



# Performance preditiva com: **metrica**

## Básico

**metrica** é um compilado de mais de 80 funções elaboradas para avaliar quantitativa e visualmente a performance preditiva de regressões (dados contínuos) e classificações (categóricos) em modelos de simulação e predição (e.g., APSIM, DSSAT, DND, Aprendizado de máquina supervisionado).

## Usando as funções

Existem dois argumentos básicos comuns à todas as funções do **metrica**: (i) **obs** (O; vetor referente à valores observados ou reais, medidos, verdadeiros, alvos, etiquetas), e (ii) **pred** (P; preditos ou simulados, ajustados, modelados, estimados). Argumentos opcionais incluem **data**, que permite incluir uma tabela contendo ambos, valores observados e preditos, e **tidy**, que controla o tipo de saída, como lista (**tidy = FALSE**) ou como um **data.frame** (**tidy = TRUE**).

## Instalação

```
install.packages("metrica")
```

Você pode instalar a versão em desenvolvimento no [GitHub](#) com

```
#install.packages("devtools")
devtools::install_github("adriancorrendo/metrica")
```

## Bancos de dados nativos

O pacote **metrica** acompanha quatro bancos de dados como exemplo de variáveis contínuas (regressão) do software APSIM

- **Wheat**: 137 dados de nitrogênio em grãos de trigo;
- **Barley**: 69 dados de números de grãos em cevada;
- **Sorghum**: 36 dados de números de grãos em sorgo;
- **Chickpea**: 39 dados de massa seca aérea em grão-de-bico.

Além disso, **metrica** também provê dois exemplos nativos de variáveis categóricas (classificação):

- **land\_cover**: dados binários de cobertura do solo usando imagens de satélite. Valores: 1 = vegetação, 0 = outra cobertura/uso do solo.
- **maize\_phenology**: banco de dados de fenologia do milho (*Zea mays*L.) com 16 estágios de desenvolvimento da cultura.

Confira a documentação do **metrica** para encontrar todas as métricas de performance e seus detalhes: [metrica](#)

## Regressão

```
R2(data = wheat, obs = obs, pred=pred, tidy = TRUE)
#> R2
#> [1] 0.8455538
```

```
RMSE(data = wheat, obs = obs, pred = pred)
#> $RMSE
#> [1] 1.666441
```

```
KGE(data = wheat, obs = obs, pred = pred)
#> $KGE
#> [1] 0.9106471
```

Usuários também podem calcular **todas (padrão)** ou uma **lista de métricas** selecionadas de uma só vez usando **metrics\_summary()**:

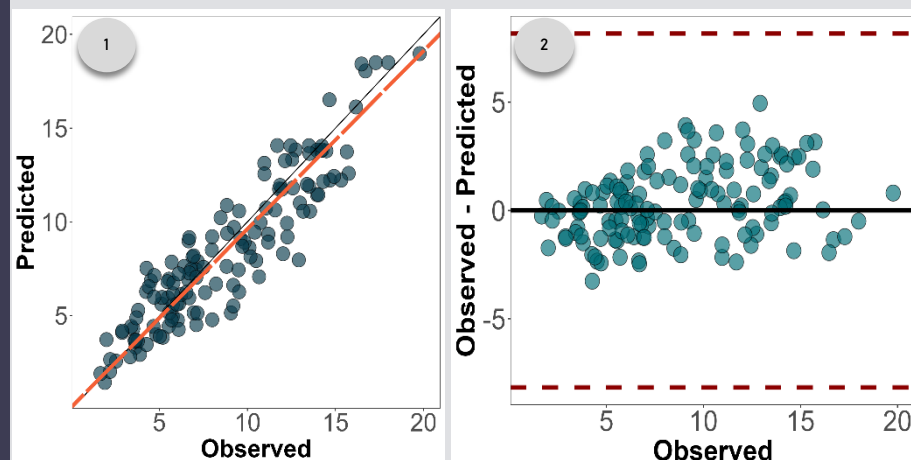
```
sel_r_metrics <- c("R2", "MBE", "RMSE", "RSR", "NSE",
                  "KGE", "CCC")
```

```
metrics_summary(data = wheat,
                obs = obs,
                pred = pred,
                type = "regression",
                metrics_list = sel_r_metrics)
```

## Gráficos

```
1. scatter_plot(data = wheat,
                obs = obs, pred = pred)
```

```
2. bland_altman_plot(data = wheat,
                    obs = obs, pred = pred)
```



## Classificação

```
accuracy(data=maize_phenology, obs=actual, pred=predicted)
#> $accuracy
#> [1] 0.8834951
```

```
precision(data=maize_phenology, obs=actual, pred=predicted)
#> $precision
#> [1] 0.8335108
```

```
recall(data = maize_phenology, obs=actual, pred=predicted)
#> $recall
#> [1] 0.8405168
```

Para classificação, usuários podem aplicar a função **metrics\_summary()** para obter várias métricas de uma só vez

```
sel_c_metrics <- c("accuracy", "precision", "recall",
                  "fscore")
```

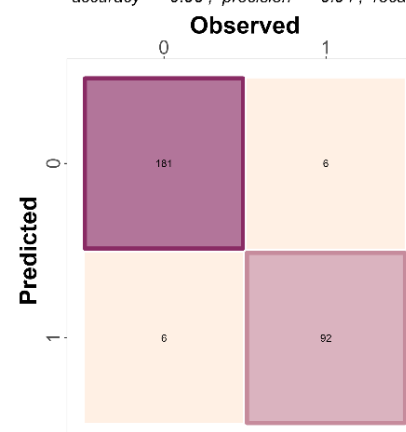
```
metrics_summary(data = landcover,
                obs = actual, pred = predicted,
                type = "classification",
                metrics_list = sel_c_metrics,
                pos_level = 1)
```

## Matriz de confusão

```
confusion_matrix(data = .,
                 obs = labels, pred = predictions,
                 plot = TRUE,
                 unit="count")
```

### Binomial case

Performance metrics:  
accuracy = 0.96 ; precision = 0.94 ; recall = 0.94 .



### Multinomial case

Performance metrics:  
accuracy = 0.43 ; precision = 0.43 ; recall = 0.43 .

