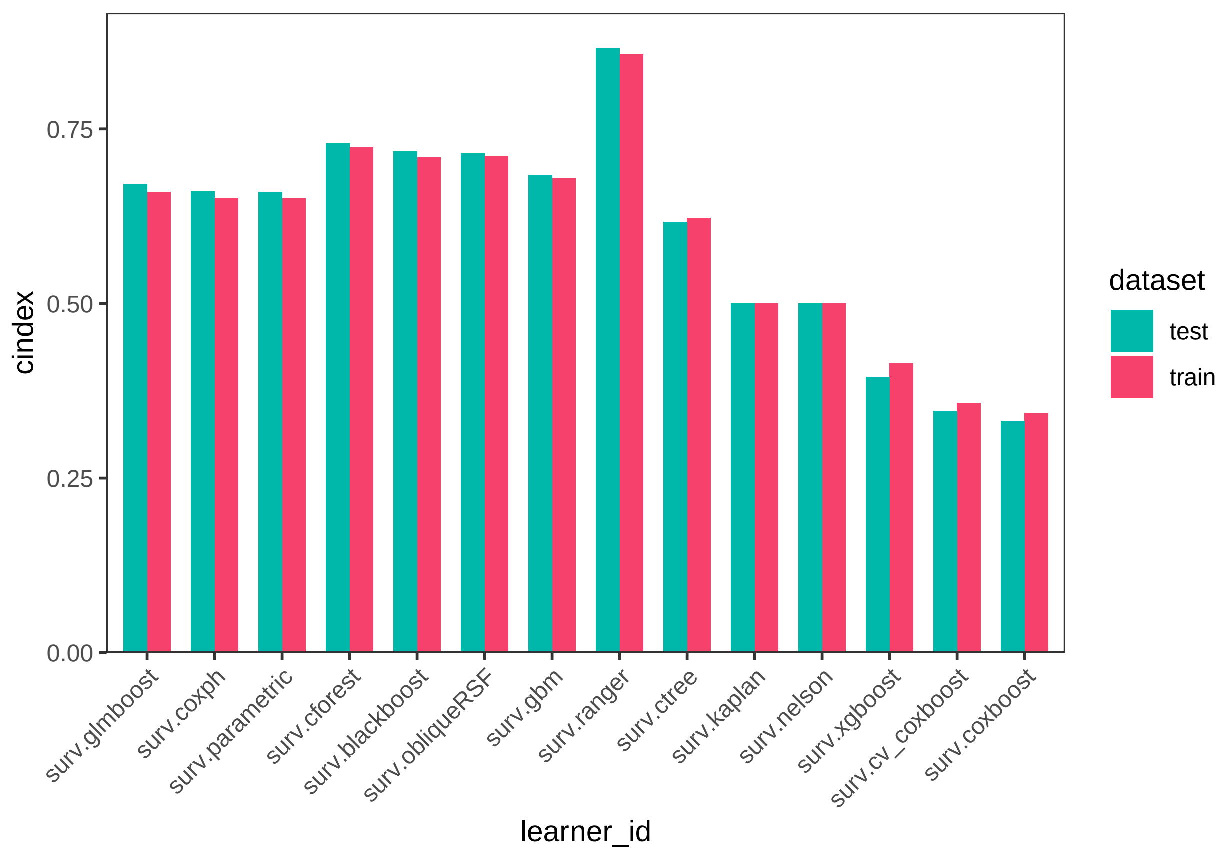
Viz Prediction

After running SurvPred model, users can visualize the results by running "VizSurvPred()". The function includes five parameters:
1. PID: chr. Program ID. It must be the same with the PID generated by InitSurv.
2. OutPath: chr. Output file directory, e.g. "D:/test". If "default", the current working directory will be set.
3. Layout: chr. chr. Visualization layout. Available options include "curve" , "bar" and 'all'.
4. Brightness: chr. Visualization brightness. Available options include "light" and "dark".
5. Palette: chr. Visualization palette. Available options include "default1", "default2" and several journal preference styles (i.e., "cell", "nature", "science", "lancet", "nejm", and "jama").

```
res8 = VizSurvPred(PID = res$PID,
                   Layout="all",
                   Brightness="light",
                   Palette='default1')
res8$PredSurvCurvPlot
```



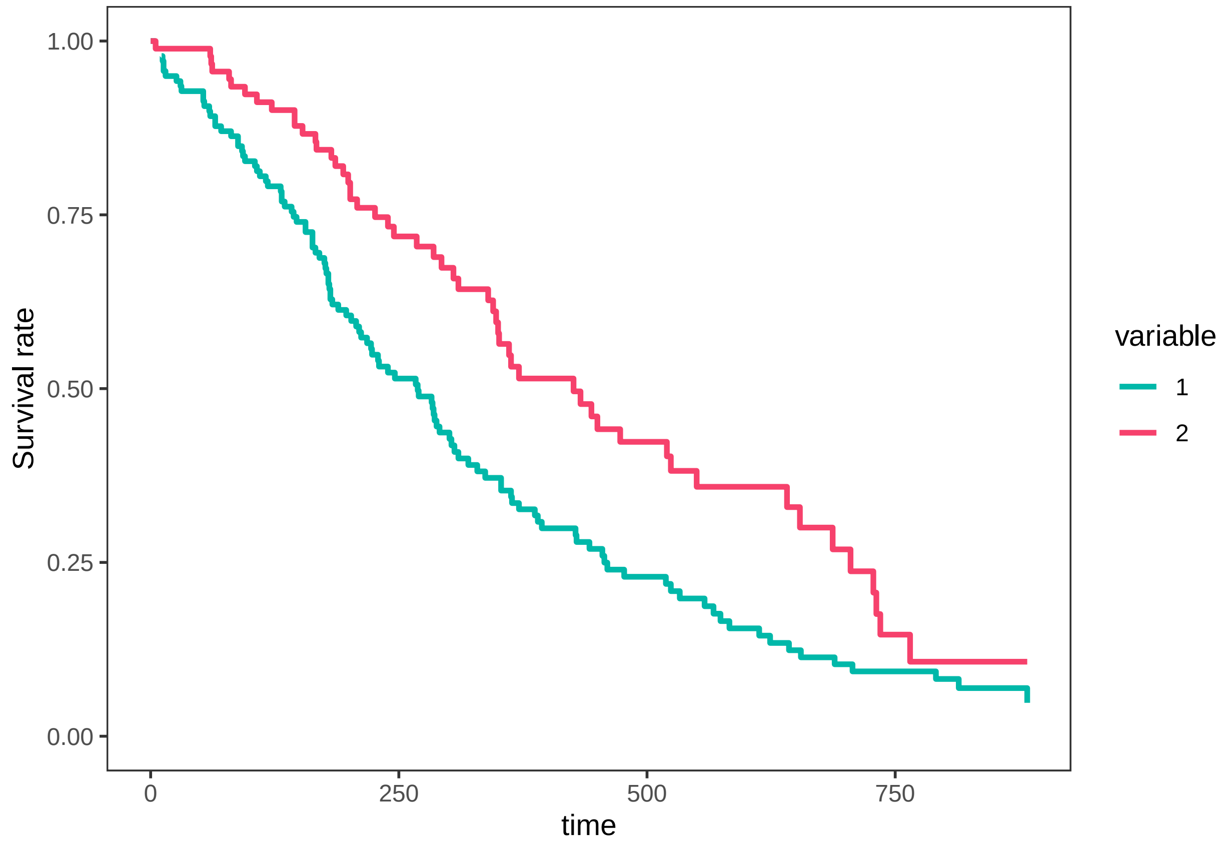
```
res8$BmrBarPlot
```



VizSurvCompGroup

We also provide a function to compare the survival curves of two groups. This function does not depend on the previous functions result. Users can run it after "LoadSurv()". The function includes ten parameters:
1. PID: chr. Program ID. It must be the same with the PID generated by InitSurv.
2. OutPath: chr. Output file directory, e.g. "D:/test". If "default", the current working directory will be set.
3. TimeY: chr. Outcome variable of survival time used for modelling. Only one variable can be entered.
4. EventY: chr. Outcome variable of status used for modelling. Only one variable can be entered.
5. VarsG: chr. Grouping variable, must be binary variables. It should be noted that there is fixed format for the entering characters separated with comma and without space, e.g., "X1,X2,X3" 6. Model: chr. Methods to depict the survival curve. Options include 'km' (Kaplan-Meier estimate) and "coxph" (Cox proportional hazards regression mode).
7. VarsAdj: chr. If you choose the cox model, co-variables used for modelling. It should be noted that there is fixed format for the entering characters separated with comma and without space, e.g., "X1,X2,X3".
8. AdjMethod: chr. If you choose the coxph model, method for adjusting model, includes: "average","single","margin" and "conditional".
9. Brightness: chr. Visualization brightness. Available options include "light" and "dark".
10. Palette: chr. Visualization palette. Available options include "default1", "default2" and several journal preference styles (i.e., "cell", "nature", "science", "lancet", "nejm", and "jama").

```
res9 = VizSurvCompGroup(PID = res$PID,
                        TimeY="Y1",
                        EventY="Y2",
                        VarsG="C3",
                        Model="coxph",
                        VarsAdj='X1,X2,X3,X4,X5',
                        AdjMethod='average',
                        Brightness="light",
                        Palette='default1')
res9$Comp_Coxph_Plot[[1]]
```



Exit

After all the analysis is done, please run the "FuncExit()" function to delete the data uploaded to the server.

```
FuncExit(PID = res$PID)
```

```
## [1] "Success to exit. Thanks for using ExposomeX platform!"
```