

# Package: naarma (via r-universe)

December 28, 2024

**Type** Package

**Title** Connect nanoarrow with (Rcpp)Armadillo

**Version** 0.0.1

**Date** 2024-12-17

**Author** Dirk Eddelbuettel

**Maintainer** Dirk Eddelbuettel <edd@debian.org>

**Description** The nanoarrow package offers C-level functionality to work with Arrow object, along with a small amount of C++ integration. This package uses it to interact with Armadillo objects. Some auxiliary testing facility from the nanoarrow package is included here as well.

**License** GPL (>= 2)

**Imports** Rcpp (>= 1.0.11), nanoarrow

**LinkingTo** Rcpp, RcppArmadillo, nanoarrow

**Suggests** tinytest, arrow

**RoxygenNote** 6.0.1

**Repository** <https://eddelbuettel.r-universe.dev>

**RemoteUrl** <https://github.com/eddelbuettel/naarma>

**RemoteRef** HEAD

**RemoteSha** 4ec729aa0f36dd93f453333ea69d994d9e02bb59

## Contents

armaMatrixExample . . . . .	2
armaVectorExample . . . . .	2
collectMatrixFromStream . . . . .	3

<b>Index</b>	<b>4</b>
--------------	----------

---

armaMatrixExample	<i>Given a (nano)arrow object via two pointers, and a column size, return an arma matrix</i>
-------------------	--

---

**Description**

Given a (nano)arrow object via two pointers, and a column size, return an arma matrix

**Usage**

```
armaMatrixExample(vec, ncol, verbose = FALSE)
```

**Arguments**

vec	A nanoarrow object
ncol	Number of columns in returned matrix
verbose	A logical value, default is false

**Limitations**

As this aims at **armadillo** operations, this functionality is limited to numeric vector columns. The framework used here could of course be extended to other Arrow formats.

**Examples**

```
sv <- nanoarrow::as_nanoarrow_array(c(4:8), nanoarrow::na_int16()) # internally int16
armaMatrixExample(sv, 2)
```

---

armaVectorExample	<i>Given a (nano)arrow array external pointer object via two pointers (with the second tucked away as the tag), return an arma vector</i>
-------------------	---

---

**Description**

Given a (nano)arrow array external pointer object via two pointers (with the second tucked away as the tag), return an arma vector

**Usage**

```
armaVectorExample(vec, verbose = FALSE)
```

**Arguments**

vec	A nanoarrow array object as an external pointer
verbose	A logical value, default is false

### Limitations

As this aims at **armadillo** interoperation, this functionality is limited to numeric vector columns. The framework used here could of course be extended to other Arrow formats.

### Examples

```
sv <- nanoarrow::as_nanoarrow_array(c(4:7), nanoarrow::na_int16()) # internally int16
armaVectorExample(sv)
```

---

collectMatrixFromStream

*Given a (nano)arrow stream object pointer, return an arma matrix*

---

### Description

Given a (nano)arrow stream object pointer, return an arma matrix

### Usage

```
collectMatrixFromStream(obj, verbose = FALSE)
```

### Arguments

obj	A nanoarrow array stream object via an external pointer
verbose	A logical value, default is false

### Limitations

As this aims at **armadillo** operation, the returned matrix is always double while supported many possible column types.

# Index

`armaMatrixExample`, [2](#)

`armaVectorExample`, [2](#)

`collectMatrixFromStream`, [3](#)