

# Package ‘wevid’

December 16, 2018

**Type** Package

**Title** Quantifying Performance of a Binary Classifier Through Weight of Evidence

**Version** 0.5.2

**Date** 2018-12-16

## Description

The distributions of the weight of evidence (log Bayes factor) favouring case over noncase status in a test dataset (or test folds generated by cross-validation) can be used to quantify the performance of a diagnostic test (McKeigue (2018), <doi:10.1177/0962280218776989>). The package can be used with any test dataset on which you have observed case-control status and have computed prior and posterior probabilities of case status using a model learned on a training dataset. To quantify how the predictor will behave as a risk stratifier, the quantiles of the distributions of weight of evidence in cases and controls can be calculated and plotted.

**Depends** R (>= 2.10)

**License** GPL-3

**URL** <http://www.homepages.ed.ac.uk/pmckeigu/preprints/classify/wevidtutorial.html>

**LazyLoad** yes

**Imports** ggplot2, pROC, reshape2, zoo

**ByteCompile** TRUE

**Encoding** UTF-8

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Author** Paul McKeigue [aut] (<<https://orcid.org/0000-0002-5217-1034>>),  
Marco Colombo [ctb, cre] (<<https://orcid.org/0000-0001-6672-0623>>)

**Maintainer** Marco Colombo <[mar.colombo13@gmail.com](mailto:mar.colombo13@gmail.com)>

**Repository** CRAN

**Date/Publication** 2018-12-16 18:40:03 UTC

## R topics documented:

wevid-package . . . . .	2
plotcumfreqs . . . . .	3
plotroc . . . . .	4
plotWdists . . . . .	4
prop.belowthreshold . . . . .	5
recalibrate.p . . . . .	6
summary-densities . . . . .	6
Wdensities . . . . .	7
weightsofevidence . . . . .	9
wevid.datasets . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

wevid-package	<i>Quantifying performance of a diagnostic test using the sampling distribution of the weight of evidence favouring case over noncase status</i>
---------------	--

---

### Description

The **wevid** package provides functions for quantifying the performance of a diagnostic test (or any other binary classifier) by calculating and plotting the distributions in cases and noncases of the weight of evidence favouring case over noncase status.

### Details

The distributions of the weight of evidence (log Bayes factor) favouring case over noncase status in a test dataset (or test folds generated by cross-validation) can be used to quantify the performance of a diagnostic test.

In comparison with the C-statistic (area under ROC curve), the expected weight of evidence (expected information for discrimination) has several advantages as a summary measure of predictive performance. To quantify how the predictor will behave as a risk stratifier, the quantiles of the distributions of weight of evidence in cases and controls can be calculated and plotted.

This package can be used with any test dataset on which you have observed case-control status and have computed prior and posterior probabilities of case status using a model learned on a training dataset. Therefore, you should have computed on a test dataset (or on test folds used for cross-validation):

1. The prior probability of case status (this may be just the frequency of cases in the training data).
2. The posterior probability of case status (using the model learned on the training data to predict on the test data).
3. The observed case status (coded as 0 for noncases, 1 for cases).

The main function of the package is `Wdensities` which computes the crude and model-based densities of weight of evidence in cases and controls. Once these are computed, they can be plotted with `plotWdists` and `plotcumfreqs`. Summary statistics can be reported with `summary`.

**Author(s)**

Paul McKeigue <paul.mckeigue@ed.ac.uk>

**References**

Paul McKeigue (2018), Quantifying performance of a diagnostic test as the expected information for discrimination: Relation to the C-statistic. *Statistical Methods for Medical Research*, <https://doi.org/10.1177/0962280218776989>.

**See Also**

Useful links:

- <http://www.homepages.ed.ac.uk/pmckeigu/preprints/classify/wevidtutorial.html>

---

plotcumfreqs

*Plot the cumulative frequency distributions in cases and in controls*

---

**Description**

Plot the cumulative frequency distributions in cases and in controls

**Usage**

```
plotcumfreqs(densities)
```

**Arguments**

`densities`      Densities object produced by [Wdensities](#).

**Value**

A ggplot object representing the cumulative frequency distributions of the smoothed densities of the weights of evidence in cases and in controls.

**Examples**

```
data(cleveland)
densities <- with(cleveland, Wdensities(y, posterior.p, prior.p))
plotcumfreqs(densities)
```

---

`plotroc`*Plot crude and model-based ROC curves*

---

**Description**

While the crude ROC curve can be non-concave and is generally not smooth, the model-based ROC curve is always concave, as the corresponding densities have been adjusted to be mathematically consistent.

**Usage**

```
plotroc(densities)
```

**Arguments**

`densities`      Densities object produced by [Wdensities](#).

**Value**

A ggplot object representing crude and model-based ROC curves.

**Examples**

```
data(cleveland)
densities <- with(cleveland, Wdensities(y, posterior.p, prior.p))
plotroc(densities)
```

---

`plotWdists`*Plot the distribution of the weight of evidence in cases and in controls*

---

**Description**

Plot the distribution of the weight of evidence in cases and in controls

**Usage**

```
plotWdists(densities, distlabels = c("Crude", "Model-based"))
```

**Arguments**

`densities`      Densities object produced by [Wdensities](#).  
`distlabels`      Character vector of length 2 to be used to label the crude and the model-based curves (in that order).

**Value**

A ggplot object representing the distributions of crude and model-based weights of evidence in cases and in controls.

**Examples**

```
data(cleveland)
densities <- with(cleveland, Wdensities(y, posterior.p, prior.p))
plotWdists(densities)

# Example which requires fitting a mixture distribution
data(fitonly)
densities <- with(fitonly, Wdensities(y, posterior.p, prior.p,
                                     in.spike=posterior.p < 0.1))

# truncate spike
plotWdists(densities) + ggplot2::scale_y_continuous(limits=c(0, 0.5))
```

---

prop.belowthreshold	<i>Proportions of cases and controls below a threshold of weight of evidence</i>
---------------------	--

---

**Description**

Proportions of cases and controls below a threshold of weight of evidence

**Usage**

```
prop.belowthreshold(densities, w.threshold)
```

**Arguments**

`densities`      Densities object produced by [Wdensities](#).

`w.threshold`    Threshold value of weight of evidence (natural logs).

**Value**

Numeric vector of length 2 listing the proportions of controls and cases with weight of evidence below the given threshold.

**Examples**

```
data(cleveland)
densities <- with(cleveland, Wdensities(y, posterior.p, prior.p))
w.threshold <- log(4) # threshold Bayes factor of 4
prop.belowthreshold(densities, w.threshold)
```

---

recalibrate.p	<i>Recalibrate posterior probabilities</i>
---------------	--

---

**Description**

Transforms posterior probabilities to logits, fits a logistic regression model and returns the predictive probabilities from this model.

**Usage**

```
recalibrate.p(y, posterior.p)
```

**Arguments**

y	Binary outcome label (0 for controls, 1 for cases).
posterior.p	Vector of posterior probabilities.

**Value**

Recalibrated posterior probabilities.

---

summary-densities	<i>Summary evaluation of predictive performance</i>
-------------------	---

---

**Description**

Summary evaluation of predictive performance

**Usage**

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

## S3 method for class 'Wdensities'
mean(x, ...)

auroc.crude(densities)

auroc.model(densities)

lambda.crude(densities)

lambda.model(densities)
```

**Arguments**

object, x, densities  
Densities object produced by [Wdensities](#).

... Further arguments passed to or from other methods. These are currently ignored.

**Value**

summary returns a data frame that reports the number of cases and controls, the test log-likelihood, the crude and model-based C-statistic and expected weight of evidence Lambda.

mean returns a numeric vector listing the mean densities of the weight of evidence in controls and in cases.

auroc.crude and auroc.model return the area under the ROC curve according to the crude and the model-based densities of weight of evidence, respectively.

lambda.crude and lambda.model return the expected weight of evidence (expected information for discrimination) in bits from the crude and the model-based densities, respectively.

**Examples**

```
data(cleveland)
densities <- with(cleveland, Wdensities(y, posterior.p, prior.p))

summary(densities)
mean(densities)
auroc.model(densities)
lambda.model(densities)
```

---

Wdensities

*Compute densities of weights of evidence in cases and controls*

---

**Description**

The function computes smoothed densities of the weight of evidence in cases and in controls from the crude probabilities, then adjusts them to make them mathematically consistent.

**Usage**

```
Wdensities(y, posterior.p, prior.p, range.xseq = c(-25, 25),
  x.stepsize = 0.01, adjust.bw = 1, in.spike = NULL,
  recalibrate = TRUE, debug = FALSE)
```





---

weightsofevidence	<i>Calculate weights of evidence in natural log units</i>
-------------------	---

---

### Description

Calculate weights of evidence in natural log units

### Usage

```
weightsofevidence(posterior.p, prior.p)
```

### Arguments

posterior.p	Vector of posterior probabilities generated by using model to predict on test data.
prior.p	Vector of prior probabilities.

### Value

The weight of evidence in nats for each observation.

### Examples

```
data(cleveland) # load example dataset
W <- with(cleveland, weightsofevidence(posterior.p, prior.p))
```

---

wevid.datasets	<i>Example datasets</i>
----------------	-------------------------

---

### Description

Example datasets

The **wevid** package comes with the following dataset:

- `cleveland` is based on cross-validated prediction of coronary disease in the Cleveland Heart Study (297 observations).
- `pima` is based on cross-validated prediction of diabetes in Pima Native Americans (768 observations).
- `fitonly` is based on cross-validated prediction of colorectal cancer from fecal immunochemical test (FIT) only in Michigan (242 observations). As most controls and some cases have zero values in the FIT test, to fit densities to the sampled values of weight of evidence in controls and cases it is necessary to specify spike-slab mixtures.

**Format**

Each dataset consists of a data frame with the following variables:

- `prior.p`: Prior probabilities of case status.
- `posterior.p`: Posterior probabilities of case status.
- `y`: Case-control status.

**Source**

<http://www.homepages.ed.ac.uk/pmckeigu/preprints/classify/wevidtutorial.html>

# Index

## \*Topic **datasets**

- wevid.datasets, 9
- auroc.crude (summary-densities), 6
- auroc.model (summary-densities), 6
- cleveland (wevid.datasets), 9
- fitonly (wevid.datasets), 9
- lambda.crude (summary-densities), 6
- lambda.model (summary-densities), 6
- mean.Wdensities (summary-densities), 6
- pima (wevid.datasets), 9
- plotcumfreqs, 2, 3
- plotroc, 4
- plotWdists, 2, 4
- prop.belowthreshold, 5
- recalibrate.p, 6
- summary, 2
- summary-densities, 6
- summary.Wdensities (summary-densities), 6
- Wdensities, 2-5, 7, 7
- weightsofevidence, 9
- wevid (wevid-package), 2
- wevid-package, 2
- wevid.datasets, 9