

Package: JPSurv (via r-universe)

May 15, 2026

Version 3.0.20

Date 2025-05-15

Title Joinpoint Model for Relative and Cause-Specific Survival

Description Contains functions for fitting a joinpoint proportional hazards model to relative survival or cause-specific survival data, including estimates of joinpoint years at which survival trends have changed and trend measures in the hazard and cumulative survival scale. See Yu et al.(2009) [doi:10.1111/j.1467-985X.2009.00580.x](https://doi.org/10.1111/j.1467-985X.2009.00580.x).

License GPL (>= 2)

Imports ggplot2,ggrepel, scales

Depends R (>= 3.5.0)

NeedsCompilation no

Author Angela Mariotto [aut], Theresa Devasia [aut], Yongwu Shao [aut], Jun Luo [ctb], Daniel Miller [ctb], Fanni Zhang [aut], Bill Wheeler [cre]

Maintainer Bill Wheeler <wheelerb@imsweb.com>

Repository <https://wheelerb.r-universe.dev>

Date/Publication 2025-05-19 14:20:09 UTC

RemoteUrl <https://github.com/cran/JPSurv>

RemoteRef HEAD

RemoteSha 10049edffe7c7fd20f52a286dab2a9b61a278094

Contents

JPSurv-package	2
aapc	3
aapc.multiints	4
breast.example	5
dictionary.overview	5
download.data	6

fit1	7
fit2	7
fit3	7
input.valid	8
joinpoint	8
joinpoint.choose.cutpoint	11
joinpoint.cond	12
joinpoint.conditional	14
joinpoint.relaxProp	15
joinpoint.seerdata	18
jp.plot.cond.year	19
jp.plot.death	20
jp.plot.surv	21
Plot.dying.year.annotate	23
Plot.surv.int.multiyears	24
Plot.surv.year.annotate	26
Index	28

JPSurv-package

Joinpoint Model for Relative and Cause-Specific Survival

Description

This package is to analyze trends in survival with respect to year of diagnosis using joinpoint model.

Details

Survival data includes two temporal dimensions that are important to consider: the calendar year of diagnosis and the time since diagnosis. The JPSurv model is an extension of Cox proportional hazard model and of Hakulinen and Tenkanen in the case of relative survival, and fits a proportional hazard joinpoint model to survival data on the log hazard scale. Joinpoint models consist of linear segments connected through joinpoints. The probability (hazard) of cancer death is specified as the product of a baseline hazard (on time since diagnosis) and a multiplicative factor describing the effect of year of diagnosis and possibly other covariates. The effect of year of diagnosis is modeled as joined linear segments on the log scale. The number and location of joinpoints are estimated from data and represent the times at which trends changed. This model implies that the probability of cancer death as a function of time since diagnosis is proportional for individuals diagnosed in different calendar years. The software uses discrete-time survival data, i.e. survival data grouped by years since diagnosis in the life table format. The package accommodates both relative survival and cause-specific survival.

References

Yu B, Huang L, Tiwari RC, Feuer EJ, Johnson KA. Modelling population-based cancer survival trends by using join point models for grouped survival data. *Journal of the Royal Statistical Society Series a-Statistics in Society*. 2009;172:405-25.

Hakulinen T, Tenkanen L. Regression Analysis of Relative Survival Rates. *Applied Statistics*. 1987;36(3):309-17.

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009.

aapc

Trend summary measures for joinpoint relative survival model

Description

Get the trend summary measures for joinpoint relative survival model quickly using default interval=5 (if available). A function that returns the trend summary measures including the Absolute Change in Survival (ACS), the Relative Change in Survival (RCS) . and the Relative Change in Hazard (RCH) for each joinpoint segment

Usage

```
aapc(fit, type="AbsChgSur", interval=5)
```

Arguments

<code>fit</code>	Joinpoint object with predicted values and joinpoint model selections.
<code>type</code>	Type of trend summary measure. Supported measures are: <code>RelChgHaz</code> - Hazard of cancer death, <code>AbsChgSur</code> - Absolute change in survival. <code>RelChgSur</code> - Relative change in survival. The default is <code>AbsChgSur</code> .
<code>interval</code>	The number of years since diagnosis (follow-up years).

Value

A data frame with columns `start.year`, `end.year`, `estimate`, `std.error`, `lowCI`, `upCI` containing the estimates, standard errors, and confidence interval of the trend summary measure.

Examples

```
data("fit2", package="JPSurv")

# Get the estimate, standard error, and confidence interval of
# the annual changes of cumulative relative survival for interval=5.
aapc(fit2, type="RelChgHaz")
```

aapc.multiints

Trend summary measures for multiple selected intervals

Description

A function that returns the trend summary measures including the Absolute Change in Survival (ACS), the Relative Change in Survival (RCS) and the Relative Change in Hazard (RCH) for each joinpoint segment given multiple selected intervals. It can also return the Weighted Average Absolute Change in Survival for the user-specified ACS year range.

Usage

```
aapc.multiints(fit, type = "AbsChgSur", int.select = NULL,
               ACS.range = NULL, ACS.out = NULL)
```

Arguments

fit	Joinpoint object with predicted values and joinpoint model selections.
type	Type of trend summary measure. Supported measures are: RelChgHaz - Hazard of cancer death, AbsChgSur - Cumulative Relative Survival Average Annual Absolute Change. RelChgSur - Cumulative Relative Survival Average Annual Relative Change. The default is AbsChgSur.
int.select	The single or multiple interval values selected for the trend measures calculation. The default is NULL.
ACS.range	It is needed for type="AbsChgSur" only. The minimum and maximum of the year range specified for the Average Absolute Change in Survival trend measure. The default is NULL. If ACS.range=NULL, the average absolute change in survival for joinpoint segments will be returned. Otherwise, the weighted average absolute change in survival for the desired range of years along with the the average absolute change in survival for joinpoint segments will be produced. Note that, the ACS.range should be within the entire range of data.
ACS.out	It is used to specify which ACS trend results will be returned when type="AbsChgSur". ACS.out can be defined as "user" or "both". If ACS.range=NULL, ACS.out should always be NULL and the ACS trend results between joinpoints will be returned. If ACS.range is the year range, the ACS trend measure results between the selected years will be returned when ACS.out="user" and the ACS trend results for both between joinpoints and user-selected years will be returned when ACS.out="both". The default is ACS.out=NULL.

Value

A list of data frames corresponding to each interval from int.select. Each data frame contains the columns start.year, end.year, estimate, std.error, lowCI, upCI for the estimates, standard errors, and confidence interval of the trend summary measure.

Author(s)

Fanni Zhang <zhangf@imsweb.com>

See Also

[aapc](#)

Examples

```
data("fit1", package="JPSurv")  
  
aapc.multiints(fit1, type="RelChgHaz", int.select=c(1,3,5))
```

breast.example	<i>Data for examples</i>
----------------	--------------------------

Description

Data frame containing breast cancer data used in examples.

Examples

```
data("breast.example", package="JPSurv")  
  
# Display first two rows  
breast.example[1:2,]
```

dictionary.overview	<i>SEERStat Dictionary Overview</i>
---------------------	-------------------------------------

Description

Reads the dictionary file and returns a parsed overview.

Usage

```
dictionary.overview(DICfilename)
```

Arguments

DICfilename File name/path for dictionary (.dic) file corresponding to SEERStat data file.

Value

A list which contains the parsed information stored in the dictionary file.

download.data *Combine inputs and outputs*

Description

A function that returns a merged data set including the selected cohort input data and the accompanying data for plot.surv.year/plot.dying.year for the download feature in the JPSurv web app.

Usage

```
download.data(input, fit, nJP, yearvar, downloadtype, subset = NULL,
              interval = "Interval", int.select = NULL)
```

Arguments

input	The input dataset read in by function joinpoint.seerdata.
fit	The joinpoint object containing the model output.
nJP	The number of joinpoints in the model.
yearvar	The variable name for year of diagnosis used in argument 'year' of the function joinpoint.
downloadtype	Either "graph" for graph data or "full" for full data.
subset	An optional string specifying a subset of observations used in the fitting process.
interval	The variable name for year since diagnosis. The default is 'Interval'.
int.select	The interval values selected for the plot if the downloadtype="graph". The default is NULL.

Value

A data frame containing the selected cohort input data and the below predicted columns.

Predicted_Survival_Int	The predicted interval survival.
Predicted_ProbDeath_Int	The predicted probability of Dying of Cancer.
Predicted_Survival_Cum	The predicted cumulative survival.
Predicted_Survival_Int_SE	The standard error of the predicted interval survival.
Predicted_Survival_Cum_SE	The standard error of the predicted probability of Dying of Cancer.
Predicted_ProbDeath_Int_SE	The standard error of the predicted cumulative survival.

Author(s)

Fanni Zhang <zhangf@imsweb.com>

References

The JPSurv web app <https://analysisistools.nci.nih.gov/jpsurv/>.

See Also

[Plot.dying.year.annotate](#), [Plot.surv.year.annotate](#), [Plot.surv.int.multiyears](#).

Examples

```
data("breast.example", package="JPSurv")
data("fit1", package="JPSurv")

nJP<-2
data.graph135<-download.data(breast.example,fit1,nJP,"Year_of_diagnosis_1975","graph",
                             subsetStr1,interval="Interval",int.select=c(1,3,5))
data.full<-download.data(breast.example,fit1,nJP,"Year_of_diagnosis_1975","full",subsetStr1)
```

fit1	<i>Fitted joinpoint object and subset string</i>
------	--

Description

An object of class "joinpoint" used in examples.

fit2	<i>Fitted joinpoint object and subset string</i>
------	--

Description

An object of class "joinpoint" used in examples.

fit3	<i>Fitted joinpoint object and subset string</i>
------	--

Description

An object of class "joinpoint" used in examples.

input.valid	<i>Input data validation</i>
-------------	------------------------------

Description

A function that validates the selected cohort data and returns an indicator along with a warning message if the selected cohort is not valid.

Usage

```
input.valid(input, subset = NULL)
```

Arguments

input	Input dataset imported by function <code>joinpoint.seerdata</code> .
subset	An optional string specifying a subset of observations used in the fitting process.

Value

If the cohort selection is available, value 1 will be returned; otherwise, value 0 will be returned and a warning message printed.

joinpoint	<i>Fitting a join point relative survival model</i>
-----------	---

Description

Fitting a joinpoint relative survival model

Usage

```
joinpoint(data, subset=NULL, na.action = na.fail,
  year="Year", interval="Interval",
  number.event="Died", number.alive="Alive_at_Start",
  number.loss="Lost_to_Followup",
  expected.rate="Expected_Survival_Interval", observedrelsurv = NULL,
  model.form = NULL, maxnum.jp = 0, proj.year.num=5,
  op=list(),
  delLastIntvl=FALSE, add.data.cols="_ALL_")
```

Arguments

<code>data</code>	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> .
<code>subset</code>	an optional string specifying a subset of observations to be used in the fitting process.
<code>na.action</code>	how NAs are treated. The default is first, any <code>na.action</code> attribute of data, second a <code>na.action</code> setting of options, and third <code>na.fail</code> if that is unset. The "factory-fresh" default is <code>na.omit</code> . Another possible value is <code>NULL</code> . The default is <code>na.fail</code>
<code>year</code>	It is about the year or time values and could be a vector of numeric or a character string giving a column name of the argument 'data'. The default is the string 'Year'.
<code>interval</code>	It is about the time elapsed from start to the event and could be a vector of numeric or a character string giving a column name of the argument 'data'. The default is the string 'Interval'.
<code>number.event</code>	It is the number of events or died and could be a vector of numeric or a character string giving a column name of the argument 'data'. The default is the string 'Died'.
<code>number.alive</code>	It is about the number of alive and could be a vector of numeric or a character string giving a column name of the argument 'data'. The default is the string 'Alive_at_Start'.
<code>number.loss</code>	It is about the number of <code>Lost_to_Followup</code> and could be a vector of numeric or a character string giving a column name of the argument 'data'. The default is the string 'Lost_to_Followup'.
<code>expected.rate</code>	It is about the interval expected survival and could be a vector of numeric or a character string giving a column name of the argument 'data'. The default is the string 'Expected_Survival_Interval'. If this column does not exist, then a column of ones will be created for <code>expected.rate</code> .
<code>observedrelsurv</code>	It is about the observed cumulative relative survival and could be a vector of numeric or a character string giving a column name of the argument 'data'. If <code>NULL</code> , then no observed values. The default is <code>NULL</code> .
<code>model.form</code>	an object of class "formula": a symbolic description of covariates. Example: <code>~1+age+as.factor(stage)</code>
<code>maxnum.jp</code>	The maximum number of join points allowed. The default is zero, which is equivalent to a proportional hazard relative survival model.
<code>proj.year.num</code>	The number of projection years for use in the prediction step. Default value is 5 years, with a valid range of 0 to 30 years.
<code>op</code>	List of more options. Details — <ul style="list-style-type: none"> • numbetwn: integer value, number of skipped obs between joinpoints exclusive (not count for the joinpoints). Default is 2. • numfromstart: integer value, number of skipped obs from the first obs to joinpoints exclusive (not count for the joinpoint). Default is 3.

- **numtoend**: integer value, number of skipped obs from the first obs to joinpoints exclusive (not count for the joinpoint). Default is 4.
- delLastIntvl an logical value indicating whether or not deleting records of last intervals of all years. The default is false.
- add.data.cols Character vector of column names in data to add onto the returned data frames of results. Use "_ALL_" to add all columns and use NULL to not add any columns. The default is "_ALL_".

Value

An object of class "joinpoint" will be returned with attributes:

- coefficients a named vector of coefficients and standard errors
- jp the estimates of the join points
- converged convergence status
- predicted the fitted relative survival rates
- fullpredicted the full output matrix, with all year/interval combinations and projections
- xbeta the linear predictor
- ll log likelihood
- aic AIC
- bic BIC
- FitList a list that contains fitting results for the number of joinpoints = 0,1,...,numJPoints respectively.

References

Yu, B., Huang, L., Tiwari, R. C., Feuer, E. J. and Johnson, K. A. (2009), Modeling population-based cancer survival trends by using join point models for grouped survival data. *Journal of the Royal Statistical Society: Series A*, 172, 405-425.

Examples

```
#Load the provided SEER 18 breast cancer example data.
data("breast.example", package="JPSurv")

subsetStr="Year_of_diagnosis_1975 >= 1975 & Age_groups == '00-49' & Breast_stage == 'Localized'"
# Fit the survival join point model with zero join points,
# i.e., fit the proportional hazard relative survival model.
fit = joinpoint(data=breast.example,
               subset = subsetStr,
               year="Year_of_diagnosis_1975",
               observedrelsurv="Relative_Survival_Cum",
               model.form = NULL,
               maxnum.jp = 0)
```

`joinpoint.choose.cutpoint`*Choose cutpoint from `joinpoint.relaxProp` results*

Description

Choose the cutpoint from the `joinpoint.relaxProp` returned object

Usage

```
joinpoint.choose.cutpoint(obj, cutpoint)
```

Arguments

<code>obj</code>	Returned object from joinpoint.relaxProp .
<code>cutpoint</code>	The cutpoint to choose. It should correspond to a set of results in <code>obj\$results</code> . If not, then an error will be thrown.

Value

A list of class with the following objects:

<code>predicted</code>	The fitted relative survival rates
<code>fullpredicted</code>	The full output matrix, with all year/interval combinations and projections
<code>fit.uncond</code>	The fitted model from the (unconditional) joinpoint model corresponding to the cutpoint.
<code>fit.cond</code>	The fitted model from the conditional joinpoint model corresponding to the cutpoint.

References

Yu, B., Huang, L., Tiwari, R. C., Feuer, E. J. and Johnson, K. A. (2009), Modeling population-based cancer survival trends by using join point models for grouped survival data. *Journal of the Royal Statistical Society: Series A*, 172, 405-425.

See Also

[joinpoint.relaxProp](#)

joinpoint.cond

*Fitting a join point conditional relative survival model***Description**

Fitting a joinpoint conditional relative survival model

Usage

```
joinpoint.cond(data, subset, start.interval, end.interval=NULL,
  year="Year", interval="Interval",
  number.event="Died", number.alive="Alive_at_Start",
  number.loss="Lost_to_Followup",
  expected.rate="Expected_Survival_Interval",
  model.form=NULL, maxnum.jp=0, proj.year.num=5,
  op=list(), delLastIntvl=FALSE, add.data.cols="_ALL_")
```

Arguments

<code>data</code>	Data frame containing all variables in the model.
<code>subset</code>	A logical vector of length <code>nrow(data)</code> , a character string or <code>NULL</code> to include a particular subset of data in the analysis. See Details and Examples .
<code>start.interval</code>	A positive integer giving the number of intervals to condition on.
<code>end.interval</code>	A positive integer $>$ <code>start.interval</code> giving the end number of intervals.
<code>year</code>	Column name of data giving the year or year code. This column must be numeric. The default is 'Year'.
<code>interval</code>	Column name of data giving the time interval elapsed from the starting time to the event time. This column must be numeric. The default is 'Interval'.
<code>number.event</code>	Column name of data giving the number of events or deaths. This column must be numeric. The default is 'Died'.
<code>number.alive</code>	Column name of data giving the number of subjects alive. This column must be numeric. The default is 'Alive_at_Start'.
<code>number.loss</code>	Column name of data giving the number of subjects lost to followup. This column must be numeric. The default is 'Lost_to_Followup'.
<code>expected.rate</code>	Column name of data giving the interval survival. This column must be numeric. The default is 'Expected_Survival_Interval'.
<code>model.form</code>	an object of class "formula": a symbolic description of covariates. Example: <code>~1+age+as.factor(stage)</code>
<code>maxnum.jp</code>	The maximum number of join points allowed. The default is zero, which is equivalent to a proportional hazard relative survival model.
<code>proj.year.num</code>	The number of projection years for use in the prediction step. Default value is 5 years, with a valid range of 0 to 30 years.
<code>op</code>	List of more options. Details —

- **numbetwn:** integer value, number of skipped obs between joinpoints exclusive (not count for the joinpoints). Default is 2.
 - **numfromstart:** integer value, number of skipped obs from the first obs to joinpoints exclusive (not count for the joinpoint). Default is 3.
 - **numtoend:** integer value, number of skipped obs from the first obs to joinpoints exclusive (not count for the joinpoint). Default is 4.
- delLastIntvl an logical value indicating whether or not deleting records of last intervals of all years. The default is FALSE.
- add.data.cols Character vector of column names in data to add onto the returned data frames of results. Use "_ALL_" to add all columns and use NULL to not add any columns. The default is "_ALL_".

Details

The data to be included in the analysis must contain unique year, interval pairs. If not, then an error will be thrown. The subset option can be used to ensure that there are unique pairs of year and interval.

This function will set up the data based on the value of start by removing all rows with interval less than or equal to start and then call the [joinpoint](#) function.

Value

An object of class "joinpoint" will be returned with attributes:

- | | |
|---------------|--|
| coefficients | a named vector of coefficients and standard errors |
| jp | the estimates of the join points |
| converged | convergence status |
| predicted | the fitted relative survival rates |
| fullpredicted | the full output matrix, with all year/interval combinations and projections |
| xbeta | the linear predictor |
| ll | log likelihood |
| aic | AIC |
| bic | BIC |
| FitList | a list that contains fitting results for the number of joinpoints = 0,1,...,numJPoints respectively. |

References

Yu, B., Huang, L., Tiwari, R. C., Feuer, E. J. and Johnson, K. A. (2009), Modeling population-based cancer survival trends by using join point models for grouped survival data. *Journal of the Royal Statistical Society: Series A*, 172, 405-425.

See Also

[joinpoint](#), [joinpoint.conditional](#)

Examples

```
#Load the provided SEER 18 breast cancer example data.
data("breast.example", package="JPSurv")

# Subset of observations to use
subset <- "Age_groups == '00-49' & Breast_stage == 'Localized'"

# Fit the conditional survival join point model with starting
# interval 5
fit <- joinpoint.cond(breast.example, subset, 5,
                     year="Year_of_diagnosis_1975",
                     model.form=NULL, maxnum.jp=0)
```

joinpoint.conditional *Fitting a join point conditional relative survival model from the unconditional model*

Description

Fitting a joinpoint conditional relative survival model from the unconditional model

Usage

```
joinpoint.conditional(fit.uncond, start.intervals, end.intervals, njp=NULL)
```

Arguments

fit.uncond	Object returned from joinpoint .
start.intervals	Vector of integers giving the intervals to condition on.
end.intervals	Vector of integers giving the end number of intervals. This vector must have the same length and order as start.intervals with end.intervals[i] > start.intervals[i].
njp	NULL or the number of joinpoints corresponding to one of the fitted models in fit.uncond. If njp = k, then the conditional probabilities will be based on the model with k joinpoints. If NULL, then the model corresponding to the best fit returned by joinpoint will be used.

Details

This function computes the conditional survival

$$P(T > t_{j+k} | T > t_j) = \frac{P(T > t_{j+k})}{P(T > t_j)}, \quad k = 1, \dots, m$$

Value

A data frame similar to the fullpredicted data frame returned from [joinpoint](#) except that it will only contain rows corresponding to the start.intervals and end.intervals that were specified. The data frame will also contain the additional column "Start.interval", and will be grouped by the start.intervals.

References

Yu, B., Huang, L., Tiwari, R. C., Feuer, E. J. and Johnson, K. A. (2009), Modeling population-based cancer survival trends by using join point models for grouped survival data. *Journal of the Royal Statistical Society: Series A*, 172, 405-425.

See Also

[joinpoint](#), [joinpoint.seerdata](#)

Examples

```
#Load the provided SEER 18 breast cancer example data.
data("breast.example", package="JPSurv")

# Subset of observations to use
subset <- "Age_groups == '00-49' & Breast_stage == 'Localized'"

# Fit the unconditional survival join point model
fit <- joinpoint(breast.example, subset,
                year="Year_of_diagnosis_1975",
                model.form=NULL, maxnum.jp=0)

# Compute conditional survival  $S(10 | 5) = P(T>10 | T>5)$ 
ret <- joinpoint.conditional(fit, 5, 10)
```

joinpoint.relaxProp *Relaxing the proportionality assumption*

Description

Fitting a joinpoint survival model by relaxing the proportionality assumption

Usage

```
joinpoint.relaxProp(data, subset, max.cutpoint=5,
                   year="Year", interval="Interval", number.event="Died",
                   number.alive="Alive_at_Start", number.loss="Lost_to_Followup",
                   expected.rate="Expected_Survival_Interval",
                   observed.rate="Observed_Survival_Interval",
                   model.form=NULL, maxnum.jp=0,
                   proj.year.num=5, op=list(), delLastIntvl=FALSE, add.data.cols=NULL)
```

Arguments

data Data frame containing all variables in the model.

subset A logical vector of length `nrow(data)`, a character string or `NULL` to include a particular subset of data in the analysis. See **Details** and **Examples**.

<code>max.cutpoint</code>	A positive integer or NULL giving the number of cutpoints to consider. If NULL, then it will be set to the number of intervals minus one. The default is 5.
<code>year</code>	Column name of data giving the year or year code. This column must be numeric. The default is 'Year'.
<code>interval</code>	Column name of data giving the time interval elapsed from the starting time to the event time. This column must be numeric. The default is 'Interval'.
<code>number.event</code>	Column name of data giving the number of events or deaths. This column must be numeric. The default is 'Died'.
<code>number.alive</code>	Column name of data giving the number of subjects alive. This column must be numeric. The default is 'Alive_at_Start'.
<code>number.loss</code>	Column name of data giving the number of subjects lost to followup. This column must be numeric. The default is 'Lost_to_Followup'.
<code>expected.rate</code>	Column name of data giving the expected interval survival. This column must be numeric. The default is 'Expected_Survival_Interval'.
<code>observed.rate</code>	Column name of data giving the observed interval survival. This column must be numeric. The default is 'Observed_Survival_Interval'.
<code>model.form</code>	an object of class "formula": a symbolic description of covariates. Example: <code>~1+age+as.factor(stage)</code>
<code>maxnum.jp</code>	The maximum number of join points allowed. The default is zero, which is equivalent to a proportional hazard relative survival model.
<code>proj.year.num</code>	The number of projection years for use in the prediction step. Default value is 5 years, with a valid range of 0 to 30 years.
<code>op</code>	List of more options. Details — <ul style="list-style-type: none"> • numbetwn: integer value, number of skipped obs between joinpoints exclusive (not count for the joinpoints). Default is 2. • numfromstart: integer value, number of skipped obs from the first obs to joinpoints exclusive (not count for the joinpoint). Default is 3. • numtoend: integer value, number of skipped obs from the first obs to joinpoints exclusive (not count for the joinpoint). Default is 4.
<code>delLastIntvl</code>	an logical value indicating whether or not deleting records of last intervals of all years. The default is FALSE.
<code>add.data.cols</code>	Character vector of column names in data to add onto the returned data frames of results. Use "_ALL_" to add all columns and use NULL to not add any columns. The default is NULL.

Details

This function finds the optimal clustering of intervals (1, ..., I), where I is the number of intervals, such that there are at most two ordered clusters of the form (1, ..., j) and (j+1, ..., I). For each ordered cluster, a model is fit and the BIC is computed. The algorithm is as follows:

1. Fit the (unconditional) joinpoint survival model on intervals (1, ..., I) and compute the BIC and call it BIC-0.
2. For each cutpoint j , $j = 1, \dots, \text{max.cutpoint}$, fit the (unconditional) joinpoint survival model on

joinpoint.seerdata *Read in and format SEER*Stat data*

Description

A function that reads in and format SEER*Stat data in a single step. Either numeric values or value labels can be imported.

Usage

```
joinpoint.seerdata(seerfilename,newvarnames,UseVarLabelsInData=FALSE)
```

Arguments

seerfilename The file name of the seer files. Here one needs two files: the seer directory file with name extention 'dic' and the seer data file with the same name of the directory file but different name extention, i.e. 'txt'.

newvarnames A list of key words used for referencing the variable in the seer directory file and data file.

UseVarLabelsInData A logic value or character variables giving variable names. If true or the variable names, then variable labels read from the dic file will replace associated numeric values in the data.frame object, which stores data from the associated data file, for all vars in dic or those (names of which are given by UseVarLabelsInData). If false, then data read from the associated data file won't be changed.

Value

A data frame containing the SEER*Stat data read in.

Examples

```
# For this example we will be referencing the SEER*Stat session that was used to
# create the breast.example data included with the package.
# If the "breast.example.txt" is the output data file from SEER*Stat, then joinpoint.seerdata
# can be used to input the data quickly, while taking into account the relevant information
# in the accompanying dictionary file.
# Input data is stored in breast.example
# breast.data = joinpoint.seerdata(seerfilename="breast.example",
#                               newvarnames=c("Age_groups", "Breast_Stage", "Year_of_diagnosis_1975"),
#                               UseVarLabelsInData=FALSE)
# breast.data = joinpoint.seerdata(seerfilename="breast.example",
#                               newvarnames=c("Breast_Stage", "Year_of_diagnosis_1975"),
#                               UseVarLabelsInData=c("Breast_Stage", "Year_of_diagnosis_1975"))
# The breast.data object can immediately be used as input to the joinpoint modeling function.
```

jp.plot.cond.year *Plot of conditional survival versus year at diagnosis*

Description

Plot of conditional survival versus year at diagnosis

Usage

```
jp.plot.cond.year(fit.cond, start.interval=NULL, end.interval=NULL,  
                 year.col="Year", interval.col="Interval",  
                 relSurvInt.col="Relative_Survival_Interval", addToYear=0,  
                 ylim=NULL)
```

Arguments

fit.cond	Object returned from joinpoint.conditional .
start.interval	NULL or the starting interval. If NULL, then the smallest starting interval will be chosen.
end.interval	NULL or the ending interval. If NULL, then the largest interval will be chosen.
year.col	The name of the year at diagnosis column in fit.cond. The default is "Year".
interval.col	The name of the interval column in fit.cond. The default is "Interval".
relSurvInt.col	The name of the relative survival interval column in fit.cond. The default is "Relative_Survival_Interval".
addToYear	Integer to add to the year at diagnosis column to give the correct years at diagnosis for displaying. The default is 0.
ylim	NULL or the y-axis limits of the plot. If NULL, then the limits will be determined from the data.

Details

The lines on the plot use the pred_cum column in fit.cond, while the points are from the relSurvInt.col column.

Value

NULL

See Also

[joinpoint](#), [joinpoint.conditional](#)

Examples

```
#Load the provided SEER 18 breast cancer example data.
data("breast.example", package="JPSurv")

# Subset of observations to use
subset <- "Age_groups == '00-49' & Breast_stage == 'Localized'"

# Fit the unconditional survival join point model
fit <- joinpoint(breast.example, subset,
                year="Year_of_diagnosis_1975",
                model.form=NULL, maxnum.jp=0)

# Compute conditional survival  $S(10 | 5) = P(T>10 | T>5)$ 
ret <- joinpoint.conditional(fit, 5, 10)

jp.plot.cond.year(ret, year.col="Year_of_diagnosis_1975")
```

jp.plot.death

Plot of the annual probability of dying

Description

Plot of the annual probability of dying

Usage

```
jp.plot.death(fit.uncond, fit.cond, start.interval=NULL, end.interval=NULL,
              year.col="Year", interval.col="Interval",
              relSurvInt.col="Relative_Survival_Interval", addToYear=0,
              ylim=NULL)
```

Arguments

fit.uncond	Object returned from joinpoint .
fit.cond	Object returned from joinpoint.conditional .
start.interval	NULL or the starting interval. If NULL, then the smallest starting interval will be chosen.
end.interval	NULL or the ending interval. If NULL, then the largest interval will be chosen.
year.col	The name of the year at diagnosis column in fit.cond. The default is "Year".
interval.col	The name of the interval column in fit.cond. The default is "Interval".
relSurvInt.col	The name of the relative survival interval column in fit.cond. The default is "Relative_Survival_Interval".
addToYear	Integer to add to the year at diagnosis column to give the correct years at diagnosis for displaying. The default is 0.
ylim	NULL or the y-axis limits of the plot. If NULL, then the limits will be determined from the data.

Details

The lines on the plot use the `pred_int` column in `fit.cond`, while the points are from the `relSurvInt.col` column.

Value

NULL

See Also

[joinpoint](#), [joinpoint.conditional](#)

Examples

```
#Load the provided SEER 18 breast cancer example data.
data("breast.example", package="JPSurv")

# Subset of observations to use
subset <- "Age_groups == '00-49' & Breast_stage == 'Localized'"

# Fit the unconditional survival join point model
fit <- joinpoint(breast.example, subset,
                year="Year_of_diagnosis_1975",
                model.form=NULL, maxnum.jp=0)

# Compute conditional survival  $S(10 | 5) = P(T>10 | T>5)$ 
ret <- joinpoint.conditional(fit, 5, 10)

jp.plot.death(fit, ret, year.col="Year_of_diagnosis_1975")
```

jp.plot.surv

Plot of conditional survival versus interval

Description

Plot of conditional survival versus interval

Usage

```
jp.plot.surv(fit.uncond, fit.cond, start.interval=NULL, end.interval=NULL,
            year.col="Year", interval.col="Interval",
            relSurvInt.col="Relative_Survival_Interval", addToYear=0,
            ylim=NULL, yearsToPlot=NULL, legend.pos="bottom")
```

Arguments

<code>fit.uncond</code>	Object returned from joinpoint .
<code>fit.cond</code>	Object returned from joinpoint.conditional .
<code>start.interval</code>	NULL or the starting interval. If NULL, then the smallest starting interval will be chosen.
<code>end.interval</code>	NULL or the ending interval. If NULL, then the largest interval will be chosen.
<code>year.col</code>	The name of the year at diagnosis column in <code>fit.cond</code> . The default is "Year".
<code>interval.col</code>	The name of the interval column in <code>fit.cond</code> . The default is "Interval".
<code>relSurvInt.col</code>	The name of the relative survival interval column in <code>fit.cond</code> . The default is "Relative_Survival_Interval".
<code>addToYear</code>	Integer to add to the year at diagnosis column to give the correct years at diagnosis for displaying. The default is 0.
<code>ylim</code>	NULL or the y-axis limits of the plot. If NULL, then the limits will be determined from the data.
<code>yearsToPlot</code>	NULL or the years at diagnosis to display in the plot. If NULL, then no more than five years will be selected for the plot.
<code>legend.pos</code>	Character string to give the legend position in the plot. The default is "bottom".

Details

The lines on the plot use the `pred_cum` column in `fit.cond`, while the points are from the `relSurvInt.col` column.

Value

NULL

See Also

[joinpoint](#), [joinpoint.conditional](#)

Examples

```
#Load the provided SEER 18 breast cancer example data.
data("breast.example", package="JPSurv")

# Subset of observations to use
subset <- "Age_groups == '00-49' & Breast_stage == 'Localized'"

# Fit the unconditional survival join point model
fit <- joinpoint(breast.example, subset,
                year="Year_of_diagnosis_1975",
                model.form=NULL, maxnum.jp=0)

# Compute conditional survival  $S(10 | 5) = P(T>10 | T>5)$ 
ret <- joinpoint.conditional(fit, 5, 10)

jp.plot.surv(fit, ret, year.col="Year_of_diagnosis_1975")
```

 Plot.dying.year.annotate

Figure - Percent Change in the Annual Probability of Dying of Cancer by Diagnosis Year

Description

A function that returns a plot (or a list including the Relative Change in Hazard trend measures and the plot) for percent change in the annual probability of dying of cancer by diagnosis year using ggplot. The annotation feature is available for nJP<=3 and the number of multiple intervals selected <=3.

Usage

```
Plot.dying.year.annotate(plotdata, fit, nJP, yearvar,
                        obsintvar = "Relative_Survival_Interval",
                        predintvar = "Predicted_ProbDeath_Int",
                        interval = "Interval",
                        annotation = 0, topanno = 1, trend = 0,
                        title = NULL)
```

Arguments

plotdata	The graph data returned by function download.data with downloadtype="graph".
fit	The joinpoint object containing the model output.
nJP	The number of joinpoints in the model.
yearvar	The variable name for year of diagnosis used in argument 'year' of the function joinpoint.
obsintvar	The variable name for observed interval survival. The default is "Relative_Survival_Interval" for relative survival data. For cause-specific data, it needs to be changed accordingly.
predintvar	The variable name for predicted interval survival. The default is "Predicted_ProbDeath_Int".
interval	The variable name for year since diagnosis. The default is 'Interval'.
annotation	The indicator for the annotation feature. The default is 0 (no annotation on the plot). Two plots with and without annotations will be returned in a list when annotation=1.
topanno	The indicator for showing the top curve annotation. The default is 1 (annotation for the top curve).
trend	The indicator for returning the Relative Change in Hazard trend measures . The default is 0 (no trend tables returned).
title	The title used for the plot. If title=NULL, the default title "Annual Probability of Dying of Cancer by Diagnosis Year" will be used.

Value

The returned object is a list and depends on the values for trend and annotation. The list will contain some or all of the names plot_anno, plot_no_anno, and trends, where plot_anno is an object of class ggplot for a plot with annotations, plot_no_anno is an object of class ggplot for a plot without annotations, and trends is a list of data frames for the change in hazard trend measures with columns start.year, end.year, estimate, std.error, lowCI, upCI, and interval.

Author(s)

Fanni Zhang <zhangf@imsweb.com>

See Also

[download.data](#), [Plot.surv.year.annotate](#), [Plot.surv.int.mutiyears](#), [aapc](#).

Examples

```
data("breast.example", package="JPSurv")
data("fit1", package="JPSurv")

yearvar<-"Year_of_diagnosis_1975"
obsintvar<-"Relative_Survival_Interval"
predintvar<-"Predicted_ProbDeath_Int"
interval<-"Interval"
nJP<-3
data.graph<-download.data(breast.example, fit1, nJP, yearvar, "graph",
                          subsetStr1, interval, int.select=c(1,3,5))
out.anno<-Plot.dying.year.annotate(data.graph, fit1, nJP, yearvar, obsintvar,
                                   predintvar, interval, annotation=1, topanno=1, trend=1)
out<-Plot.dying.year.annotate(data.graph, fit1, nJP, yearvar, obsintvar,
                              predintvar, interval, annotation=0, topanno=0, trend=1)
trend.rch<-out.anno$trends
plot.rch.anno<-out.anno$plot_anno
plot.rch<-out$plot_no_anno
```

Plot.surv.int.mutiyears

Figure - Cumulative Survival by Interval

Description

A function that returns a plot for Cumulative Survival by Interval supporting multiple selected years.

Usage

```
Plot.surv.int.mutiyears(plotdata, fit, nJP, yearvar,
                        obscumvar = "Relative_Survival_Cum",
                        predcumvar = "Predicted_Survival_Cum",
                        interval = "Interval", year.select = NULL)
```

Arguments

<code>plotdata</code>	Either of the following data set would work. 1) the graph data returned by function <code>download.data</code> with <code>downloadtype="graph"</code> and all intervals needed for <code>int.col</code> . 2) the full data returned by function <code>download.data</code> with <code>downloadtype="full"</code> .
<code>fit</code>	The joinpoint object containing the model output.
<code>nJP</code>	The number of joinpoints in the model.
<code>yearvar</code>	The variable name for year of diagnosis used in argument 'year' of the function <code>joinpoint</code> .
<code>obscumvar</code>	The variable name for observed relative cumulative survival. The default is "Relative_Survival_Cum" for relative survival data. For cause-specific data, it needs to be changed accordingly.
<code>predcumvar</code>	The variable name for predicted cumulative survival. The default is "Predicted_Survival_Cum".
<code>interval</code>	The variable name for year since diagnosis. The default is 'Interval'.
<code>year.select</code>	The year values selected for the plot. The default is NULL.

Value

An object of class `ggplot` containing the plot.

Author(s)

Fanni Zhang <zhangf@imsweb.com>

See Also

[download.data](#), [Plot.dying.year.annotate](#), [Plot.surv.year.annotate](#).

Examples

```
data("breast.example", package="JPSurv")
data("fit3", package="JPSurv")

yearvar<-"Year_of_diagnosis_1975"
obscumvar<-"Relative_Survival_Cum"
predcumvar<-"Predicted_Survival_Cum"
interval<-"Interval"
nJP<-2
data.full<-download.data(breast.example,fit3,nJP,yearvar,"full",
                        subsetStr3,interval="Interval")
plot<-Plot.surv.int.multiyears(data.full,fit3,nJP,yearvar,obscumvar,
                              predcumvar,interval,year.select=c(1985,1990))
```

 Plot.surv.year.annotate

Figure - Average Change in Cumulative Survival by Diagnosis Year

Description

A function that returns a plot (or a list including the Absolute Change in Survival trend measures and the plot) for average change in cumulative survival by diagnosis year using ggplot. The annotation feature is available for nJP<=3 and the number of multiple intervals selected <=3.

Usage

```
Plot.surv.year.annotate(plotdata, fit, nJP, yearvar,
                        obscumvar = "Relative_Survival_Cum",
                        predcumvar = "Predicted_Survival_Cum",
                        interval = "Interval", annotation = 0,
                        trend = 0, title = NULL)
```

Arguments

plotdata	The graph data returned by function download.data with downloadtype="graph".
fit	The joinpoint object containing the model output.
nJP	The number of joinpoints in the model.
yearvar	The variable name for year of diagnosis used in argument 'year' of the function joinpoint.
obscumvar	The variable name for observed relative cumulative survival. The default is "Relative_Survival_Cum" for relative survival data. For cause-specific data, it needs to be changed accordingly.
predcumvar	The variable name for predicted cumulative survival. The default is "Predicted_Survival_Cum".
interval	The variable name for year since diagnosis. The default is 'Interval'.
annotation	The indicator for the annotation feature. The default is 0 (no annotation on the plot). Two plots with and without annotations will be returned in a list when annotation=1.
trend	The indicator for returning the Absolute Change in Survival trend measure tables. The default is 0 (no trend tables returned).
title	The title used for the plot. If title=NULL, the default title "Relative Survival by Diagonosis Year" or "Cause-Specific Survival by Diagnosis Year" will be used according to the input data type.

Value

The returned object is a list and depends on the values for trend and annotation. The list will contain some or all of the names plot_anno, plot_no_anno, and trends, where plot_anno is an object of class ggplot for a plot with annotations, plot_no_anno is an object of class ggplot for a plot without annotations, and trends is a list of data frames for the change in hazard trend measures with columns start.year, end.year, estimate, std.error, lowCI, upCI, and interval.

Author(s)

Fanni Zhang <zhangf@imsweb.com>

See Also

[download.data](#), [Plot.dying.year.annotate](#), [Plot.surv.int.multiyears](#), [aapc.multiints](#).

Examples

```
data("breast.example", package="JPSurv")
data("fit3", package="JPSurv")

yearvar<-"Year_of_diagnosis_1975"
obscumvar<-"Relative_Survival_Cum"
predcumvar<-"Predicted_Survival_Cum"
interval<-"Interval"
nJP<-2
data.graph<-download.data(breast.example,fit3,nJP,yearvar,"graph",
                          subsetStr3,interval,int.select=c(1,5))

out<-Plot.surv.year.annotate(data.graph,fit3,nJP,yearvar,obscumvar,
                             predcumvar,interval,annotation=0,trend=0)
```

Index

* data

- breast.example, 5
- fit1, 7
- fit2, 7
- fit3, 7

aapc, 3, 5, 24

aapc.multiints, 4, 27

breast.example, 5

dictionary.overview, 5

download.data, 6, 24, 25, 27

fit1, 7

fit2, 7

fit3, 7

input.valid, 8

joinpoint, 8, 13–15, 17, 19–22

joinpoint.choose.cutpoint, 11

joinpoint.cond, 12, 17

joinpoint.conditional, 13, 14, 19–22

joinpoint.relaxProp, 11, 15

joinpoint.seerdata, 15, 18

jp.plot.cond.year, 19

jp.plot.death, 20

jp.plot.surv, 21

JPSurv (JPSurv-package), 2

JPSurv-package, 2

Plot.dying.year.annotate, 7, 23, 25, 27

Plot.surv.int.multiyears, 7, 24, 24, 27

Plot.surv.year.annotate, 7, 24, 25, 26

subsetStr1 (fit1), 7

subsetStr2 (fit2), 7

subsetStr3 (fit3), 7