Create a confusion matrix

Description

Calculates a cross-tabulation of observed and predicted classes with associated statistics.

Usage

```
confusionMatrix(data, ...)

## Default S3 method:
confusionMatrix(data, reference, positive = NULL,
  dnn = c("Prediction", "Reference"), prevalence = NULL,
  mode = "sens_spec", ...)

## S3 method for class 'table'
confusionMatrix(data, positive = NULL,
  prevalence = NULL, mode = "sens_spec", ...)
```

Arguments

`data`	a factor of predicted classes (for the default method) or an object of class `table`.
`...`	options to be passed to `table`. NOTE: do not include `dnn` here
`reference`	a factor of classes to be used as the true results
`positive`	an optional character string for the factor level that corresponds to a "positive" result (if that makes sense for your data). If there are only two factor levels, the first level will be used as the "positive" result. When `mode = "prec_recall"`, `positive` is the same value used for `relevant` for functions `precision`, `recall`, and `F_meas.table`.
`dnn`	a character vector of dimnames for the table
`prevalence`	a numeric value or matrix for the rate of the "positive" class of the data. When `data` has two levels, `prevalence` should be a single numeric value. Otherwise, it should be a vector of numeric values with elements for each class. The vector should have names corresponding to the classes.
`mode`	a single character string either "sens_spec", "prec_recall", or "everything"

Details

The functions requires that the factors have exactly the same levels.

For two class problems, the sensitivity, specificity, positive predictive value and negative predictive value is calculated using the `positive` argument. Also, the prevalence of the "event" is computed from the data (unless passed in as an argument), the detection rate (the rate of true events also predicted to be events) and the detection prevalence (the prevalence of predicted events).

Suppose a 2x2 table with notation

	Reference	
Predicted	Event	No Event
Event	A	B
No Event	C	D

The formulas used here are:

$$Sensitivity = A/(A+C)$$

$$Specificity = D/(B+D)$$

$$Prevalence = (A+C)/(A+B+C+D)$$

$$PPV = (sensitivity * prevalence)/((sensitivity*prevalence) + ((1-specificity)*(1-prevalence)))$$

$$NPV = (specificity * (1-prevalence))/(((1-sensitivity)*prevalence) + ((specificity)*(1-prevalence)))$$

$$Detection\ Rate = A/(A+B+C+D)$$

$$Detection\ Prevalence = (A+B)/(A+B+C+D)$$

$$Balanced\ Accuracy = (sensitivity+specificity)/2$$

$$Precision = A/(A+B)$$

$$Recall = A/(A+C)$$

$$F1 = (1+beta^2)*precision*recall/((beta^2 * precision)+recall)$$

where `beta = 1` for this function.

See the references for discussions of the first five formulas.

For more than two classes, these results are calculated comparing each factor level to the remaining levels (i.e. a "one versus all" approach).

The overall accuracy and unweighted Kappa statistic are calculated. A p-value from McNemar's test is also computed using `mcnemar.test` (which can produce `NA` values with sparse tables).

The overall accuracy rate is computed along with a 95 percent confidence interval for this rate (using `binom.test`) and a one-sided test to see if the accuracy is better than the "no information rate," which is taken to be the largest class percentage in the data.

Value

a list with elements

`table`	the results of `table` on `data` and `reference`
`positive`	the positive result level
`overall`	a numeric vector with overall accuracy and Kappa statistic values
`byClass`	the sensitivity, specificity, positive predictive value, negative predictive value, precision, recall, F1, prevalence, detection rate, detection prevalence and balanced accuracy for each class. For two class systems, this is calculated once using the `positive` argument

Note

If the reference and data factors have the same levels, but in the incorrect order, the function will reorder them to the order of the data and issue a warning.

Author(s)

Max Kuhn

References

Kuhn, M. (2008), “Building predictive models in R using the caret package, ” *Journal of Statistical Software*, (http://www.jstatsoft.org/article/view/v028i05/v28i05.pdf).

Altman, D.G., Bland, J.M. (1994) “Diagnostic tests 1: sensitivity and specificity,” *British Medical Journal*, vol 308, 1552.

Altman, D.G., Bland, J.M. (1994) “Diagnostic tests 2: predictive values,” *British Medical Journal*, vol 309, 102.

Velez, D.R., et. al. (2008) “A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction.,” *Genetic Epidemiology*, vol 4, 306.

See Also

`as.table.confusionMatrix`, `as.matrix.confusionMatrix`, `sensitivity`, `specificity`, `posPredValue`, `negPredValue`, `print.confusionMatrix`, `binom.test`

Examples

```
###################
## 2 class example

lvs <- c("normal", "abnormal")
truth <- factor(rep(lvs, times = c(86, 258)),
                levels = rev(lvs))
pred <- factor(
               c(
                 rep(lvs, times = c(54, 32)),
                 rep(lvs, times = c(27, 231))),
               levels = rev(lvs))

xtab <- table(pred, truth)

confusionMatrix(xtab)
confusionMatrix(pred, truth)
confusionMatrix(xtab, prevalence = 0.25)

###################
## 3 class example

confusionMatrix(iris$Species, sample(iris$Species))

newPrior <- c(.05, .8, .15)
names(newPrior) <- levels(iris$Species)

confusionMatrix(iris$Species, sample(iris$Species))
```