HAL
open science

INTAROS – IMR Dataset

Didier Renard, Fabien Ors

To cite this version:

Didier Renard, Fabien Ors. INTAROS – IMR Dataset. [Research Report] MINES ParisTech - PSL Research University. 2018. hal-01820877

HAL Id: hal-01820877

https://minesparis-psl.hal.science/hal-01820877v1

Submitted on 22 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INTAROS – IMR Dataset

Didier Renard, Fabien Ors

Mai 2018

N° R180502FORS

MINES ParisTech - CENTRE DE GÉOSCIENCES
35, Rue Saint Honoré – 77305 Fontainebleau - France
Secrétariat : +33 (0)1 64 69 47 78, Fax : +33 (0)1 64 69 47 05 - http://www.geosciences.mines-paristech.fr

MINES ParisTech - Centre de Géosciences

Equipe Géostatistique

35, rue Saint Honoré

77300 Fontainebleau, France

Tél. 01 64 69 47 81

Fax 01 64 69 47 05

Didier Renard, Fabien Ors

INTAROS – IMR Dataset

Equipe	Géostatistique
Visa	J. Rivoirard

Intaros - IMR Dataset

D. Renard, F. Ors

May 2nd 2018

Introduction

This paper is meant to demonstrate how to use a simple **Kriging interpolation** from the **RGeostats** package applied to the **Annual CTD datasets** from R/V Håkon Mosby (Norwegian research vessel).

Download of NetCDF files

Use the following **bash** script for downloading the data files in the current directory:

```
lftp -c 'open -e "mget 58AA_CTD_2002.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2003.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2004.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2005.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2006.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2007.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2008.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2009.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2010.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2011.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2012.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2013.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2014.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2015.nc" ftp.nmdc.no/nmdc/IMR/CTD'
lftp -c 'open -e "mget 58AA_CTD_2016.nc" ftp.nmdc.no/nmdc/IMR/CTD'
```

Definition of R functions

Definition of some functions:

- **load_data**: Load the data set (all NetCDF files (*.nc) from the given directory) into a *Db* RGeostats database. The data files must contain the following variables:
  - LATITUDE: Latitude (degrees) of the vessel position. 1D size = {Nb Positions}
  - LONGITUDE: Longitude (degrees) of the vessel position. 1D size = {Nb Positions}
  - TIME: Time of the position (number of days since January 1st, 1950). 1D size = {Nb Positions}
  - DEPH: Depth of each measure (m). 2D size = {Nb Positions, Nb Measures}
  - TEMP: Temperature (°C). 2D size = {Nb Positions, Nb Measures}
  - CNDC: Electrical conductivity (S m-1). 2D size = {Nb Positions, Nb Measures}
  - PSAL: Sea water practical salinity (0.001). 2D size = {Nb Positions, Nb Measures}
- **time2date**: Convert a time value from the dataset into a readable date (YYYY-MM-DD).
- **date2time**: Convert a readable date (YYYY-MM-DD) into a time value used by the dataset.
- **som**: Return the first day of the month from a given date

- **eom**: Return the last day of the month from a given date
- **apply_sel**: Apply a selection on the *Db* by masking samples falling outside from a given interval (Depth, Longitude, Latitude, Dates, Variable values). Accepted variable names are: *Temperature*, *Conductivity* and *Salinity*.
- **remove_sel**: Remove all selection variables from the *Db*.
- **get_xlim**: Get integer Longitude limits of the given [filtered] *Db*
- **get_ylim**: Get integer Latitude limits of the given [filtered] *Db*
- **get_tlim**: Get time limits of the given [filtered] *Db*
- **get_zlim**: Get range values for a given variable of the [filtered] *Db*
- **get_title**: Return a suffix for plot titles which indicates the period covered by a [filtered] *Db*
- **create_grid**: Create a grid centered on the data from the given [filtered] *Db* with a given square mesh
- **display_grunit**: Display a grid centered on the data from the [filtered] *Db* with a given square mesh into the current plot
- **display_var**: Display the variable of interest from the given [filtered] *Db*
- **display_stat**: Display the statistics map coming from the function **stats_grid**
- **display_result**: Display the estimated results coming from the function **interpol_var** and overlay data in the same plot
- **interpol_var**: Interpolate a variable on a 2D grid at a given depth from a [filtered] *Db*
- **stats_grid**: Generate a statistics map for a given variable from a [filtered] *Db*

Quick example

- Here is a quick R script using these functions which performs:
  - Some simple display of available data
  - Some data selection and dates manipulation
  - An analysis of the correlation between Temperature and Conductivity for a given year
  - The display of data points density in each grid cell for a given month
  - Trimester interpolations of the Temperature data at the surface for a given year

Note that subsequent interpolation is performed in 2D (considering the 3D information would require a different script).

Loading Data

First of all, we load the data from the current directory into a new RGeostats database *Db*. We check its contents and the number of active samples.

```
db = load_data(dir = ".")
```

```
## Reading 58AA_CTD_2002.nc : Number of positions = 54 , Maximum depth = 3433 m)
## Reading 58AA_CTD_2003.nc : Number of positions = 693 , Maximum depth = 2810 m)
## Reading 58AA_CTD_2004.nc : Number of positions = 1213 , Maximum depth = 2182 m)
## Reading 58AA_CTD_2005.nc : Number of positions = 1191 , Maximum depth = 2703 m)
## Reading 58AA_CTD_2006.nc : Number of positions = 1164 , Maximum depth = 3298 m)
## Reading 58AA_CTD_2007.nc : Number of positions = 1410 , Maximum depth = 3211 m)
## Reading 58AA_CTD_2008.nc : Number of positions = 1205 , Maximum depth = 2935 m)
```

```
## Reading 58AA_CTD_2009.nc : Number of positions = 960 , Maximum depth = 3011 m)
## Reading 58AA_CTD_2010.nc : Number of positions = 1413 , Maximum depth = 3390 m)
## Reading 58AA_CTD_2011.nc : Number of positions = 985 , Maximum depth = 3233 m)
## Reading 58AA_CTD_2012.nc : Number of positions = 1114 , Maximum depth = 3058 m)
## Reading 58AA_CTD_2013.nc : Number of positions = 1102 , Maximum depth = 2990 m)
## Reading 58AA_CTD_2014.nc : Number of positions = 966 , Maximum depth = 1714 m)
## Reading 58AA_CTD_2015.nc : Number of positions = 764 , Maximum depth = 3055 m)
## Reading 58AA_CTD_2016.nc : Number of positions = 559 , Maximum depth = 3654 m)
```

The following paragraph shows the contents of the newly created Db

```
db
```

```
##
## Data Base Characteristics
## ========================
##
## Data Base Summary
## -----------------
## File is organized as a set of isolated points
## Space dimension              = 3
## Number of fields             = 8
## Maximum Number of attributes = 8
## Total number of samples      = 5080116
##
## Variables
## ---------
## Field =   1 - Name    =  rank - Locator =  rank
## Field =   2 - Name    =  Longitude - Locator =  x1
## Field =   3 - Name    =  Latitude - Locator =  x2
## Field =   4 - Name    =  Depth - Locator =  x3
## Field =   5 - Name    =  Temperature - Locator =  NA
## Field =   6 - Name    =  Conductivity - Locator =  NA
## Field =   7 - Name    =  Salinity - Locator =  NA
## Field =   8 - Name    =  Time - Locator =  NA
```

Manipulate the dates:

```
dlim = time2date(get_tlim(db))
# Calculate first day of each month of the dataset
smon = seq(from = som(dlim[1]), to = eom(dlim[2]), by="1 month")
# Calculate last day of each month of the dataset
emon = do.call("c", lapply(smon,eom))
cat(get_title("The dataset period is:",dlim))
```

```
## The dataset period is: (2002-03-07 => 2016-10-16)
```

Exploratory data Analysis

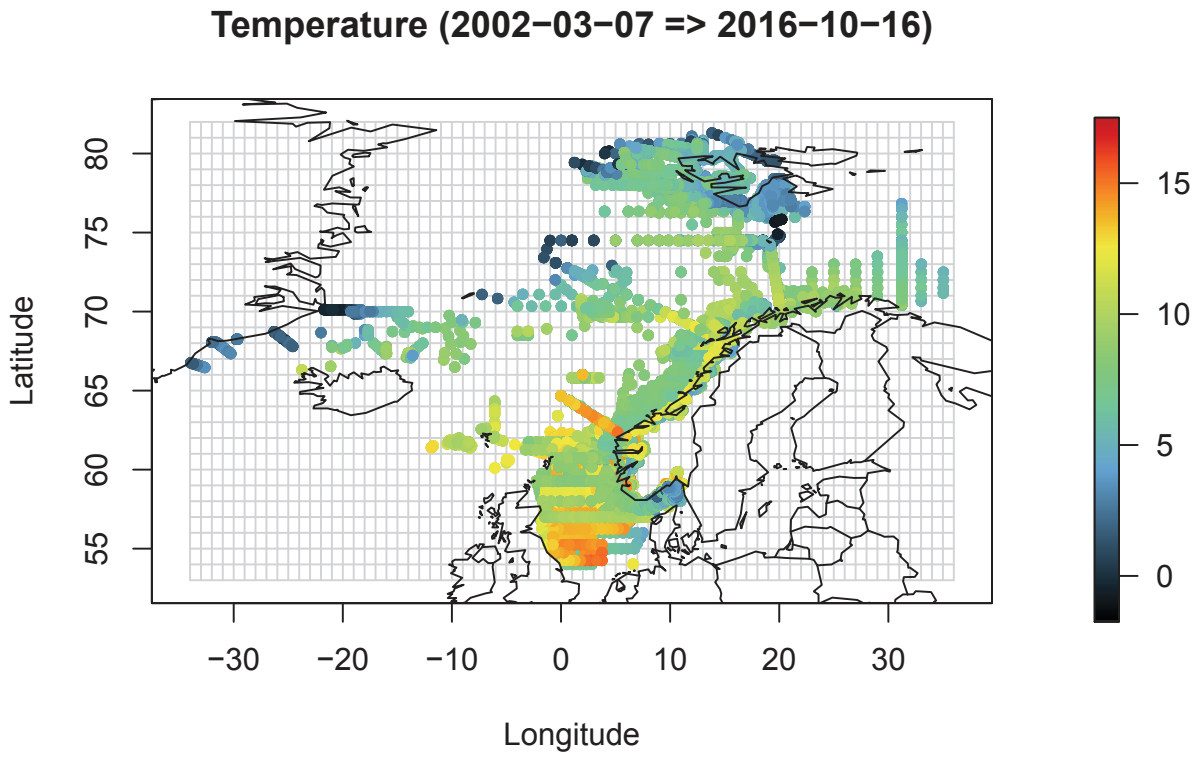
We apply a selection to mask all samples deeper than 10m depth:

```
db = remove_sel(db)
db = apply_sel(db, depth_lim = c(0., 10))
```

```
## New number of active samples =  117756 ( out of 5080116 )
```

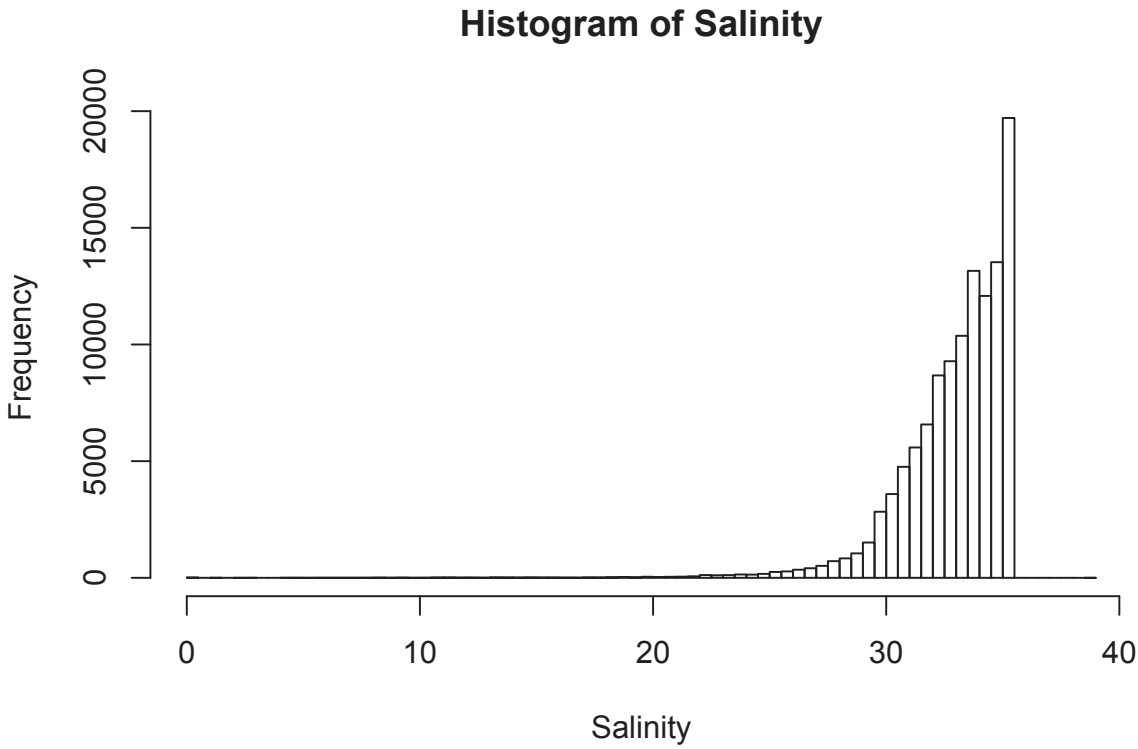
In a 2D aerial view, we display the previously selected *Temperature* values and the countries borders:

```r
display_var(db, "Temperature")
```



Then, we inquire the database for the range of *Salimity* values:

```r
Salinity = db.extract(db, "Salinity", flag.compress = TRUE)
hist(Salinity, breaks = 100, main = "Histogram of Salinity")
```




```
cat("Range of values = ", range(Salinity, na.rm = TRUE), "\n")
```

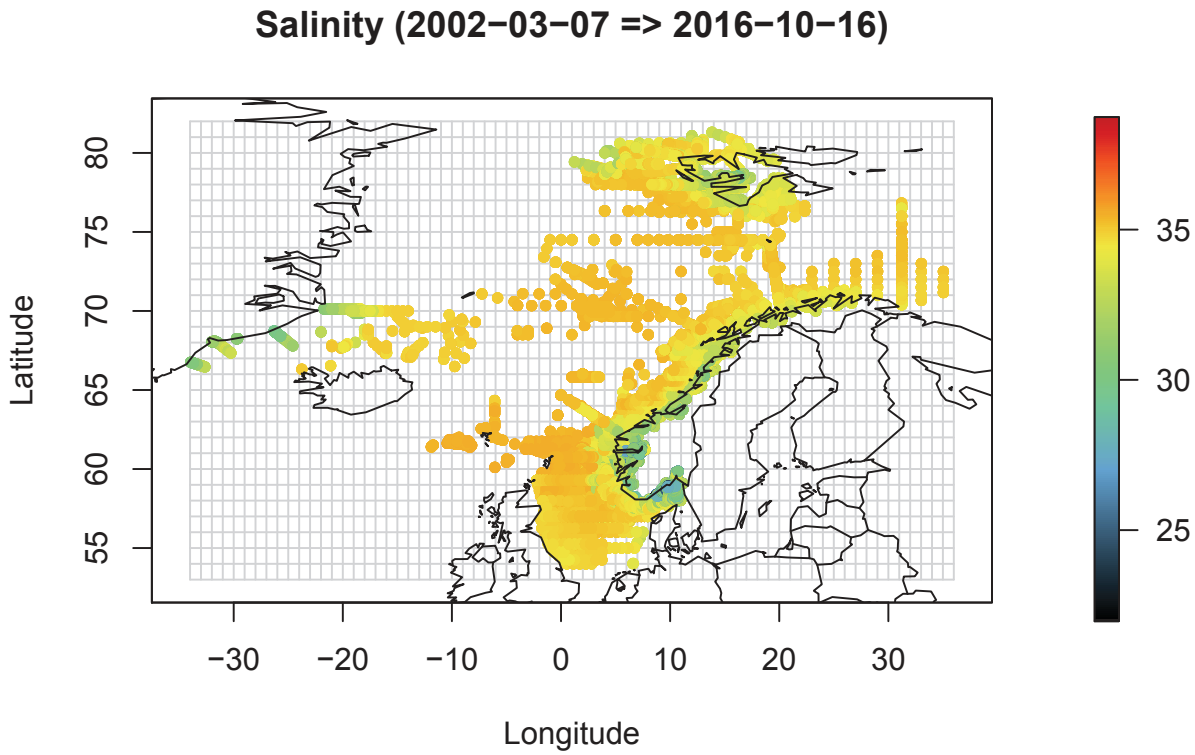
```
## Range of values =  0.1694 38.7109
```

We combine the previous selection with a new selection for removing *Salinity* outliers:

```
db = apply_sel(db, z_lim = c(22, 39), z_var = "Salinity")
```

```
## New number of active samples =  116836 ( out of 5080116 )
```

```
display_var(db, "Salinity")
```

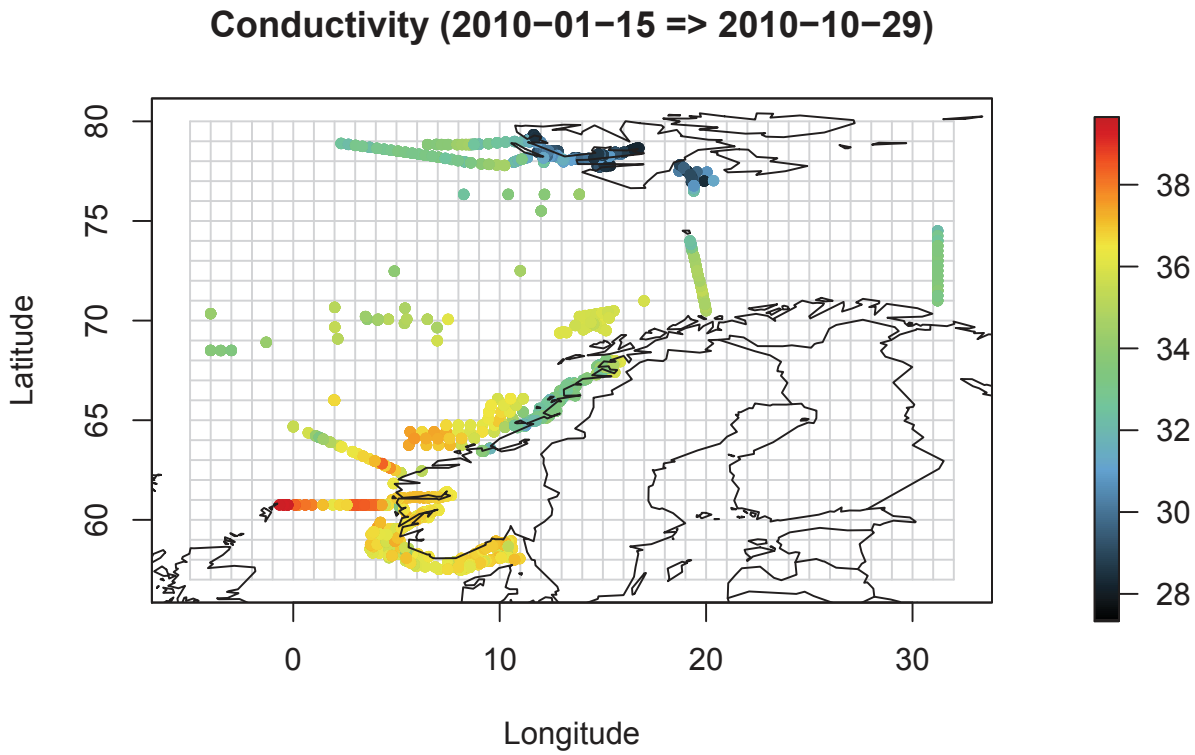


We display *Conductivity* for 2010's measures at a depth around 100m (previous selection is cancelled first):

```
datelim = c("2010-01-01", "2010-12-31")
db      = remove_sel(db)
db      = apply_sel(db, depth_lim = c(95., 105),
                    dates_lim = datelim)
```

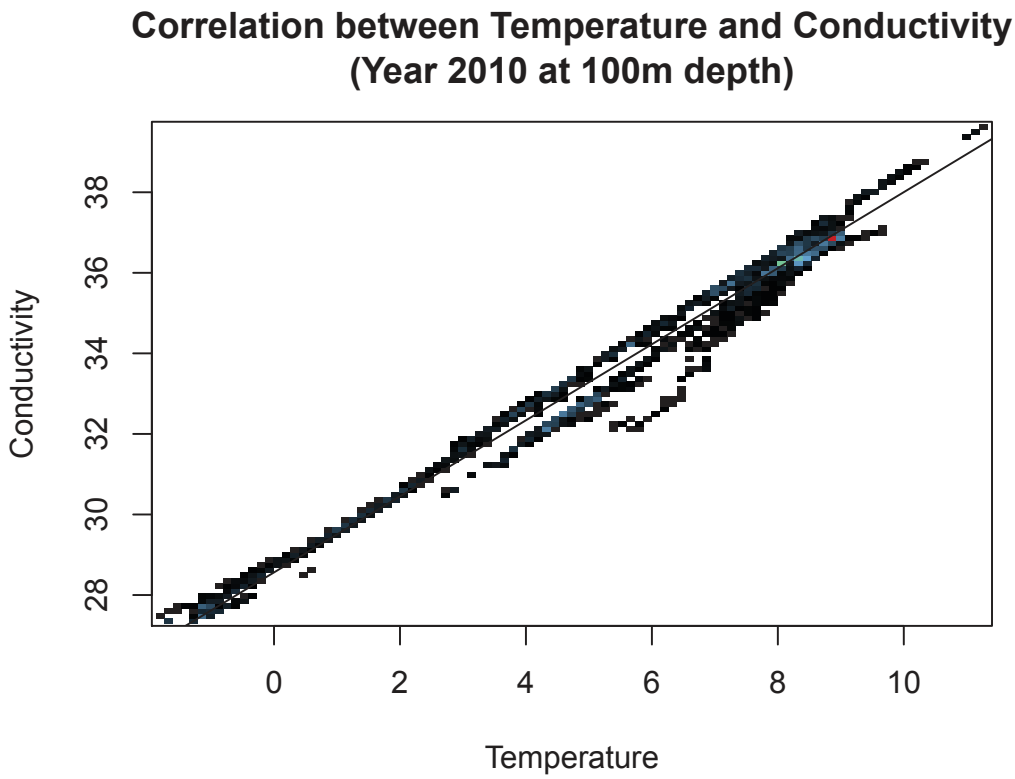
```
## New number of active samples =  9454 ( out of 5080116 )
```

```
display_var(db, "Conductivity")
```



We show the correlation between *Conductivity* and *Temperature*. As a result, we also obtain the correlation coefficient.

```
title  = "Correlation between Temperature and Conductivity\n(Year 2010 at 100m depth)"
corval = correlation(db, "Temperature", "Conductivity", flag.regr = TRUE, title = title)
```




```
cat("Correlation Coefficient = ", corval, "\n")
```

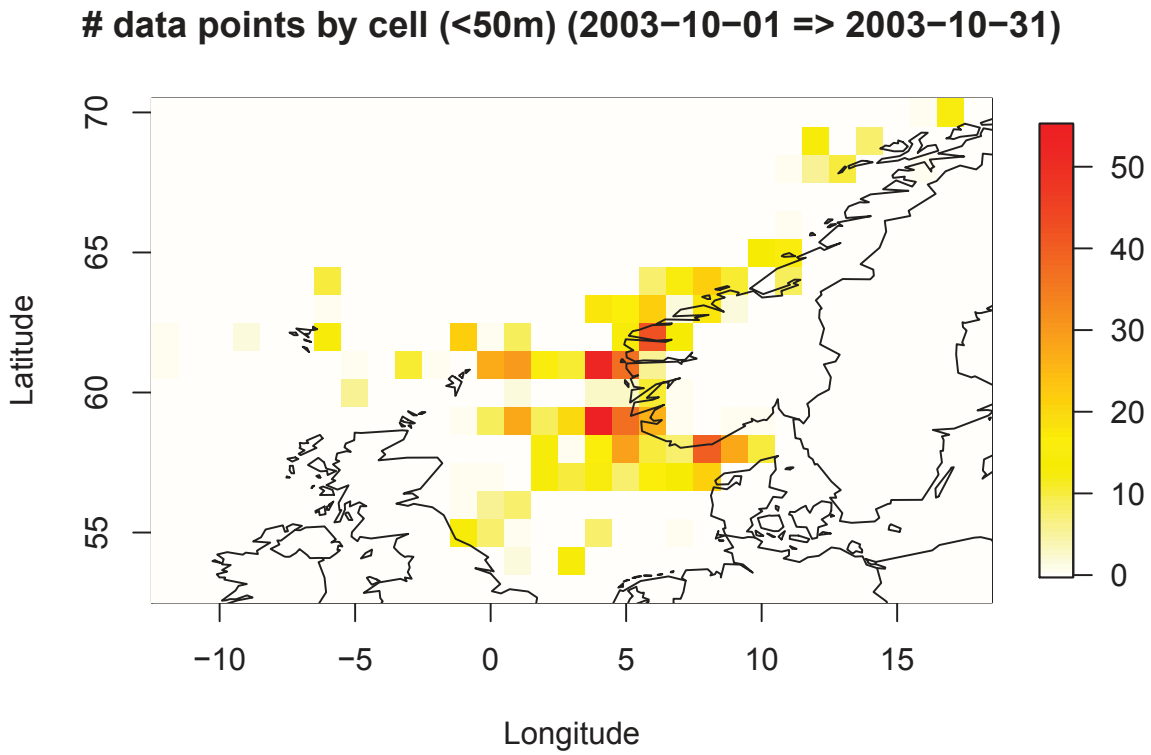
```
## Correlation Coefficient =  0.9936022
```

We display the number of data points (<50m depth) that fall in each grid cell (with a mesh equal to 1 degree) for a given month:

```
datelim = c(smon[20], emon[20])
db      = remove_sel(db)
db      = apply_sel(db, depth_lim = c(0., 50), dates_lim = datelim)
```

```
## New number of active samples =  1082 ( out of 5080116 )
```

```
dbstat  = stats_grid(db, "Temperature", fun = "count")
title   = get_title("# data points by cell (<50m)", datelim)
display_stat(dbstat, title = title)
```



Estimation

We estimate the 2007 *Temperature* season maps at sea surface (1m depth +-1m) on a refined grid (mesh = 0.5°) which covers all data available. The geostatistical model is automatically fitted using default parmaters: It could be improved using more expertise and providing additional parameters to function **interpol_var**. For comparison purpose, the maps are displayed using the same colorscale, latitude and longitude limits.

```
year     = 2007
var      = "Temperature"
mesh     = 0.5
depth    = 1
depthtol = 1
datelim  = c(paste(year, "-01-01", sep = ""), paste(year, "-12-31", sep = ""))
db       = remove_sel(db)
db       = apply_sel(db, dates_lim = datelim)
```

```
## New number of active samples =  546718 ( out of 5080116 )
```

```
xlim     = get_xlim(db)
ylim     = get_ylim(db)
zlim     = get_zlim(db, var = var)
cat("Range of Temperatures for", year, "=", zlim, "\n")
```

```
## Range of Temperatures for 2007 = -1.8927 13.7984
```

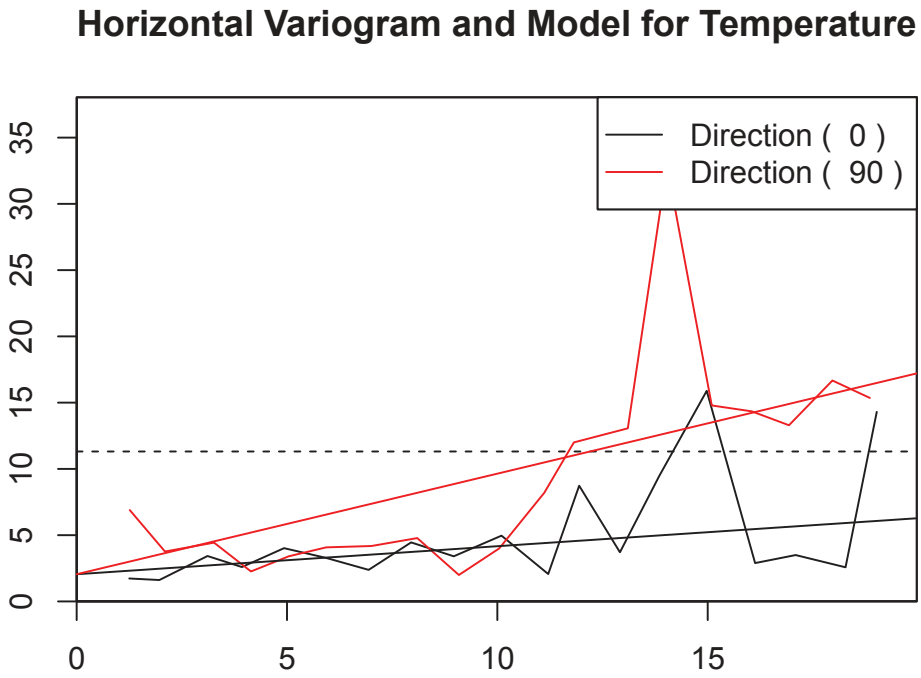
First three months of the year

```r
datelim  = c(paste(year, "-01-01", sep = ""), paste(year, "-03-31", sep = ""))
db       = remove_sel(db)
db       = apply_sel(db, dates_lim = datelim, verbose=FALSE)
cat("Temperature in First trimester of", year, "\n")
```

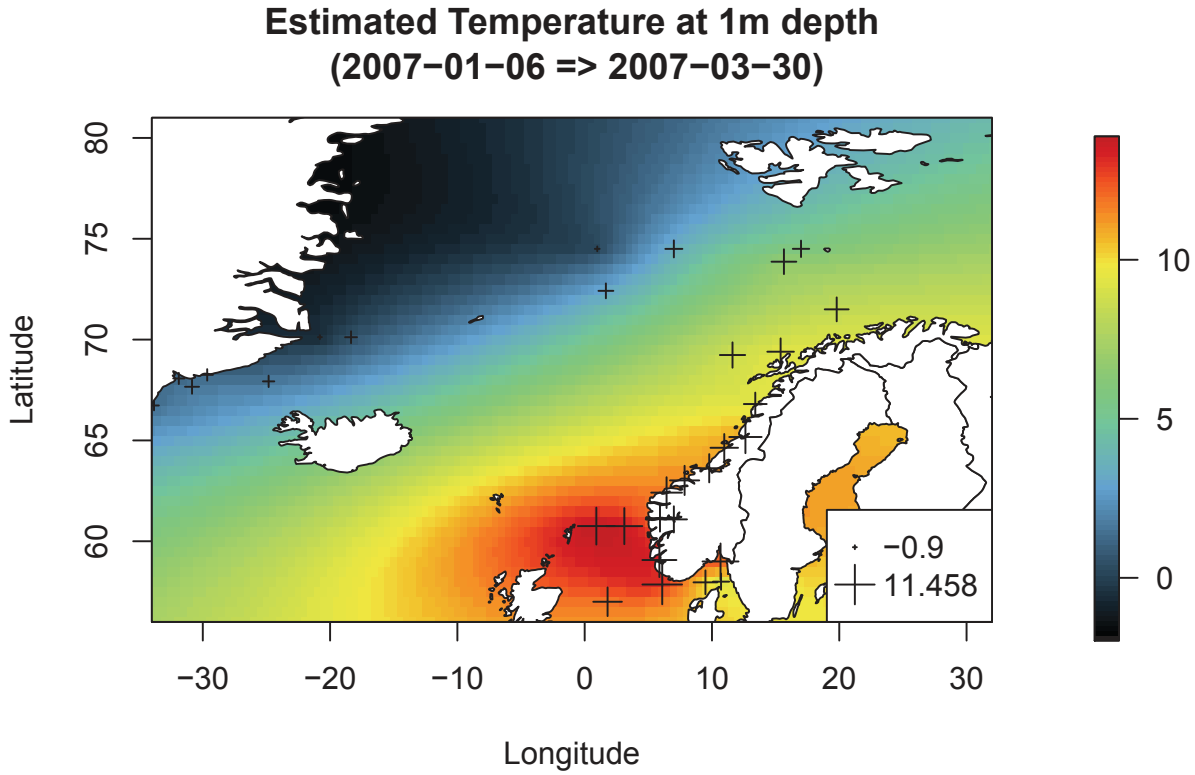
```
## Temperature in First trimester of 2007
```

```r
res      = interpol_var(db, depth = depth, depth_tol = depthtol,
                        var = var, mesh = mesh)
```

```
## New number of active samples =  142 ( out of 5080116 )
## Range of Information = -1.8725 13.7984
```




```r
display_result(res = res, flag.estim = TRUE,
               xlim = xlim, ylim = ylim, zlim = zlim)
```



Second trimester of the year

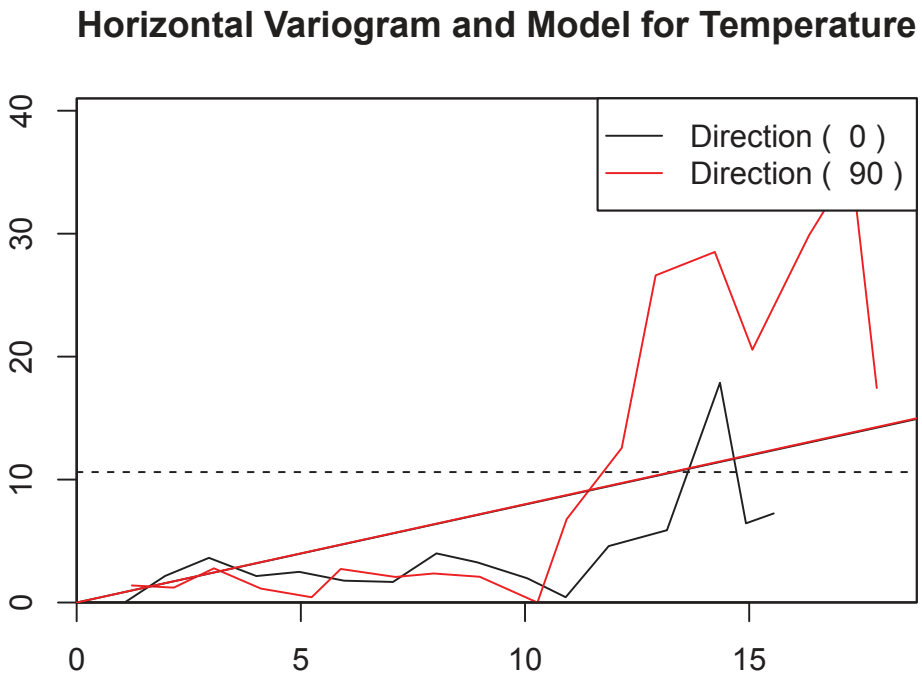
```
datelim  = c(paste(year, "-04-01", sep = ""), paste(year, "-06-30", sep = ""))
db       = remove_sel(db)
db       = apply_sel(db, dates_lim = datelim, verbose=FALSE)

cat("Temperature in Second trimester of", year, "\n")
```

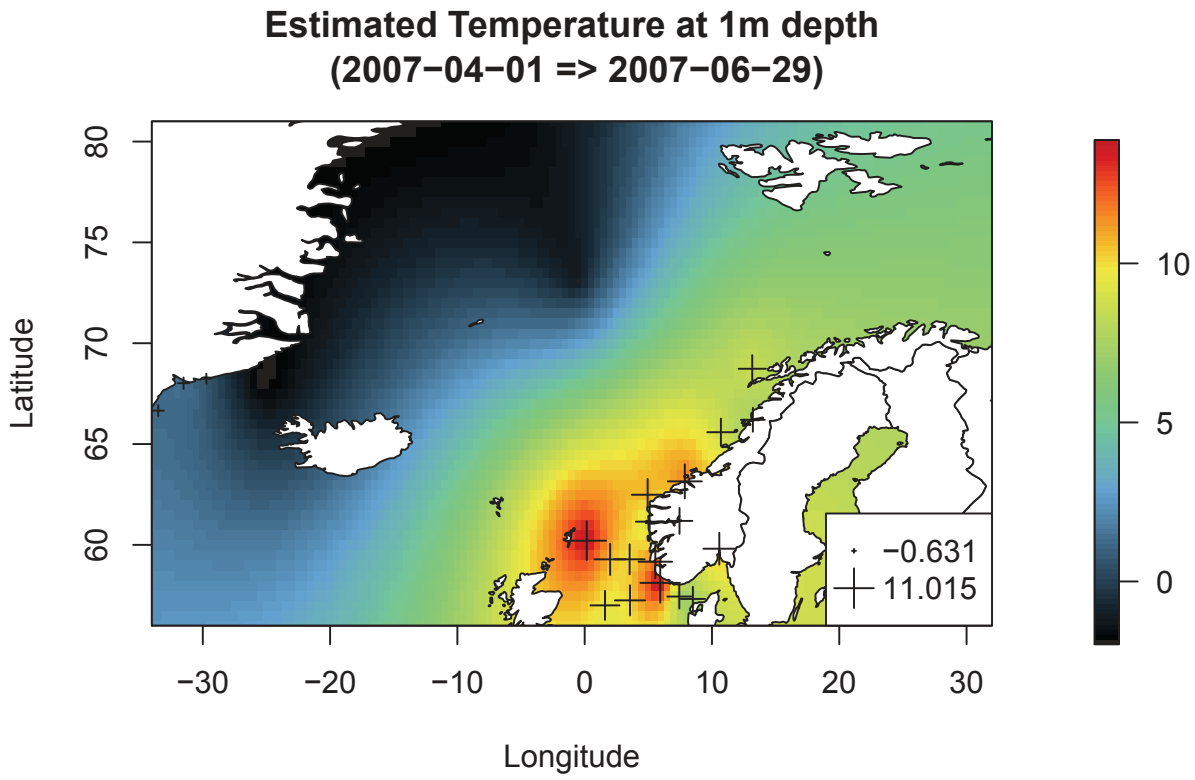
```
## Temperature in Second trimester of 2007
```

```
res      = interpol_var(db, depth = depth, depth_tol = depthtol,
                        var = var, mesh = mesh)
```

```
## New number of active samples =  53 ( out of 5080116 )
## Range of Information = -1.8927 13.4265
```




```
display_result(res = res, flag.estim = TRUE,
               xlim = xlim, ylim = ylim, zlim = zlim)
```



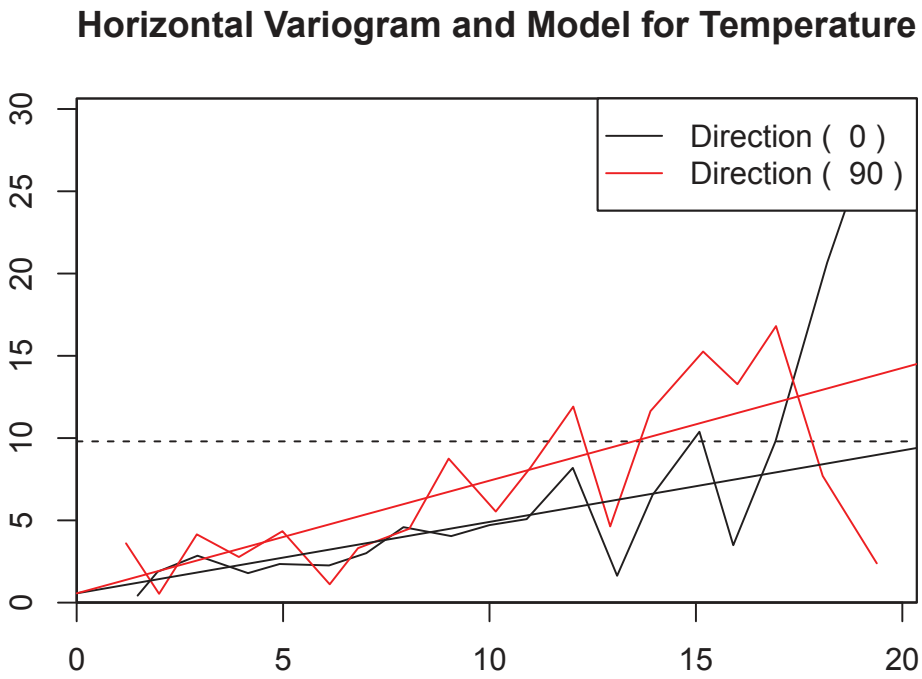
Third trimester of the year

```
datelim  = c(paste(year, "-07-01", sep = ""), paste(year, "-09-30", sep = ""))
db       = remove_sel(db)
db       = apply_sel(db, dates_lim = datelim, verbose=FALSE)
cat("Temperature in Third trimester of", year, "\n")
```

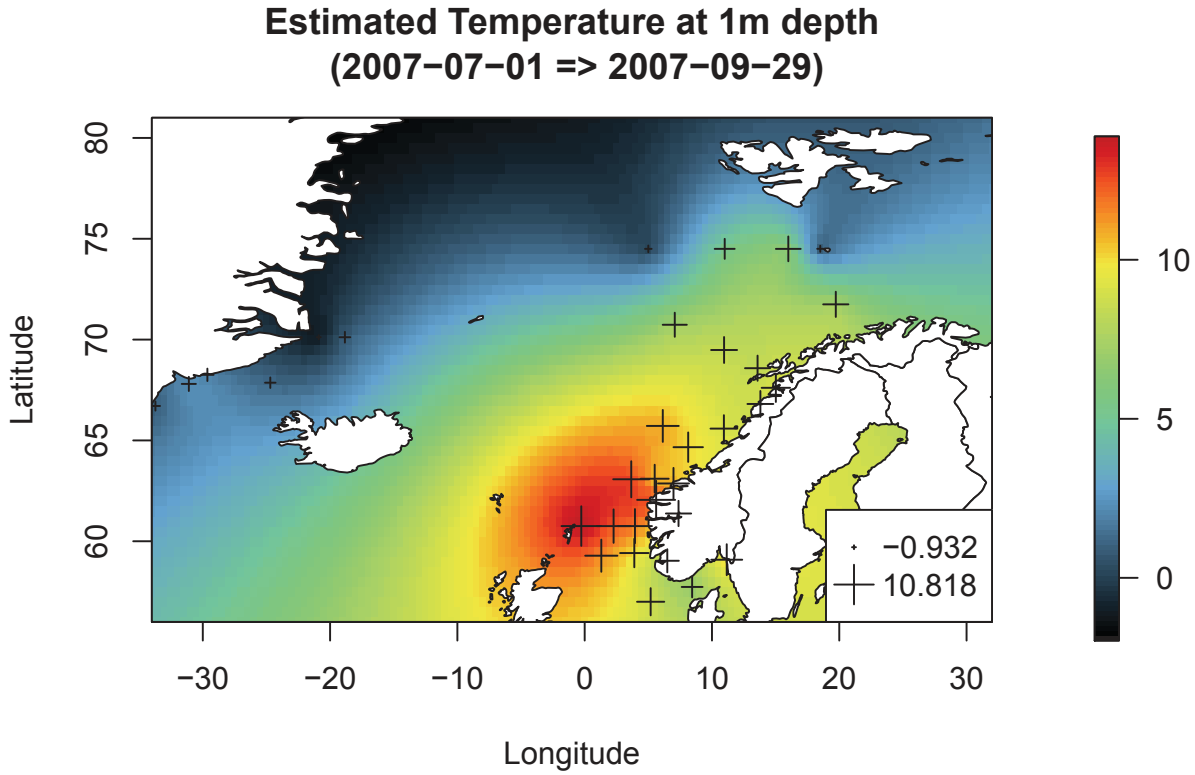
```
## Temperature in Third trimester of 2007
```

```
res      = interpol_var(db, depth = depth, depth_tol = depthtol,
                        var = var, mesh = mesh)
```

```
## New number of active samples =  186 ( out of 5080116 )
## Range of Information = -1.8803 12.8682
```




```
display_result(res = res, flag.estim = TRUE,
               xlim = xlim, ylim = ylim, zlim = zlim)
```



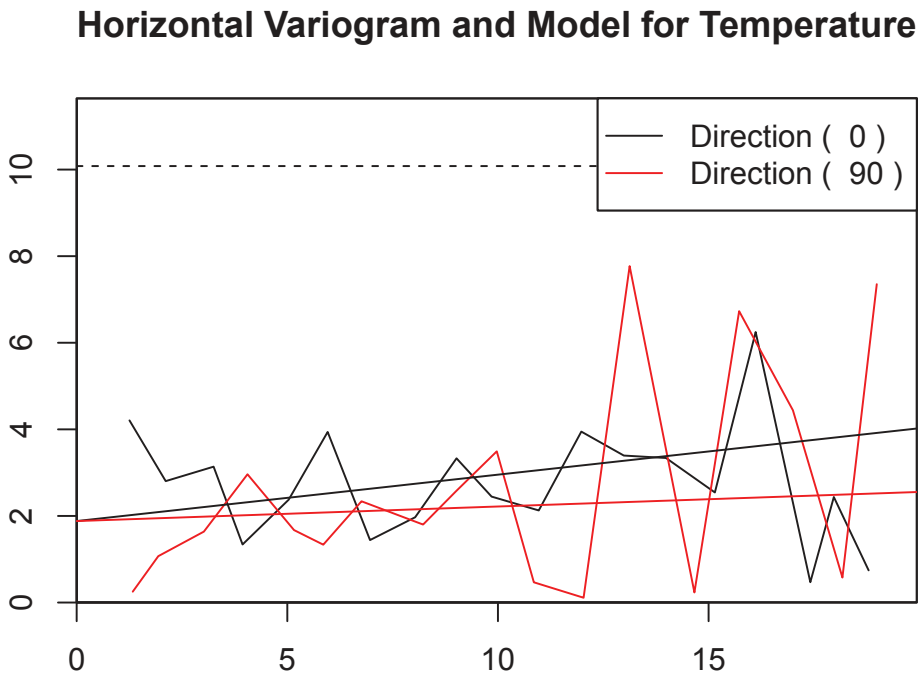
Fourth trimester of the year

```
datelim  = c(paste(year, "-10-01", sep = ""), paste(year, "-12-31", sep = ""))
db       = remove_sel(db)
db       = apply_sel(db, dates_lim = datelim, verbose=FALSE)
cat("Temperature in Fourth trimester of", year, "\n")
```

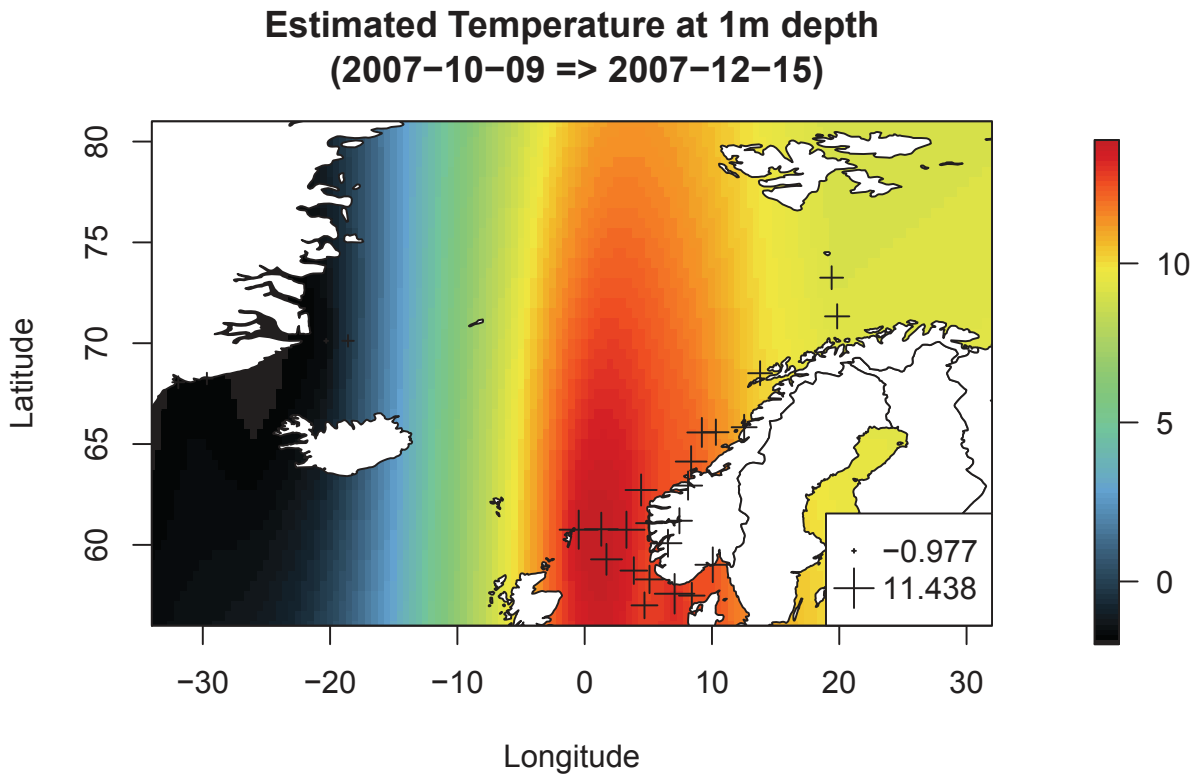
```
## Temperature in Fourth trimester of 2007
```

```
res      = interpol_var(db, depth = depth, depth_tol = depthtol,
                        var = var, mesh = mesh)
```

```
## New number of active samples =  69 ( out of 5080116 )
## Range of Information = -1.8804 12.7919
```

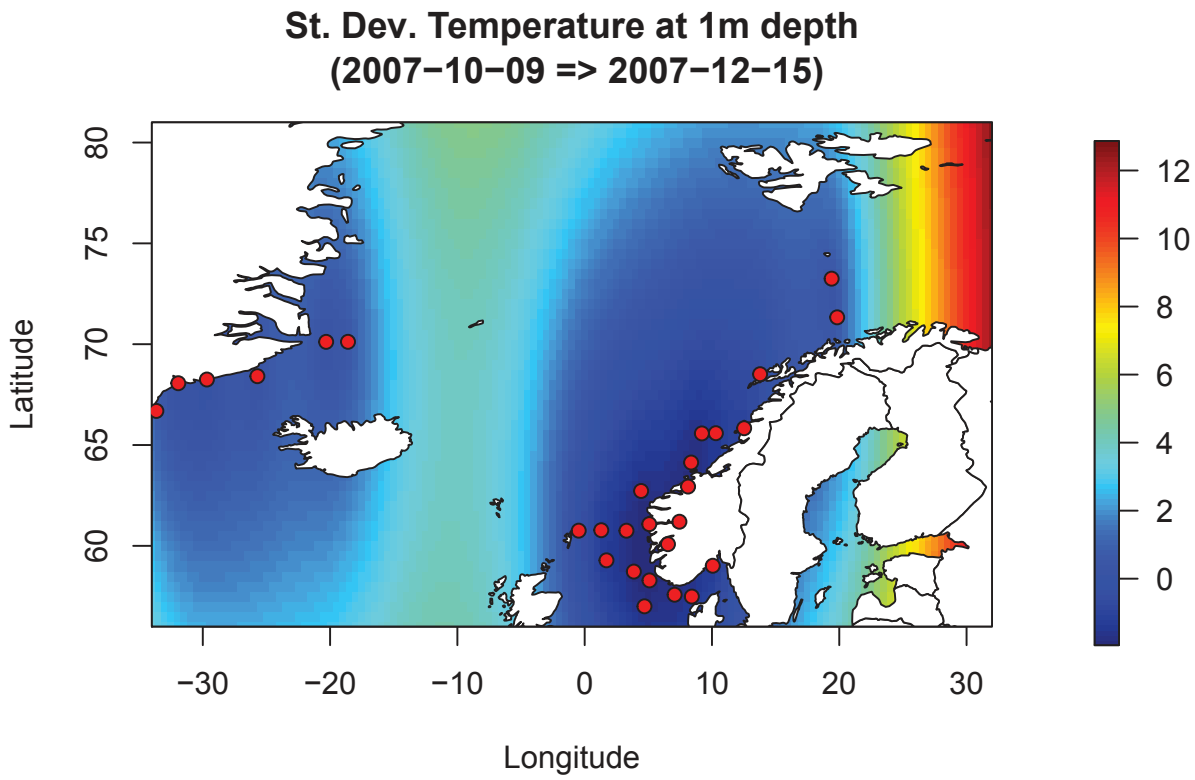



```
display_result(res = res, flag.estim = TRUE,
               xlim = xlim, ylim = ylim, zlim = zlim)
```



Finally, we display the *Standard Deviation of the Estimation Error* for the previous last trimester interpolation (could be done after each interpolation and would give different results).

```r
display_result(res = res, flag.estim = FALSE)
```



Perspectives

- Several perspectives can be imagined for this showcase script:
  - Tackling the data in a 3D approach in order to account for any possible vertical trend. Specific display would be welcomed.
  - Using a secondary exhaustive information (i.e. satellite image) in order to improve the knowledge provided by the measures: external drift concept.
  - Using simulations (instead of kriging) in order to derive maps of quantities which are not linearly related to the data: Probability that the temperature passes a given limit