

# Package ‘survout’

October 14, 2022

**Type** Package

**Title** Excel Conversion of R Survival Analysis Output

**Version** 0.1.0

**Author** Xuefei Jia

**Maintainer** Xuefei Jia <xuefeij.ai@gmail.com>

**Description** Simple and quick method of exporting the most often used survival analysis results to an Excel sheet.

**License** GPL-3

**Imports** cmprsk, dplyr, openxlsx, stats, survival, tibble

**Suggests** covr, MASS, reshape2, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-09-26 12:50:02 UTC

## R topics documented:

crisk_cat . . . . .	2
crisk_con . . . . .	3
crisk_multi . . . . .	3
crisk_multiuni . . . . .	4
crisk_ord . . . . .	5
factor2ind . . . . .	6
p2excel . . . . .	7
p2excel_pre . . . . .	8
surv_multi . . . . .	8
surv_multiuni . . . . .	9
surv_uni_cat . . . . .	10
surv_uni_con . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

crisk_cat	<i>Modify the Output for Uni-variable and Multi-variable Competing Risk Analysis (Categorical Only)</i>
-----------	---

---

### Description

This function generates a table of competing risk analysis result with number of patients, number of event, number of competing event,

### Usage

```
crisk_cat(  
  csurv,  
  cevent,  
  cvars,  
  gnames,  
  month = 0,  
  y1 = TRUE,  
  y2 = TRUE,  
  y5 = TRUE  
)
```

### Arguments

csurv	the duration of follow-up time in months.
cevent	the status indicator, which is generally 0 = alive, 1 = event, 2 = other event
cvars	a vector, which has the variable's values (categorical only)
gnames	a text string, which is the name of the variable.
month	a number to get the month-rate of competing risk.
y1	logical value indicating whether the 1-year competing risk rate should be reported.
y2	logical value indicating whether the 2-year competing risk rate should be reported.
y5	logical value indicating whether the 5-year competing risk rate should be reported.

### Value

a tibble of competing risk analysis output.

### Examples

```
Dat <- MASS::Melanoma  
Dat$time <- Dat$time/30.5  
output <- risk_cat(Dat$time, Dat$status, Dat$ulcer, "ulcer")
```

---

crisk_con	<i>Modify the Output for Uni-variable and Multi-variable Competing Risk Analysis (Continuous and Ordinal Only)</i>
-----------	--

---

**Description**

This function generates a table of competing risk analysis result with HR (95% Confidence Interval), P value.

**Usage**

```
crisk_con(csurv, cevent, cvars, gnames)
```

**Arguments**

csurv	the duration of follow-up time in months.
cevent	the status indicator, which is generally 0 = alive, 1 = event, 2 = other event
cvars	a matrix, which has the variables' values (continuous and ordinal only)
gnames	a text vector, which are the names of the variables.

**Value**

a dataframe containing HRs (with 95% Confidence Intervals) and P values

**Examples**

```
Dat <- MASS::Melanoma
Dat$time <- Dat$time/30.5
X <- cbind(Dat$age, Dat$thickness)
Gnames <- c('age', 'thickness')
output <- risk_con(Dat$time, Dat$status, X, Gnames)
```

---

crisk_multi	<i>Modify the Output for a Multi-variable Competing Risk Analysis .</i>
-------------	---

---

**Description**

Create a table with the general multi-variable competing risk analysis results, including the HR (95 percent CI), P value.

**Usage**

```
crisk_multi(dat, csurv, cevent, convars = NULL, catvars = NULL)
```

**Arguments**

dat	a data.frame in which to interpret the variables.
csurv	this is the follow up time.
cevent	the status indicator, normally 0=alive, 1=dead.
convars	a vector of con variable names.
catvars	a vector of cat variable names.

**Value**

a tibble of competing risk analysis output.

**Examples**

```
Dat <- MASS::Melanoma
Dat$time <- Dat$time/30.5
con_var <- c("age", "thickness")
cat_var <- c("sex", "ulcer")
multi_out <- crisk_multi(Dat, "time", "status", catvars = cat_var, convars = con_var)
```

---

crisk_multiuni	<i>Modify the Output for Multiple Uni-variable Competing Risk Analysis</i>
----------------	--

---

**Description**

This function generates a table of competing risk analysis result with number of patients, number of event, number of competing event,

**Usage**

```
crisk_multiuni(
  dat,
  csurv,
  cevent,
  catvars = NULL,
  convars = NULL,
  ordvars = NULL,
  y1 = TRUE,
  y2 = TRUE,
  y5 = TRUE,
  month = 0
)
```

**Arguments**

dat	a data.frame in which to interpret the variables.
csurv	this is the follow up time.
cevent	the status indicator, normally 0=alive, 1=dead.
catvars	a vector of cat variable names.
convars	a vector of con variable names.
ordvars	a vector of ordinal variable names.
y1	logical value indicating whether the 1-year competing risk rate should be reported.
y2	logical value indicating whether the 2-year competing risk rate should be reported.
y5	logical value indicating whether the 5-year competing risk rate should be reported.
month	a number to get the month-rate of competing risk.

**Value**

a tibble of competing risk analysis output.

**Examples**

```
Dat <- MASS::Melanoma
Dat$time <- Dat$time/30.5
Dat$ulcer <- as.factor(Dat$ulcer)
con_var <- c("age")
ord_var <- c("ulcer")
cat_var <- c("sex")
uni_out <- crisk_multiuni(Dat, "time", "status", cat_var, con_var, ord_var)
```

---

crisk_ord	<i>Modify the Output for Uni-variable and Multi-variable Competing Risk Analysis (Ordinal Only)</i>
-----------	---

---

**Description**

This function generates a table of competing risk analysis result with number of patients, number of event, number of competing event,

**Usage**

```
crisk_ord(
  csurv,
  cevent,
  cvars,
  gnames,
  month = 0,
  y1 = TRUE,
  y2 = TRUE,
  y5 = TRUE
)
```

**Arguments**

csurv	the duration of follow-up time in months.
cevent	the status indicator, which is generally 0 = alive, 1 = event, 2 = other event
cvars	a vector, which has the variable's values (ordinal only)
gnames	a text string, which is the name of the variable.
month	a number to get the month-rate of competing risk.
y1	logical value indicating whether the 1-year competing risk rate should be reported.
y2	logical value indicating whether the 2-year competing risk rate should be reported.
y5	logical value indicating whether the 5-year competing risk rate should be reported.

**Value**

a tibble of competing risk analysis output.

**Examples**

```
Dat <- MASS::Melanoma
Dat$time <- Dat$time/30.5
output <- risk_ord(Dat$time, Dat$status, as.factor(Dat$year), "year")
```

---

factor2ind

*Make An Integer Matrix Out of A Factor Variable.*

---

**Description**

Create an indicator matrix of dimension  $\text{length}(x) \times (\text{nlevels}(x)-1)$  with the column corresponding to the baseline level removed (by default the first level is used as baseline).

**Usage**

```
factor2ind(x, baseline)
```

**Arguments**

x                    a variable.  
baseline            a string indicating the reference level.

**Value**

a matrix

**Examples**

```
x = gl(4, 2, labels = c("A", "B", "C", "D"))  
factor2ind(x)  
factor2ind(x, "C")
```

---

p2excel

*Export the A Single Dataframe to An Excel Sheet*

---

**Description**

The function saves a dataframe into an excel sheet with a predetermined format.

**Usage**

```
p2excel(  
  tabname = "Default",  
  datastable,  
  tablename = "Default",  
  filename = "Default.xlsx"  
)
```

**Arguments**

tabname            a string with the tab's name.  
datastable        the dataframe that will be exported to Excel.  
tablename        a string containing the table label and title, which will appear as the first row  
filename        the name of the spreadsheet

**Value**

a spreadsheet containing an exported tables

---

p2excel\_pre

*Prepare to Export the Dataframe to An Excel Sheet.*

---

### Description

The function saves the dataframe as a tab and prepares it for output into an excel sheet with a predetermined format.

### Usage

```
p2excel_pre(tabname = "Default", datastable, tablename = "Default", filename)
```

### Arguments

tabname	a string with the tab's name.
datastable	the dataframe that will be exported to Excel.
tablename	a string containing the table label and title, which will appear as the first row
filename	the name of the spreadsheet

### Value

a spreadsheet containing all of the exported tables

### Examples

```
Dat <- survival::lung
results <- surv_uni_cat(Dat, "time", "status", "sex", report_index = TRUE)
wb <- openxlsx::createWorkbook()
wb <- p2excel_pre("survival_results",results,"Table 1. Overall Survival anlysis",wb)
## Not run:
## saveWorkbook(wb, file = "os.xlsx", overwrite = TRUE)
## End(Not run)
```

---

surv\_multi

*Modify the Output for a Multi-variable Survival Analysis.*

---

### Description

Create a table with the general multi-variable survival analysis results, including the HR (95 percent CI), P value.

### Usage

```
surv_multi(...)
```



**Arguments**

... arguments will be passed to coxph

**Value**

a dataframe containing coxph output that includes variable names, HRs (95

**Examples**

```
Dat <- survival::lung
surv_multi(survival::Surv(time, status) ~ as.factor(sex) + age + meal.cal, data = Dat)
```

---

surv\_multiuni

*Modify the Output for Multiple Uni-variable Survival Analysis*


---

**Description**

This function generates a table with the general survival analysis results, including the number of total patients, the number of events, the estimated median, the 1,2,5 year rate, the HR (95 percent confidence interval), the P value, the AIC, and the C index. This function just modifies the output table's format.

**Usage**

```
surv_multiuni(
  dat,
  stime,
  sevent,
  catvars = NULL,
  convars = NULL,
  y1 = TRUE,
  y2 = TRUE,
  y5 = TRUE,
  medianCI = FALSE,
  report_index = FALSE
)
```

**Arguments**

dat	a dat.frame.
stime	the duration of follow-up time in months.
sevent	the status indicator, which is generally 0 = alive, 1 = dead.
catvars	a vector of categorical variable names.
convars	a vector of continuous variables names.
y1	logical value indicating whether the 1-year survival rate should be reported.

y2	logical value indicating whether the 2-year survival rate should be reported.
y5	logical value indicating whether the 5-year survival rate should be reported.
medianCI	logical value indicating whether the 95 percent confidence interval of projected median survival should be reported.
report_index	logical value indicating if to report the show AIC and C index.

## Value

A tibble of survival output

## Examples

```
Dat <- survival::lung
convars <- c("age", "meal.cal")
catvars <- c("sex")
surv_multiuni(Dat, "time", "status", catvars, convars, medianCI = TRUE)
```

---

surv\_uni\_cat

*Modify the Survival Output for a Categorical Variable.*

---

## Description

This function generates a table with the general survival analysis results, including the number of total patients, the number of events, the estimated median, the 1,2,5 year rate, the HR (95 percent confidence interval), the P value, the AIC, and the C index. This function just modifies the output table's format.

## Usage

```
surv_uni_cat(
  dat,
  stime,
  sevent,
  svar,
  month = 0,
  medianCI = TRUE,
  y1 = TRUE,
  y2 = TRUE,
  y5 = TRUE,
  report_index = FALSE
)
```

**Arguments**

dat	a data.frame.
stime	the duration of follow-up time in months.
sevent	the status indicator, which is generally 0 = alive, 1 = dead.
svar	a variable name.
month	a number to get the month-rate of survival.
medianCI	logical value indicating whether the 95 percent confidence interval of projected median survival should be reported.
y1	logical value indicating whether the 1-year survival rate should be reported.
y2	logical value indicating whether the 2-year survival rate should be reported.
y5	logical value indicating whether the 5-year survival rate should be reported.
report_index	logical value indicating if to report the show AIC and C index.

**Value**

a tibble of survival output

**Examples**

```
Dat <- survival::lung
surv_uni_cat(Dat, "time", "status", "sex", report_index = TRUE)
```

---

surv\_uni\_con

---

*Modify the Survival Output for a Continuous Variable.*


---

**Description**

This function generates a table with the general survival analysis results, including the number of total patients, the number of events, the P value, the AIC, and the C index. This function just modifies the output table's format.

**Usage**

```
surv_uni_con(dat, stime, sevent, svar, report_index = FALSE)
```

**Arguments**

dat	a data.frame.
stime	the duration of follow-up time in months.
sevent	the status indicator, which is generally 0 = alive, 1 = dead.
svar	a variable name.
report_index	logical value indicating if to report the show AIC and C index.

**Value**

a tibble of survival results.

**Examples**

```
Dat <- survival::lung  
surv_uni_con(Dat, "time", "status", "age", report_index = TRUE)
```

# Index

crisk\_cat, 2  
crisk\_con, 3  
crisk\_multi, 3  
crisk\_multiuni, 4  
crisk\_ord, 5  
  
factor2ind, 6  
  
p2excel, 7  
p2excel\_pre, 8  
  
surv\_multi, 8  
surv\_multiuni, 9  
surv\_uni\_cat, 10  
surv\_uni\_con, 11