

# Package ‘landpred’

December 4, 2025

**Type** Package

**Title** Landmark Prediction of a Survival Outcome

**Version** 2.0

**Description** Nonparametric models for landmark prediction of long-term survival outcomes, incorporating covariate and short-term event information. The package supports the construction of flexible varying-coefficient models that use discrete covariates, as well as multiple continuous covariates. The goal is to improve prediction accuracy when censored short-term events are available as predictors, using robust nonparametric procedures that do not require correct model specification and avoid restrictive parametric assumptions found in alternative methods. More information on these models can be found in Parast et al. (2012, Journal of the American Statistical Association, <doi:10.1080/01621459.2012.721281>, and Parast et al. (2011, Biometrical Journal, <doi:10.1002/bimj.201000150>). More information and examples can also be found at <<https://baolong281.github.io/landpred/>>.

**License** GPL

**Imports** survival, stats, quantreg, splines, sm, quantreg

**NeedsCompilation** no

**Suggests** testthat (>= 3.0.0)

**Author** Dylan Huynh [aut],  
Layla Parast [aut, cre]

**Maintainer** Layla Parast <parast@austin.utexas.edu>

## Contents

AUC.landmark	2
BS.landmark	3
coef.landpred_model_continuous	4
coefficient_se	4
cumsum2	5
data_example_landpred	5
fit_glm_normal	6
fit_short_glm	6
get_model	7
Ghat.FUN	7
landpred	8
mse_cv	9
optimize_bandwidth	10
predict.landpred_model_continuous	11

predict.landpred_model_discrete . . . . .	11
print.landpred_model_continuous . . . . .	12
print.landpred_model_discrete . . . . .	12
print.landpred_object . . . . .	13
Prob.Covariate . . . . .	13
Prob.Covariate.ShortEvent . . . . .	14
Prob.Null . . . . .	14
summary.landpred_model_continuous . . . . .	15
summary.landpred_object . . . . .	15
var.fun . . . . .	16

<b>Index</b>	<b>17</b>
--------------	-----------

---

AUC.landmark	<i>Estimates the area under the ROC curve (AUC).</i>
--------------	--

---

## Description

This function calculates the AUC given the data (truth) and corresponding estimated probabilities; uses a continuity correction.

## Usage

```
AUC.landmark(t0, tau, data, weight=NULL)
```

## Arguments

t0	The landmark time.
tau	The prediction window.
data	A data matrix where the first column is $XL = \min(TL, C)$ where TL is the time of the long term event, C is the censoring time; the second column is $DL = 1*(TL < C)$ ; the third column is the estimated probability $P(TL < t0 + tau   TL > t0)$ .
weight	an optional weight to be incorporated in all estimation.

## Value

AUC.est	Estimated AUC
---------	---------------

## References

Parast, Layla, Su-Chun Cheng, and Tianxi Cai. Incorporating short-term outcome information to predict long-term survival with discrete markers. *Biometrical Journal* 53.2 (2011): 294-307.

## Examples

```
data(data_example_landpred)
t0=2
tau = 8

out = Prob.Null(t0=t0,tau=tau,data=data_example_landpred)

#get data with predictions
```

```
data_pred = out$data

#calculate AUC
AUC.landmark(t0=t0,tau=tau, data = data_pred[,c("XL","DL","prob_est")])
```

---

BS.landmark	<i>Estimates the Brier score.</i>
-------------	-----------------------------------

---

## Description

This function calculates the Brier score given the data (truth) and corresponding estimated probabilities.

## Usage

```
BS.landmark(t0, tau, data, weight=NULL)
```

## Arguments

t0	The landmark time.
tau	The prediction window.
data	A data matrix where the first column is $XL = \min(TL, C)$ where TL is the time of the long term event, C is the censoring time; the second column is $DL = 1*(TL < C)$ ; the third column is the estimated probability $P(TL < t_0 + \tau \mid TL > t_0)$ .
weight	an optional weight to be incorporated in all estimation.

## Value

Brier.score	Estimated Brier score
-------------	-----------------------

## References

Parast, Layla, Su-Chun Cheng, and Tianxi Cai. Incorporating short-term outcome information to predict long-term survival with discrete markers. *Biometrical Journal* 53.2 (2011): 294-307.

## Examples

```
data(data_example_landpred)
t0=2
tau = 8

out = Prob.Null(t0=t0,tau=tau,data=data_example_landpred)

#get data with predictions
data_pred = out$data

#calculate BS
BS.landmark(t0=t0,tau=tau, data = data_pred[,c("XL","DL","prob_est")])
```

---

```
coef.landpred_model_continuous
```

*Extract Coefficients from Landpred Continuous Model*

---

### Description

Extracts coefficients. If `t_s` is provided, it fits the short-term GLM and returns its coefficients.

### Usage

```
## S3 method for class 'landpred_model_continuous'
coef(object, t_s = NULL, ...)
```

### Arguments

<code>object</code>	A <code>landpred_model_continuous</code> object.
<code>t_s</code>	Optional short-term covariate time.
<code>...</code>	Additional arguments.

### Value

A named vector of coefficients.

---

```
coefficient_se
```

*Calculate Standard Errors for Coefficients*

---

### Description

Calculates standard errors for the coefficients of the landpred model. If `t_s` is provided, it uses the perturbation resampling method. Otherwise, it returns the standard errors from the GLM.

### Usage

```
coefficient_se(model, t_s = NULL, samples = 200)
```

### Arguments

<code>model</code>	A <code>landpred_model_continuous</code> object.
<code>t_s</code>	The time of the short-term covariate measurement.
<code>samples</code>	The number of resampling iterations.

### Value

A named vector of standard errors.

cumsum2

*Cumulative Sum Helper***Description**

Cumulative Sum Helper

**Usage**

cumsum2(mydat)

**Arguments**

mydat                  Data matrix.

data\_example\_landpred *Hypothetical data to be used in examples.***Description**

Hypothetical data to be used in examples.

**Usage**

data(data\_example\_landpred)

**Format**

A data frame with 4868 observations on the following 5 variables.

XL a numeric vector.  $XL = \min(TL, C)$  where TL is the time of the long term event, C is the censoring time.

DL a 0/1 vector.  $DL = 1 \cdot (TL < C)$  where TL is the time of the long term event, C is the censoring time.

XS a numeric vector.  $XS = \min(TS, C)$  where TS is the time of the long term event, C is the censoring time.

DS a 0/1 vector.  $DS = 1 \cdot (TS < C)$  where TS is the time of the long term event, C is the censoring time.

Z a 0/1 vector of discrete covariate values.

**Examples**

data(data\_example\_landpred)

---

fit_glm_normal	<i>Fit GLM with Normal Weights (No Short Covariate Info)</i>
----------------	--

---

**Description**

Fits a GLM for the probability of the event occurring before  $t_0 + \tau$ , given survival up to  $t_0$ , using only baseline covariates.

**Usage**

```
fit_glm_normal(landpred_obj, t0, tau)
```

**Arguments**

landpred_obj	A landpred object containing the data.
t0	The landmark time.
tau	The prediction window.

**Value**

A fitted glm object.

---

fit_short_glm	<i>Fit GLM with Kernel Weights (Short Covariate Info)</i>
---------------	---

---

**Description**

Fits a GLM for the probability of the event occurring before  $t_0 + \tau$ , given survival up to  $t_0$  and information on a short-term covariate. Uses kernel weighting based on the short-term covariate value.

**Usage**

```
fit_short_glm(landpred_obj, t0, tau, t_s, bw, transform, indices = NULL)
```

**Arguments**

landpred_obj	A landpred object containing the data.
t0	The landmark time.
tau	The prediction window.
t_s	The time of the short-term covariate measurement.
bw	The bandwidth for kernel weighting.
transform	A transformation function for the time variable (e.g., log).
indices	Optional indices to subset the data.

**Value**

A fitted glm object.

---

get_model	<i>Get Landpred Model</i>
-----------	---------------------------

---

**Description**

Fits the base GLM (no short covariate info) and creates a landpred model object.

Creates a landpred model object for a specific landmark time and prediction window. Dispatches to continuous or discrete model creation based on the landpred object type.

**Usage**

```
get_model(landpred_obj, t0, tau, bw = NULL, transform = identity)
```

**Arguments**

landpred_obj	A landpred object.
t0	The landmark time.
tau	The prediction window.
bw	The bandwidth (for continuous models).
transform	Transformation function (for continuous models).

**Value**

A landpred\_model\_continuous object.

A landpred\_model object (continuous or discrete).

---

Ghat.FUN	<i>Estimate Survival Function</i>
----------	-----------------------------------

---

**Description**

Estimate Survival Function

**Usage**

```
Ghat.FUN(tt, data, type = "fl", weight.given)
```

**Arguments**

tt	Time points.
data	Data frame.
type	Type of estimator.
weight.given	Optional weights.

---

landpred	<i>Create a Landpred Object</i>
----------	---------------------------------

---

## Description

Parses the formula and data to create a landpred object used for landmark prediction. Call ‘?landpred.pacakge’ for more information on the legacy API.

## Usage

```
landpred(formula, data, discrete = FALSE)
```

## Arguments

formula	A formula object with a Surv object on the LHS and covariates on the RHS.
data	The data frame.
discrete	Logical, whether to use the discrete method (legacy).

## Value

A landpred\_object.

## Examples

```
library(landpred)
library(survival)

# Load example data
data(data_example_landpred)

# Define landmark time and prediction window
t0 <- 2
tau <- 8

# Create a landpred object using the formula interface
# The formula specifies: Long-term survival ~ Short-term survival + Covariates
# Note: The short-term covariate must be a Surv object
obj <- landpred(
  Surv(XL, DL) ~ Surv(XS, DS) + Z,
  data = data_example_landpred,
  discrete = FALSE
)

# 1. Optimize bandwidth (Optional but recommended)
# This uses cross-validation to find the optimal bandwidth for the short-term covariate
# We use log transformation for the time variable as it's often more appropriate
bw <- optimize_bandwidth(
  landpred_obj = obj,
  t0 = t0,
  tau = tau,
  lower = 0.5,
  upper = 5,
  transform = log
)
```



```

)

print(paste("Optimal bandwidth:", bw))

# 2. Fit the model
# We pass the optimized bandwidth and the transformation used
model <- get_model(
  landpred_obj = obj,
  t0 = t0,
  tau = tau,
  bw = bw,
  transform = log
)

print(model)

# 3. Predict on new data
# For demonstration, we use the first 10 rows of the original data as "new data"
new_data <- data_example_landpred[1:10, ]

# The predict function expects a data frame with the same column names as used in the formula
probs <- predict(model, newdata = new_data)

print("Predicted probabilities:")
print(probs)

```

---

mse\_cv

---

*Calculate MSE for Bandwidth Selection using Cross-Validation*


---

## Description

Calculate MSE for Bandwidth Selection using Cross-Validation

## Usage

```

mse_cv(
  bw,
  landpred_obj,
  t0,
  tau,
  transform = identity,
  reps = 50,
  train_prop = 0.66
)

```

## Arguments

bw	The bandwidth to test.
landpred_obj	The landpred object.
t0	The landmark time.
tau	The prediction window.

transform	Transformation function for short-term covariate.
reps	Number of repetitions.
train_prop	Proportion of data to use for training.

**Value**

The Mean Squared Error.

---

optimize_bandwidth	<i>Optimize Bandwidth for Continuous Landpred Models</i>
--------------------	--

---

**Description**

Selects the optimal bandwidth by minimizing the Mean Squared Error (MSE) using cross-validation.

**Usage**

```
optimize_bandwidth(
  landpred_obj,
  t0,
  tau,
  lower = 0.05,
  upper = 5,
  transform = identity,
  reps = 50,
  train_prop = 0.66
)
```

**Arguments**

landpred_obj	A landpred object.
t0	The landmark time.
tau	The prediction window.
lower	Lower bound for bandwidth search.
upper	Upper bound for bandwidth search.
transform	Transformation function for the short-term covariate (e.g., log). Default is identity.
reps	Number of cross-validation repetitions. Default is 50.
train_prop	Proportion of data used for training in each fold. Default is 0.66.

**Value**

The optimal bandwidth.

---

`predict.landpred_model_continuous`*Predict Method for Landpred Continuous Model*

---

**Description**

Predicts the probability of the event occurring given new data.

**Usage**

```
## S3 method for class 'landpred_model_continuous'
predict(object, newdata = NULL, type = "response", ...)
```

**Arguments**

<code>object</code>	A <code>landpred_model_continuous</code> object.
<code>newdata</code>	New data frame containing covariates and short-term event info.
<code>type</code>	Type of prediction (default "response").
<code>...</code>	Additional arguments

**Value**

A vector of predicted probabilities.

---

`predict.landpred_model_discrete`*Predict Method for Discrete Landpred Model*

---

**Description**

Predicts probabilities using the discrete landpred model.

**Usage**

```
## S3 method for class 'landpred_model_discrete'
predict(object, newdata = NULL, ...)
```

**Arguments**

<code>object</code>	A <code>landpred_model_discrete</code> object.
<code>newdata</code>	Optional new data.
<code>...</code>	Additional arguments.

**Value**

Predicted probabilities.

---

```
print.landpred_model_continuous
```

*Print Method for Landpred Continuous Model*

---

### Description

Prints the continuous landpred model results.

### Usage

```
## S3 method for class 'landpred_model_continuous'  
print(x, ...)
```

### Arguments

x	A landpred_model_continuous object.
...	Additional arguments.

---

```
print.landpred_model_discrete
```

*Print Method for Discrete Landpred Model*

---

### Description

Prints the discrete landpred model results.

### Usage

```
## S3 method for class 'landpred_model_discrete'  
print(x, ...)
```

### Arguments

x	A landpred_model_discrete object.
...	Additional arguments.

---

```
print.landpred_object Print Method for Landpred Object
```

---

**Description**

Prints a summary of the landpred object.

**Usage**

```
## S3 method for class 'landpred_object'
print(x, ...)
```

**Arguments**

x	A landpred_object.
...	Additional arguments.

---

```
Prob.Covariate Calculate Probability with Covariate Information
```

---

**Description**

Calculates the probability of the event occurring before  $t_0 + \tau$ , given survival up to  $t_0$ , using a single covariate.

**Usage**

```
Prob.Covariate(t0, tau, data, weight = NULL, short = TRUE, newdata = NULL)
```

**Arguments**

t0	The landmark time.
tau	The prediction window.
data	The data frame for training.
weight	Optional weights.
short	Logical, whether the covariate is short-term.
newdata	Dataframe of new data for prediction.

**Value**

A landpred\_result object.

---

 Prob.Covariate.ShortEvent

*Calculate Probability with Short Event Information*


---

### Description

Calculates the probability of the event occurring before  $t_0 + \tau$ , given survival up to  $t_0$ , using information on a short-term event.

### Usage

```
Prob.Covariate.ShortEvent(
  t0,
  tau,
  data,
  weight = NULL,
  bandwidth = NULL,
  newdata = NULL
)
```

### Arguments

<code>t0</code>	The landmark time.
<code>tau</code>	The prediction window.
<code>data</code>	The data frame.
<code>weight</code>	Optional weights.
<code>bandwidth</code>	Bandwidth for kernel smoothing.
<code>newdata</code>	Optional new data for prediction.

### Value

A `landpred_result` object.

---

 Prob.Null

*Calculate Probability with No Information*


---

### Description

Calculates the probability of the event occurring before  $t_0 + \tau$ , given survival up to  $t_0$ , without using any covariate information.

### Usage

```
Prob.Null(t0, tau, data, weight = NULL, newdata = NULL)
```

**Arguments**

t0	The landmark time.
tau	The prediction window.
data	The data frame.
weight	Optional weights.
newdata	Optional new data for prediction.

**Value**

A landpred\_result object.

---

```
summary.landpred_model_continuous
```

*Summary Method for Landpred Continuous Model*

---

**Description**

Prints a summary of the model, including coefficients and standard errors.

**Usage**

```
## S3 method for class 'landpred_model_continuous'
summary(object, t_s = NULL, ...)
```

**Arguments**

object	A landpred_model_continuous object.
t_s	Optional short-term covariate time.
...	Additional arguments.

---

```
summary.landpred_object
```

*Summary Method for Landpred Object*

---

**Description**

Prints a detailed summary of the landpred object.

**Usage**

```
## S3 method for class 'landpred_object'
summary(object, ...)
```

**Arguments**

object	A landpred_object.
...	Additional arguments.

---

var.fun	<i>Estimate Variance of Coefficients</i>
---------	--

---

**Description**

Estimates the variance of the coefficients for the short-term GLM using a perturbation resampling method.

**Usage**

```
var.fun(t, data.v, tau, s, h, vmat, Ainv, weight = NULL, transform = identity)
```

**Arguments**

t	The target time for the short-term covariate (usually t_s).
data.v	The data frame used for estimation.
tau	The landmark time.
s	The prediction window.
h	The bandwidth.
vmat	A matrix of perturbation weights.
Ainv	The inverse information matrix from the model fit.
weight	Optional weights.
transform	Transformation function.

**Value**

A list containing the estimated standard errors for the intercept and slopes.



# Index

- \* **datasets**
  - data\_example\_landpred, [5](#)
- \* **internal**
  - coefficient\_se, [4](#)
  - fit\_glm\_normal, [6](#)
  - fit\_short\_glm, [6](#)
  - mse\_cv, [9](#)
  - var.fun, [16](#)
- \* **prediction accuracy**
  - AUC.landmark, [2](#)
  - BS.landmark, [3](#)
- \* **survival**
  - AUC.landmark, [2](#)
  - BS.landmark, [3](#)

AUC.landmark, [2](#)

BS.landmark, [3](#)

coef.landpred\_model\_continuous, [4](#)

coefficient\_se, [4](#)

cumsum2, [5](#)

data\_example\_landpred, [5](#)

fit\_glm\_normal, [6](#)

fit\_short\_glm, [6](#)

get\_model, [7](#)

Ghat.FUN, [7](#)

landpred, [8](#)

mse\_cv, [9](#)

optimize\_bandwidth, [10](#)

predict.landpred\_model\_continuous, [11](#)

predict.landpred\_model\_discrete, [11](#)

print.landpred\_model\_continuous, [12](#)

print.landpred\_model\_discrete, [12](#)

print.landpred\_object, [13](#)

Prob.Covariate, [13](#)

Prob.Covariate.ShortEvent, [14](#)

Prob.Null, [14](#)

summary.landpred\_model\_continuous, [15](#)

summary.landpred\_object, [15](#)

var.fun, [16](#)