

# Package ‘vsd’

October 12, 2022

**Title** Graphical Shim for Visual Survival Data Analysis

**Version** 0.1.0

**Description** Provides a shim command for survival analysis graphic generation.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Depends** survival, ggplot2

**Imports** dplyr, survminer, flexsurv, magrittr, ggpubr, muhaz

**Suggests** plotly

**RoxygenNote** 7.1.1

**Language** en-US

**Collate** 'options.R' 'plot\_forest.R' 'plot\_hazard.R'  
'plot\_parametric.R' 'util.R' 'vsd-package.R' 'vsd.R'

**NeedsCompilation** no

**Author** Daniel Camacho [aut, cre] (<<https://orcid.org/0000-0002-4980-9391>>),  
Ana Abreu [aut, ths] (<<https://orcid.org/0000-0002-6155-8492>>)

**Maintainer** Daniel Camacho <2016711@student.uma.pt>

**Repository** CRAN

**Date/Publication** 2021-05-11 09:40:02 UTC

## R topics documented:

vsd . . . . .	2
<b>Index</b>	<b>6</b>

**Description**

This function outputs renders of the inputted survival data analysis data and/or model, and their components, into a graphically pleasing output, under the [ggplot2](#) format.

**Usage**

```
vsd(  
  model,  
  data = NULL,  
  .interactive = FALSE,  
  .include = c("fit", "parametric", "forest", "residuals", "hazard"),  
  .arguments = list(),  
  ...  
)
```

```
## S3 method for class 'formula'
```

```
vsd(  
  model,  
  data = NULL,  
  .interactive = FALSE,  
  .include = c("fit", "hazard"),  
  .arguments = list(),  
  ...  
)
```

```
## S3 method for class 'Surv'
```

```
vsd(  
  model,  
  data = NULL,  
  .interactive = FALSE,  
  .include = c("fit", "hazard"),  
  .arguments = list(),  
  ...  
)
```

```
## S3 method for class 'coxph'
```

```
vsd(  
  model,  
  data = NULL,  
  .interactive = FALSE,  
  .include = c("fit", "forest", "residuals", "hazard"),  
  .arguments = list(),  
  ...  
)
```

```

)

## S3 method for class 'survfit'
vsd(
  model,
  data = NULL,
  .interactive = FALSE,
  .include = c("fit", "hazard"),
  .arguments = list(),
  ...
)

## S3 method for class 'survfitcox'
vsd(
  model,
  data = NULL,
  .interactive = FALSE,
  .include = c("fit", "forest", "residuals", "hazard"),
  .arguments = list(),
  ...
)

## S3 method for class 'flexsurvreg'
vsd(
  model,
  data = NULL,
  .interactive = FALSE,
  .include = c("fit", "parametric", "hazard"),
  .arguments = list(),
  ...
)

```

## Arguments

<code>model</code>	The survival model, or data structure, to generate graphics from
<code>data</code>	Dataframe from where the model fetches its variables, if left blank will be extracted from the model, if possible
<code>.interactive</code>	Allows to explore the generated graphs before returning (use with <a href="#">plotly</a> package for best results)
<code>.include</code>	Graph types to output if relevant, defaults to all possible
<code>.arguments</code>	Collection of list of arguments, indexed by the specific type of graph they should be passed to, has priority over ...
<code>...</code>	Miscellaneous arguments, passed to ALL graphs

## Details

Depending on the kind of model passed to the function, the kind of generated graphics might vary, but usually an estimation of the survival and risk curves (depending if the model has covariables)

is expected. The kinds of graphics that can be created according to a specific R object are detailed on Usage on `.include`'s definition, but non-relevant graphics can be requested without error, as the function ignores them silently.

Extra options for each graph kind can be passed to either all created graphics, by having them as generic arguments, or to specific graphic types, using a list in `.arguments`. Arguments are filtered, so that generic arguments aren't applied if a graphic kind wouldn't use them. As an example,

```
vsd(model, data,
     .arguments = list(fit = (size = 3, xlab = "Weeks"),
                      xlab = "Days"))
```

would set all graphics that have an label on the x axis to "Days", except the `fit` graph, which would have "Weeks" instead.

## Value

A list of ggplot2 graphs and/or list of graphs, relevant to the model

## Methods (by class)

- `formula`: Wraps `Surv(... ) ~ (... )` in a survfit object (Kaplan-Meier model)
- `Surv`: Wraps `Surv()` in a survfit object (Kaplan-Meier model)
- `coxph`: Wraps `coxph(... )` in a survfit object (Kaplan-Meier model)
- `survfit`: Graphical output for survfit objects (Kaplan-Meier model)
- `survfitcox`: Graphical output for survfit objects (Cox model)
- `flexsurvreg`: Graphical output for flexsurvreg objects (various parametric models)

## Generic graphical arguments

Unless specified, all graphics are created under `ggpubr::ggpar()` and have as additional options `palette`, `main`, `submain`, `xlab`, `ylab`, `legend.title` and `ggtheme`. Most line graphics also allow to set the options `size`, `linetype`, `alpha` and `color` to determine line styles, as detailed on `survminer::ggsurvplot()`.

### fit:

Line graphic, with a further subset of the options present in `survminer::ggsurvplot()`:  `censor`,  `censor.shape`,  `censor.size`,  `conf.int`,  `conf.int.style`.

### parametric:

Line graphic, with a further subset of the options present in `survminer::ggflexsurvplot()`:  `conf.int.km`.

### forest:

Non-standard graphic(s), using all options within `survminer::ggforest()`:  `main`,  `cpositions`,  `fontsize`,  `refLabel`,  `noDigits`.

**residuals:**

Line graphic(s), with a further subset of options present in `survminer::ggcoxzph()`: `resid`, `se`, `df`, `nsmo`, `var`, `caption`; and point style customization options as `point.col`, `point.size`, `point.shape`, and `point.alpha`.

**hazard:**

Line graphics, using the generic graphical arguments.

**Examples**

```
# non-models are coerced into a survfit object with default arguments
vsd(coxph(Surv(time, status) ~ sex + strata(rx) + adhere, data = colon),
     .include = c("haz"))

# parametric models are also supported with flexsurv
vsd(flexsurv::flexsurvreg(Surv(rectime, censrec) ~ group, data = flexsurv::bc, dist = 'gengamma'),
     .include = c("par"))
```

# Index

ggplot2, [2](#)

ggpubr::ggpar(), [4](#)

Surv(), [4](#)

survminer::ggcoxzph(), [5](#)

survminer::ggflexsurvplot(), [4](#)

survminer::ggforest(), [4](#)

survminer::ggsurvplot(), [4](#)

vsd, [2](#)