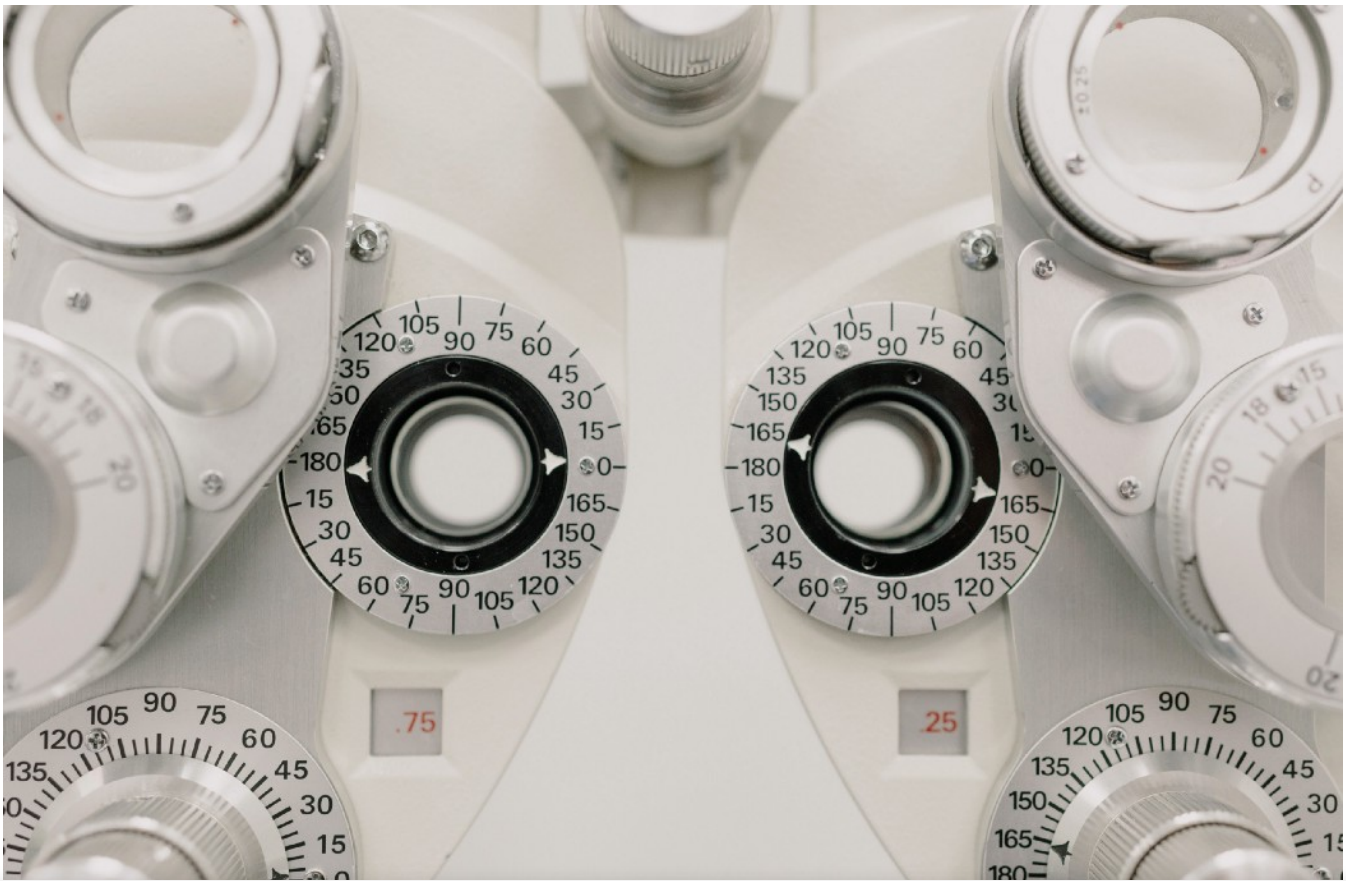
Konstantin Rink Follow
Oct 21, 2021 · 13 min read ★ · Listen

HANDS-ON TUTORIALS

Time Series Forecast Error Metrics You Should Know

An overview and introduction to the most common error metrics.



Using the right error metrics in your Data Science project is crucial. A **wrong error metric will** not only **affect** your **model's optimization** (loss function) but also might **skew your judgment** of models.

Besides the classical error metrics like Mean Absolute Error, more and more new error metrics are being developed and published regularly.

The idea of this article is to not only provide you an overview about the most used ones but also show you how they are calculated as well as their advantages and disadvantages.

Before we start, please keep in mind that **there is no silver bullet**, **no single best error metric**. The **fundamental challenge** is, that every statistical measure **condenses a large number of data into a single value**, so it only provides one projection of the model errors emphasizing a certain aspect of the error characteristics of the model performance (Chai and Draxler 2014).

Therefore it is better to have a **more practical and pragmatic view** and work with a **selection of metrics** that fit for your use case or project.

To identify the most used or common error metrics, I screened over 12 time series forecasting frameworks or libraries (i.e. kats, sktime, darts) and checked what error metrics they offer. Out of these 12 I identified the top 8 **most common forecasting error metrics** and grouped them into **four categories** (see figure 1) proposed by Hyndman and Koehler (2006).

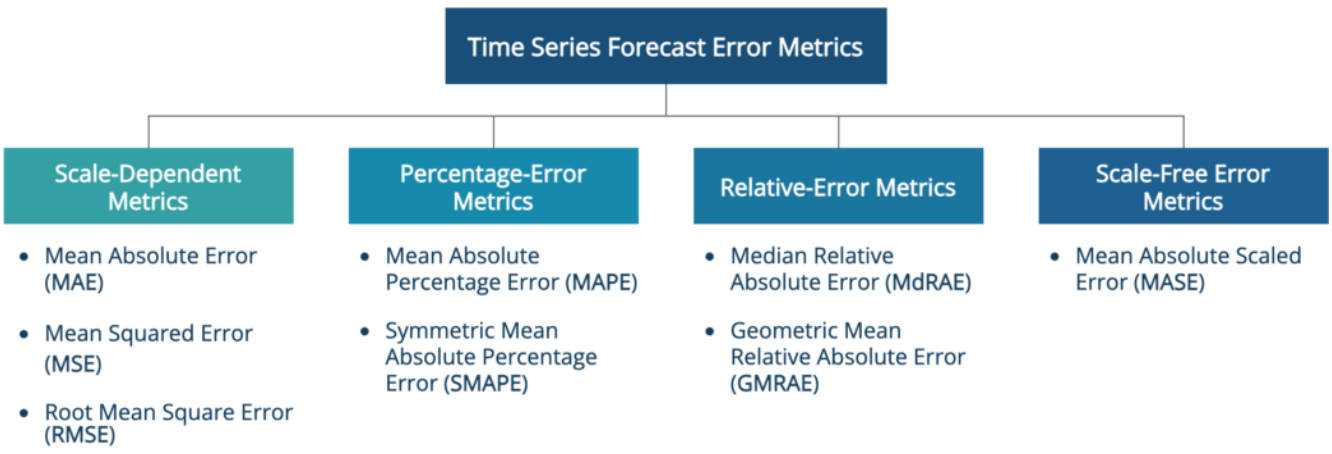


Figure 1 Overview Time Series Forecast Error Metrics (image by author)

Scale Dependent Metrics

Many popular metrics are referred to as **scale-dependent** (Hyndman, 2006). Scale-dependent means the error metrics are **expressed in the units** (i.e. Dollars, Inches, etc.) of the underlying data.

The **main advantage** of scale dependent metrics is that they are usually **easy to calculate** and **interpret**. However, they can **not be used to compare different series**, because of their **scale dependency** (Hyndman, 2006).

Please note here that Hyndman (2006) includes Mean Squared Error into a scale-dependent group (claiming that the error is "on the same scale as the data"). However, Mean Squared Error has a dimension of the squared scale/unit. To bring MSE to the data's unit we need to take the square root which leads to another metric, the RMSE. (Shcherbakov et al., 2013)

Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

```python
import numpy as np

def mae(y, y_hat):
    return np.mean(np.abs(y - y_hat))
```

ts_error_metrics_mae.py hosted with ❤ by GitHub view raw

The Mean Absolute Error (MAE) is calculated by taking the mean of the absolute differences between the actual values (also called y) and the predicted values (y_hat).

Simple, isn't it? And that's its major advantage. It is **easy to understand** (even for business users) and **to compute**. It is recommended for **assessing accuracy on a single series** (Hyndman, 2006). **However** if you want to compare different series (with different units) it is not suitable. Also you should **not use it** if you want to **penalize outliers**.


$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

```python
import numpy as np

def mse(y, y_hat):
    return np.mean(np.square(y - y_hat))
```

ts_error_metrics_mse.py hosted with ❤ by GitHub view raw

If you want to put **more attention on outliers** (huge errors) you can consider the Mean Squared Error (MSE). Like it's name implies it takes the mean of the squared errors (differences between y and y_hat). Due to its squaring, it **heavily weights large errors more than small ones**, which can be in some situations a **disadvantage**. Therefore the MSE is suitable for situations where you **really want to focus on large errors**. Also keep in mind that due to its squaring the metric **loses its unit**.

Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

```python
import numpy as np

def rmse(y, y_hat):
    return np.sqrt(np.mean(np.square(y - y_hat)))
```

ts_error_metrics_rmse.py hosted with ❤ by GitHub view raw

To **avoid the** MSE's **loss of its unit** we can take the square root of it. The outcome is then a new error metric called the Root Mean Squared Error (RMSE).

It comes with the same **advantages** as its siblings MAE and MSE. However, like MSE, it is also **sensitive to outliers**.

However, Chai and Drexler (2014) partially refuted their arguments and **recommend RMSE over MAE for your model optimization** as well as for **evaluating different models where the error distribution is expected to be Gaussian**.

Percentage Error Metrics

As we know from the previous chapter, **scale dependent metrics are not suitable** for **comparing different time series**.

Percentage Error Metrics solve this problem. They are **scale independent** and used to **compare forecast performance between different time series**. However, their **weak spots are zero values in a time series**. Then they become **infinite or undefined** which makes them **not interpretable** (Hyndman 2006).

Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i} * 100\right|$$

```python
import numpy as np

def mape(y, y_hat):
    return np.mean(np.abs((y - y_hat)/y)*100)
```

ts_error_metrics_mape.py hosted with ❤ by GitHub view raw

The mean absolute percentage error (MAPE) is one of the **most popular used error metrics** in time series forecasting. It is calculated by taking the average (mean) of the absolute difference between actuals and predicted values divided by the actuals.

> *Please note, some MAPE formulas do not multiply the result(s) with 100. However, the MAPE is presented as a percentage unit so I added the multiplication.*

MAPE's **advantages** are it's **scale-independency** and **easy interpretability**. As said at

However, MAPE also comes with some **disadvantages**. First, **it generates infinite or undefined values for zero or close-to-zero actual values** (Kim and Kim 2016).

Second, it also puts a **heavier penalty on negative than on positive errors** which leads to an **asymmetry** (Hyndman 2014).

And last but not least, MAPE **can not be used when using percentages make no sense**. This is for example the case when measuring temperatures. The units Fahrenheit or Celsius scales have relatively arbitrary zero points, and it makes no sense to talk about percentages (Hyndman and Koehler, 2006).

Symmetric Mean Absolute Percentage Error (sMAPE)

To **avoid the asymmetry** of the MAPE a new error metric was proposed. The Symmetric Mean Absolute Percentage Error (sMAPE). The sMAPE is probably one of the **most controversial** error metrics, since not only different definitions or formulas exist but also critics claim that this metric **is not symmetric** as the name suggests (Goodwin and Lawton, 1999).

The original idea of an "**adjusted MAPE**" was proposed by Armstrong (1985). **However** by his definition the **error metric can be negative or infinite** since the values in the denominator **are not set absolute** (which is then correctly mentioned as a disadvantage in some articles that follow his definition).

$$\overline{\text{MAPE}} = 100\text{mean}(2|y_t - \hat{y}_t|/(y_t + \hat{y}_t))$$

Makridakis (1993) proposed a similar metric and called it SMAPE. His formula which can be seen below **avoids the problems Armstrong's formula** had by setting the values in the denominator to absolute (Hyndman, 2014).

$$sMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2}$$

> *Note: Makridakis (1993) proposed the formula above in his paper "Accuracy measures: theoretical and practical concerns". Later in his publication (Makridakis and Hibbon,*


The sAMPE is the average across all forecasts made for a given horizon. It's **advantages** are that it **avoids MAPE's problem of large errors** when y-values are close to zero and the large difference between the absolute percentage errors when y is greater than y-hat and vice versa. Unlike MAPE which has no limits, **it fluctuates between 0% and 200%** (Makridakis and Hibon, 2000).

For the **sake of interpretation** there is also a slightly **modified version of SMAPE** that **ensures** that the metric's results **will be always between 0% and 100%**:

$$sMAPE = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)}$$

The following code snippet contains the sMAPE metric proposed by Makridakis (1993) and the modified version.


As mentioned at the beginning, there are **controversies around the sMAPE**. And **they are true**. Goodwin and Lawton (1999) pointed out that sMAPE **gives more penalties to under-estimates** more **than to over-estimates** (Chen et al., 2017). Cánovas (2009) proofs this fact with **an easy example**.

	case	y	y_hat	abs difference	MAPE	sMAPE
0	1	100	150	50	50	20
1	2	150	100	50	33	20

Table 1. Example with a symmetric sMAPE.

	case	y	y_hat	abs difference	MAPE	sMAPE
0	1	100	150	50	50	20
1	2	100	50	50	50	33

Table 2. Example with an asymmetric sMAPE.

Starting with **table 1** we have two cases. In **case 1** our actual value **y** is **100** and the prediction **y_hat 150**. This leads to a sMAPE value of 20 %. **Case 2** is the opposite. Here we have an actual value **y** of **150** and a prediction **y_hat** of **100**. This also leads to a sMAPE of 20 %. So far it seems symmetry is given…

Let us now have a look at **table 2**. We also have here two cases and as you can already see the sMAPE values **are not the same anymore**. The second case **leads to a different SMAPE value** of 33 %.

Modifying the forecast while holding fixed actual values and **absolute deviation do not produce the same sMAPE's value**. **Simply biasing the model without improving its accuracy should never produce different error values** (Cánovas, 2009).


Compared to the error metrics explained before, **relative error metrics compare your model's performance** (so it's errors) **with the performance of a baseline or benchmark model**.

The most common benchmark models are *naive*, *snaive* and the *mean* of all observations.

In a *naive* or *random walk model* the **prediction is just equal to the previous observation**.

If you have **seasonal data**, it is useful to **choose the *snaive method***. The *snaive method* sets each forecast to be **equal to the last observed value from the same season of the year** (e.g., the same month of the previous year). It is defined as follows:

$$\hat{y}_{T+h|T} = y_{T+h-m(k+1)}$$

where *m* is the seasonal period, and *k* the integer part of *(h-1)/m* (i.e., the number of complete years in the forecast period prior to time T+h). **For monthly data** this would mean that the **forecast for all future October values** is **equal to the last observed October value** (Hyndman and Athanasopoulos, 2018)

Due to their **scale-independence**, these metrics **were recommended** in studies by Armstrong and Collopy (1992) and by Fildes (1992) for assessing forecast accuracy across multiple series. **However**, when the **calculated errors** are **small**, the use of the *naive method* as a benchmark is **no longer possible** because it would lead **to division by zero** (Hyndman, 2006).

Median Relative Absolute Error (MdRAE)

$$MdRAE = Median_{i=1,n}\left(\frac{\mid y_i - \hat{y}_i \mid}{\mid y_i - b_i \mid}\right)$$

$$b_i = \begin{cases} y_{t-1} & \text{Non} - \text{Seasonal} \\ y_{t-M} & \text{Seasonal} \end{cases}$$


As mentioned in the introduction to this section, **relative error metrics compare** our **model's performance** (forecast) with a **benchmark method** (i.e. *random walk*). The Median Relative Absolute Error (MdRAE) calculates the median of the difference between the absolute error of our forecast to the absolute error of a benchmark model.

If our model's forecast **equals to the benchmark's forecast then the result is 1**. If the benchmarks forecast are **better than ours** then the result will be **above > 1**. If **ours is better** than it's **below 1**.

Since we are calculating the median, the MdRAE is **more robust to outliers** as other


Compared to the error metrics before, the relative error metrics are a bit **more complex to calculate and interpret**. Let's have an example to strengthen our understanding.

	y	y_hat	y_bnchmrk
0	4150	4000	5430
1	4253	4400	5430
2	4107	4100	5430
3	4582	4400	5430
4	4728	4600	5430
5	5720	5600	5430

Table 3. MdRAE calculation example (image by author).

Table 3 shows our actual values y, the predictions of our model y_hat and the forecasts from a *naive* benchmark model y_bnchmrk that used the last point from our training data set (see code above) as the prediction. Of course there are also other options to calculate the benchmark's predictions (e.g. including seasonality, drift or just taking the mean of the training data).

The MdRAE then takes the median of the difference between the absolute error (y-y_hat) of our forecast divided by the absolute error (y-y_bnchmrk) of our benchmark model.

The result is 0.15 which is obviously **smaller than 1** so **our forecast is better than the one from the benchmark model**.

Geometric Mean Relative Absolute Error (GMRAE)

$$GMRAE = exp\left(\frac{1}{n}\sum_{i=1}^{n} ln\left(\frac{|\ y_i - \hat{y}\ |}{|\ y_i - b_i\ |}\right)\right) = \sqrt[n]{\prod_{i=1}^{n}\left(\frac{|\ y_i - \hat{y}_i\ |}{|\ y_i - b_i\ |}\right)}$$

$$b_i = \begin{cases} y_{t-1} & \text{Non} - \text{Seasonal} \\ y_{t-M} & \text{Seasonal} \end{cases}$$

where *bi* is benchmark forecast results and *M* is the seasonal period in our time series.

Like the MdRAE the Geometric Mean Relative Absolute Error (GMRAE) **compares** the **errors of our forecast with** the one of a **defined baseline model**. However, **instead of calculating the median**, the GMRAE, as the name implies, calculates the **geometric mean** of our relative errors.

A GMRAE above **1 states that the benchmark is better**, a **result below 1** indicates **that our model's forecast performs better**.

Taking an arithmetic mean of log-scaled error ratios (see alternative representation) makes the GMRAE **more resistant to outliers**. **However**, GMRAE is **still sensitive to outliers**. It can be dominated by not only a **single large outlier**, but also an **extremely small error close to zero**. This is because there is **neither upper bound nor lower bound for the log-scaled error** ratios calculated by GMRAE (Chen and Twycross, 2017). If the **error of the benchmark method** is **zero** then a **large value is returned**


Scale-Free Error Metrics

Relative measures try to **remove the scale** of the data by **comparing** the **forecasts** with **those obtained from some benchmark** (*naive*) method. **However**, **they have problems**. **Relative errors** have a statistical distribution with **undefined mean and infinite variance**. They can only be computed when there are **several forecasts** on the **same series**, and so **cannot be used to measure out-of-sample forecast accuracy at a single forecast horizon** (Hyndman and Koehler, 2006).

To **solve this issue**, Hyndman and Koehler (2006) proposed a new kind of metric — the **scale free error metric**. Their idea was to **scale the error based on the in-sample MAE from a naive** (*random walk*) forecast method.

Mean Absolute Scaled Error (MASE)

$$MASE = \frac{MAE}{MAE_{in-sample,\,naive}}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$


The MASE is calculated by taking the MAE and dividing it by the MAE of an in-sample (so based on our training data) *naive* benchmark.

Values of MASE greater **than 1 indicate that the forecasts are worse**, on average, than in-sample one-step forecasts from the *naive model* (Hyndman and Koehler, 2006).

Since it is a **scale free metric** one is able to **compare the model's accuracy across (scale) different time series**. Unlike the relative error metrics it **does not give undefined or infinite values** which makes it a **suitable metric for time series data with zeros**. The **only case** under which the MASE would be infinite or undefined is **when all historical observations are equal** or **all of the actual values** during the **in-sample period were zeros** (Kim and Kim, 2016).

However there are also some **critical voices**. Davydenko and Fildes (2013) argue that MASE **introduces a bias towards overrating the performance** of a benchmark forecast as a result of **arithmetic averaging** and MASE is **vulnerable to outliers**, as a result of **dividing by small benchmark MAE** values. Also due the fact that the MAE in the denominator is using **in-sample data** the metric might be **more tricky to explain to business users** as other (more simple) metrics.

~~Confusion~~ Conclusion

As you have seen there is **no silver bullet**, **no single best error metric**. Each category or metric has its **advantages** and **weaknesses**. So it **always depends on your individual use case** or **purpose** and **your underlying data**. It is important **not to just**


If all series **are on the same scale**, the **data preprocessing procedures** were performed (data cleaning, anomaly detection) and the task is **to evaluate the forecast performance** then the **MAE can be preferred** because it is simpler to explain (Hyndman and Koehler, 2006; Shcherbakov et al., 2013)

Chai and Draxler (2014) recommend to **prefer RMSE over MAE** when the error distribution is **expected to be Gaussian**.

In case the data **contain outliers** it is advisable to apply scaled measures like **MASE**. In this situation the **horizon should be large enough**, **no identical values should be**, the normalized factor **should be not equal to zero** (Shcherbakov et al., 2013).

The introduced error metrics may be the common ones **but this does not imply that they are the best for your use case**. Also as I mentioned **new error metrics** like average relative MAE (AvgRelMAE) or Unscaled Mean Bounded Relative Absolute Error (UMBRAE) are being developed and published frequently. **So it is definitely worth to have a look at these metrics**, **what they are trying to improve** (e.g. becoming more robust or symmetric) and **how they might be suitable for your project**.

Bibliography

Armstrong, J. Scott, and Fred Collopy. 1992. “Error Measures for Generalizing about Forecasting Methods: Empirical Comparisons.” *International Journal of Forecasting* 8(1). doi: 10.1016/0169–2070(92)90008-W.

Chai, T., and R. R. Draxler. 2014. “Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)? — Arguments against Avoiding RMSE in the Literature.” *Geoscientific Model Development* 7(3). doi: 10.5194/gmd-7–1247–2014.

Chen, Chao, Jamie Twycross, and Jonathan M. Garibaldi. 2017. “A New Accuracy Measure Based on Bounded Relative Error for Time Series Forecasting.” *PLOS ONE* 12(3). doi: 10.1371/journal.pone.0174202.


Hyndman, Rob. 2006. “Another Look at Forecast Accuracy Metrics for Intermittent Demand.” *Foresight: The International Journal of Applied Forecasting* 4:43–46.

Hyndman, Rob J., and Anne B. Koehler. 2006. “Another Look at Measures of Forecast Accuracy.” *International Journal of Forecasting* 22(4). doi: 10.1016/j.ijforecast.2006.03.001.

Hyndman, Robin John, and George Athanasopoulos. 2018. *Forecasting: Principles and Practice*. 2nd ed. OTexts.

Kim, Sungil, and Heeyoung Kim. 2016. “A New Metric of Absolute Percentage Error for Intermittent Demand Forecasts.” *International Journal of Forecasting* 32(3):669–79. doi: https://doi.org/10.1016/j.ijforecast.2015.12.003.

Makridakis, Spyros, and Michèle Hibon. 2000. “The M3-Competition: Results, Conclusions and Implications.” *International Journal of Forecasting* 16(4):451–76. doi: https://doi.org/10.1016/S0169-2070(00)00057-1.

Shcherbakov, Maxim V., Adriaan Brebels, Anton Tyukov, Timur Janovsky, and Valeriy Anatol. 2013. “A Survey of Forecast Error Measures.”

Willmott, C. J., and K. Matsuura. 2005. “Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance.” *Climate Research* 30. doi: 10.3354/cr030079.