



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

TerraST

UM PACOTE PARA INTEGRAÇÃO DE ESTRUTURAS DE DADOS ESPAÇO-TEMPORAIS TERRALIB AO AMBIENTE R

Daniel Teixeira dos Santos

Introdução ao Geoprocessamento – SER300

Sumário

- Motivação
- Dados espaço-temporais
- TerraLib 5
- Ambiente R
- Pacotes auxiliares
- TerraST

Motivação

- Análise de dados espaço-temporais usualmente é feita primeiro sob um aspecto espacial ou sob um aspecto temporal, mas raramente sob esses dois aspectos de forma integrada.
- Falta software capaz de manipular, importar, exportar, exhibir e analisar esse tipo de dado.

Motivação

O ambiente estatístico R fornece uma linguagem amigável aliada a uma grande quantidade de pacotes e funções para estatística espacial. Lhe falta no entanto os recursos típicos presentes em Sistemas de Informação Geográficos como recuperação e armazenamento de dados espaço temporais.

Observações

tipo Observação [F: Tipo; C: Tipo; M: Tipo]

operações:

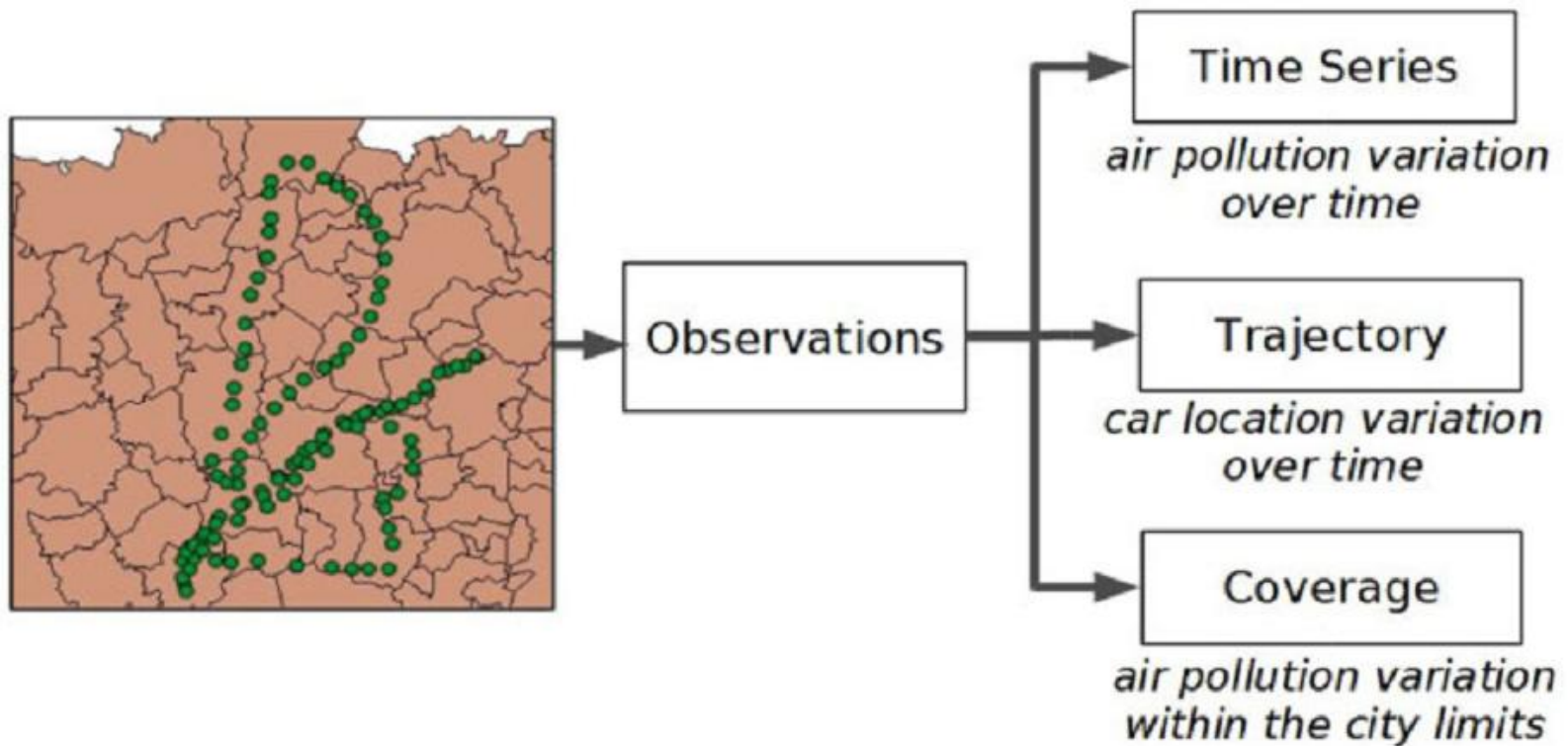
novo: $\{(F,C,M)_1, (F,C,M)_2, \dots, (F,C,M)_n\} \rightarrow \text{Observações} \mid n > 0$

referência: $\text{Observações} \rightarrow F$

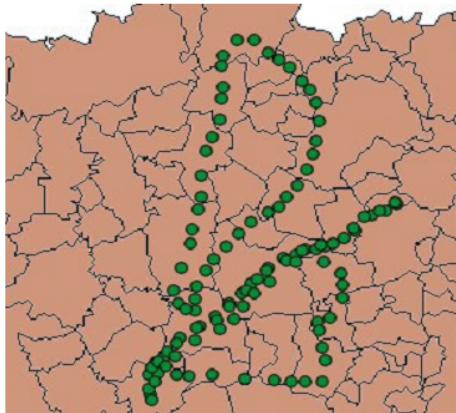
posições: $\text{Observações} \rightarrow \{C_1, C_2, \dots, C_n\}$

medições: $\text{Observações} \times C \rightarrow M$

Diferentes visualizações sobre as observações



Diferentes visualizações sobre as observações



a set of cars
equipped with GPS
and air pollution
sensors



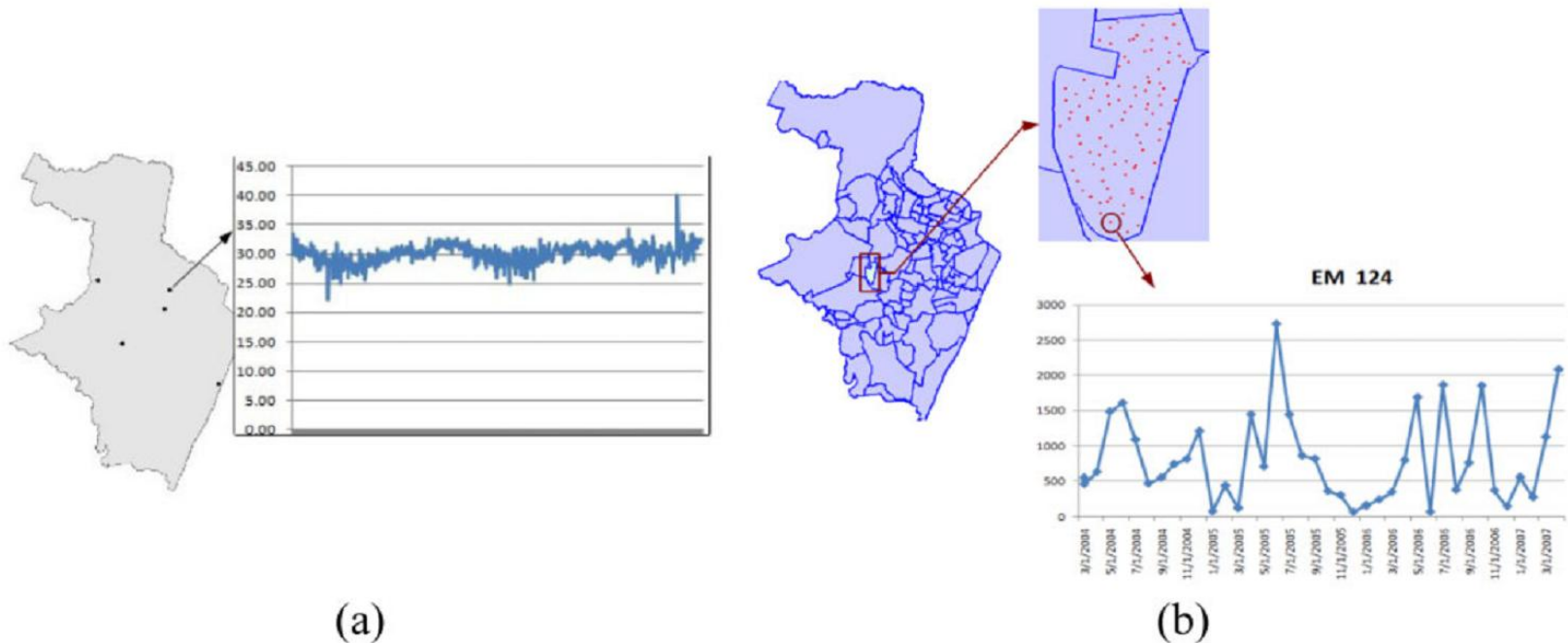
each observation
contains a car
identity, a time
instant, a location
and an air
pollution value

(1) *"When the average pollution in the city was greater than x for more than five hours?"*

(2) *"How long did car c01 stay in the south region of the city?"*

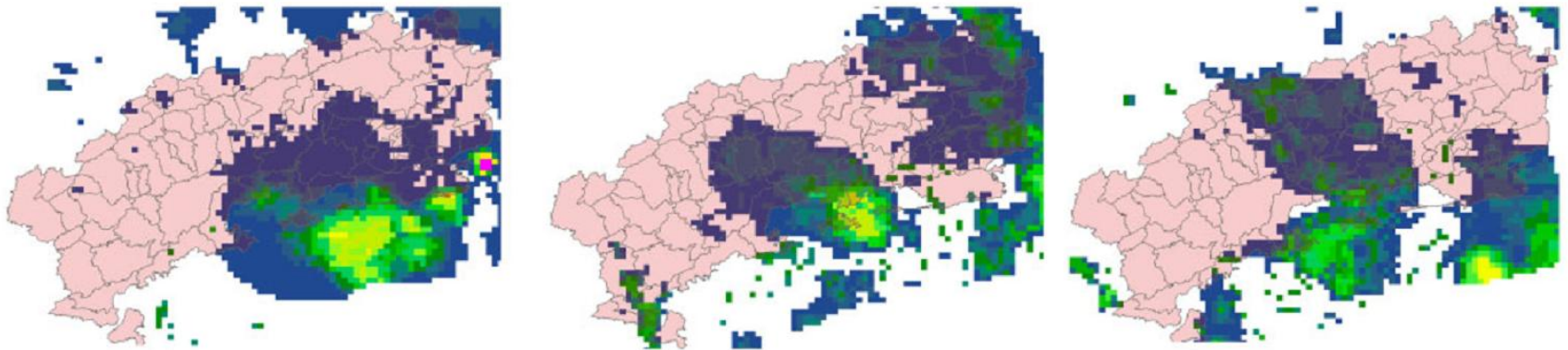
(3) *"What city district had the worst pollution index in this day?"*

Séries temporais



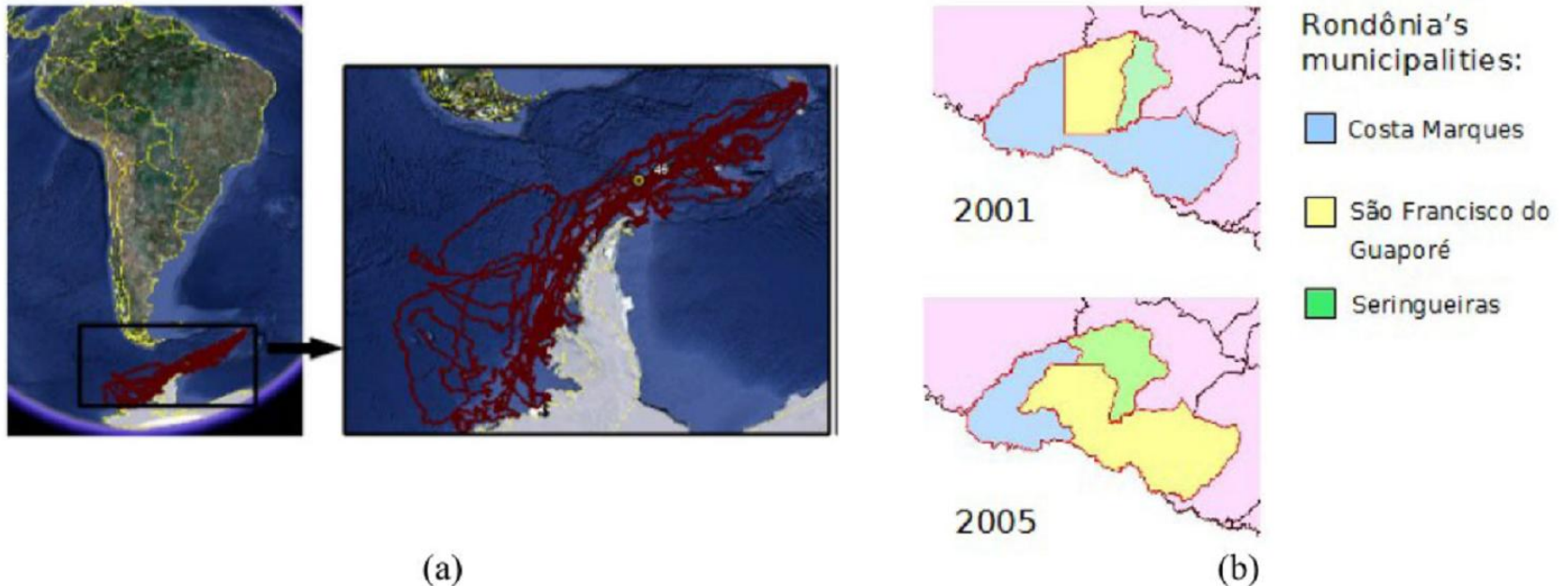
- (a) Temperatura coletada por diferentes estações metereológicas;
(b) Número de ovos coletados de uma armadilha de mosquito em um distrito de Recife.

Coberturas



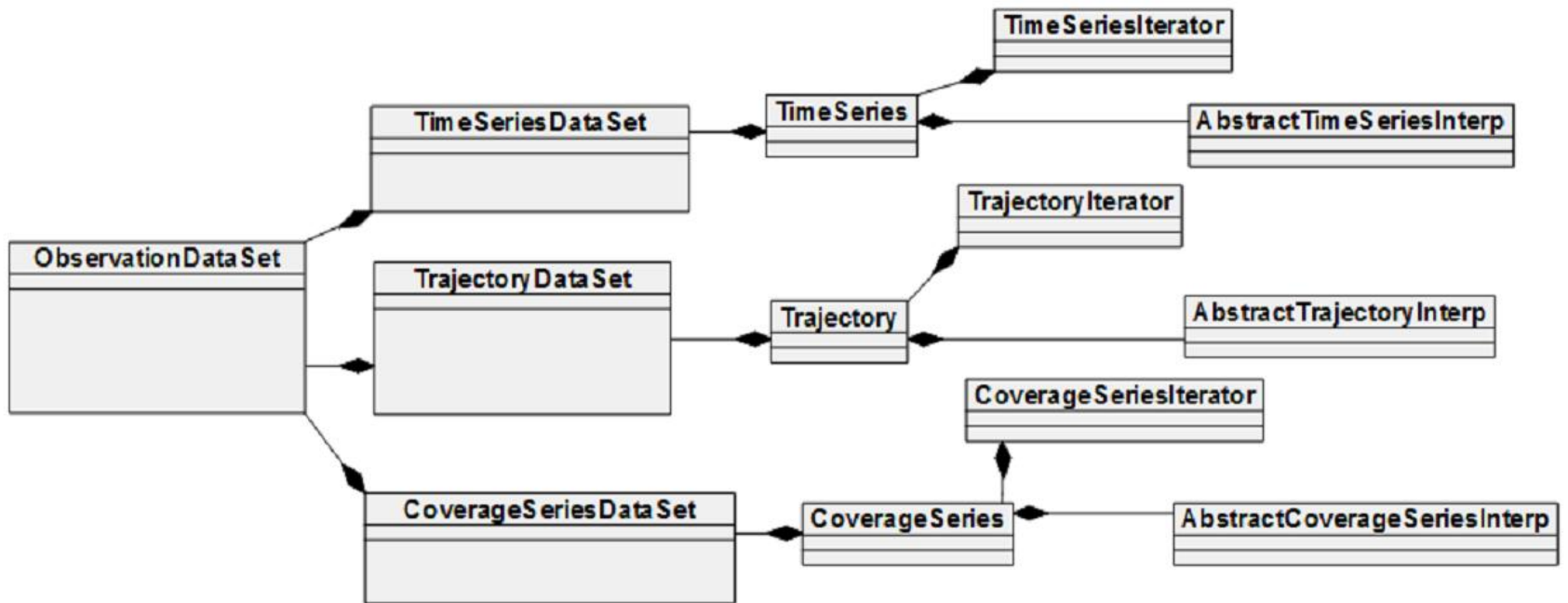
Chuva no estado do Rio de Janeiro em
11 de Janeiro de 2011

Trajетórias

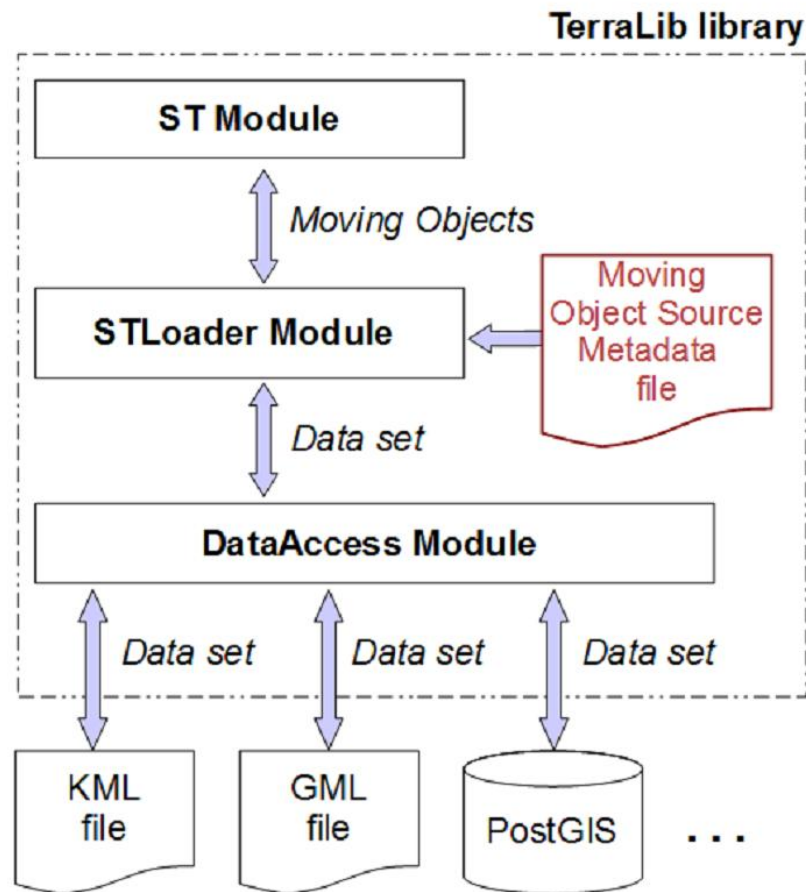


- (a) Trajetória de leões-marinhos na Antártida
- (b) Evolução dos limites de três municípios de Rondônia de 2001 a 2005

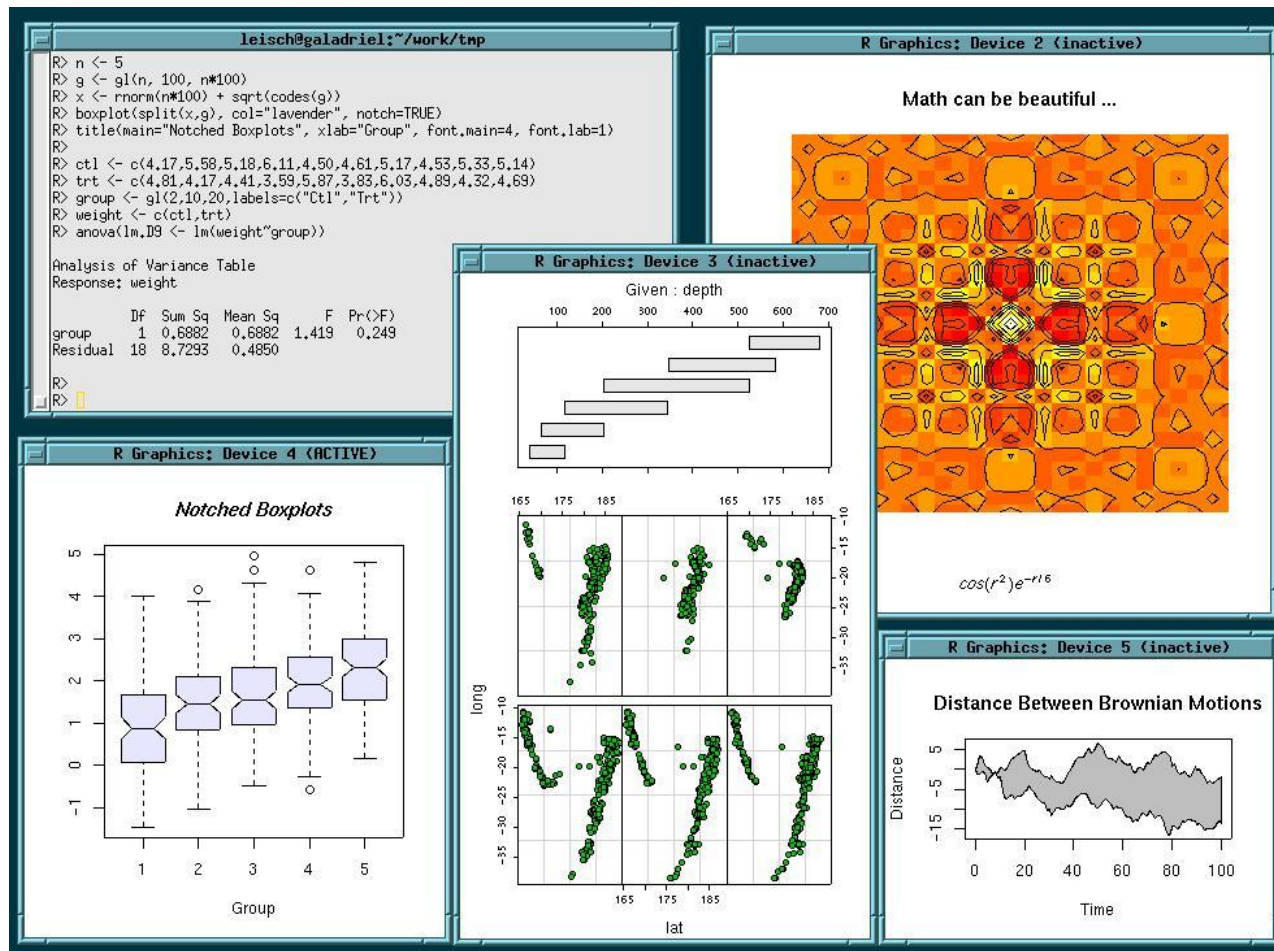
TerraLib 5



TerraLib



Ambiente R

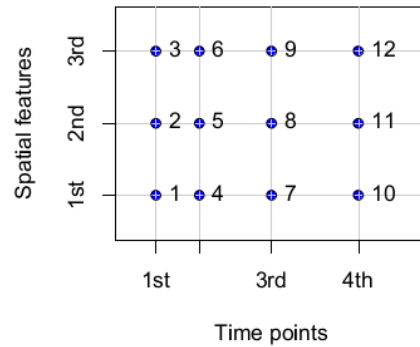


Spacetime

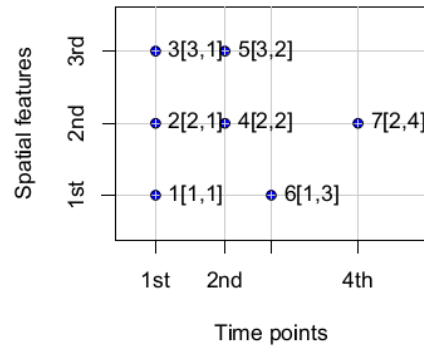
- Desenvolvido por Edzer Pebesma
- Unificação das estruturas de dados espaço temporais
- Operações de manipulação, exibição e análise

Spacetime

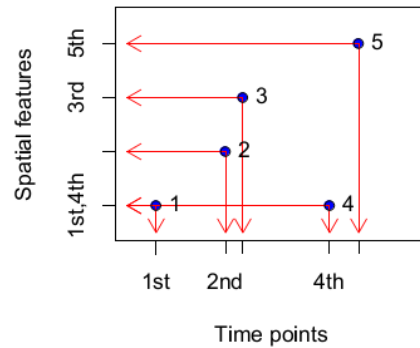
STF: full grid layout



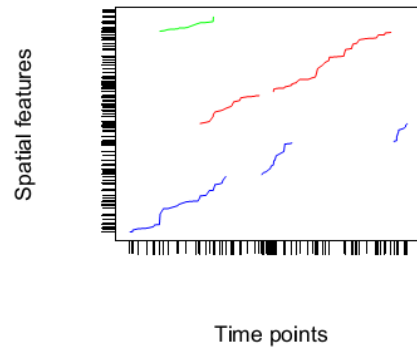
STS: sparse grid layout



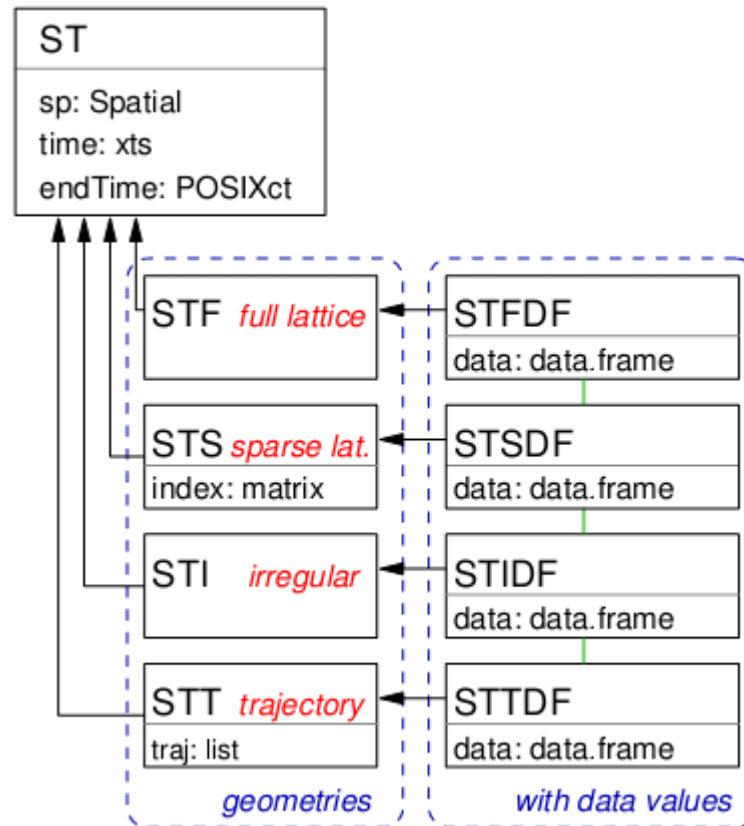
STI: irregular layout



STT: trajectory



Classes do spacetime



Rcpp

- Desenvolvido originalmente por Dirk Eddelbuettel
- Integração entre R e C++
- Classes e ferramentas
- Geração de pacotes



```

#include <R.h>
#include <Rdefines.h>
extern "C" SEXP vectorfoo(SEXP a, SEXP b){
    int i, n;
    double *xa, *xb, *xab; SEXP ab;
    PROTECT(a = AS_NUMERIC(a)); Coerção do tipo do SEXP
    PROTECT(b = AS_NUMERIC(b));
    n = LENGTH(a);
    PROTECT(ab = NEW_NUMERIC(n));
    xa=NUMERIC_POINTER(a); xb=NUMERIC_POINTER(b);
    xab = NUMERIC_POINTER(ab);
    double x = 0.0, y = 0.0 ;
    for (i=0; i<n; i++) xab[i] = 0.0;
    for (i=0; i<n; i++) {
        x = xa[i]; y = xb[i];
        xab[i] = (x < y) ? x*x : -(y*y);
    }
    UNPROTECT(3);
    return(ab);
}

```

```

#include <R.h>
#include <Rdefines.h>
extern "C" SEXP vectorfoo(SEXP a, SEXP b){
    int i, n;
    double *xa, *xb, *xab; SEXP ab;
    PROTECT(a = AS_NUMERIC(a));
    PROTECT(b = AS_NUMERIC(b));
    n = LENGTH(a);
    PROTECT(ab = NEW_NUMERIC(n));
    xa=NUMERIC_POINTER(a); xb=NUMERIC_POINTER(b);
    xab = NUMERIC_POINTER(ab);
    double x = 0.0, y = 0.0 ;
    for (i=0; i<n; i++) xab[i] = 0.0;
    for (i=0; i<n; i++) {
        x = xa[i]; y = xb[i];
        xab[i] = (x < y) ? x*x : -(y*y);
    }
    UNPROTECT(3);
    return(ab);
}

```

Proteção contra o garbage collector

Liberação de recursos (pilha)

```
#include <R.h>
#include <Rdefines.h>
extern "C" SEXP vectorfoo(SEXP a, SEXP b){
    int i, n;
    double *xa, *xb, *xab; SEXP ab;
    PROTECT(a = AS_NUMERIC(a));
    PROTECT(b = AS_NUMERIC(b));
    n = LENGTH(a);
    PROTECT(ab = NEW_NUMERIC(n));
    xa=NUMERIC_POINTER(a); xb=NUMERIC_POINTER(b);
    xab = NUMERIC_POINTER(ab);
    double x = 0.0, y = 0.0 ;
    for (i=0; i<n; i++) xab[i] = 0.0;
    for (i=0; i<n; i++) {
        x = xa[i]; y = xb[i];
        xab[i] = (x < y) ? x*x : -(y*y);
    }
    UNPROTECT(3);
    return(ab);
}
```

// Código R equivalente:

```
// vectorfoo <- function(x,y) ifelse(x<y, x*x, -  
(y*y))
```

```
#include <Rcpp.h>
```

```
RcppExport SEXP vectorFoo( SEXP a, SEXP b){
```

```
    Rcpp::NumericVector x(a), y(b);
```

```
    Rcpp::NumericVector xab =
```

```
        ifelse(x<y , x*x, -(y*y)); // Rcpp::sugar
```

```
    return xab;
```

```
}
```

Dados de C++ para R

- Tipos primitivos:
 - Int, double, bool, etc
- Objetos STL:
 - string, vector, map;
- Demais conversões devem ser declaradas pelo desenvolvedor

Exemplo

```
// [[Rcpp::export]]
std::vector<Rcpp::Datetime> getTime(te::st::TrajectoryDataSet &output)
{

    std::locale l("");
    std::setlocale(LC_ALL, "en_US.UTF-8");

    std::vector<Rcpp::Datetime> datetime;

    output.moveBeforeFirst(); // Inicio do dataset
    while(output.moveToNext())
    {
        std::auto_ptr<te::dt::DateTime> time = output.getTime();
        Rcpp::Datetime dt(time->toString(), FORMAT_TIME);
        datetime.push_back(dt);
        std::auto_ptr<te::gm::Geometry> geom = output.getGeometry();
    }

    std::setlocale(LC_ALL, l.name().c_str());

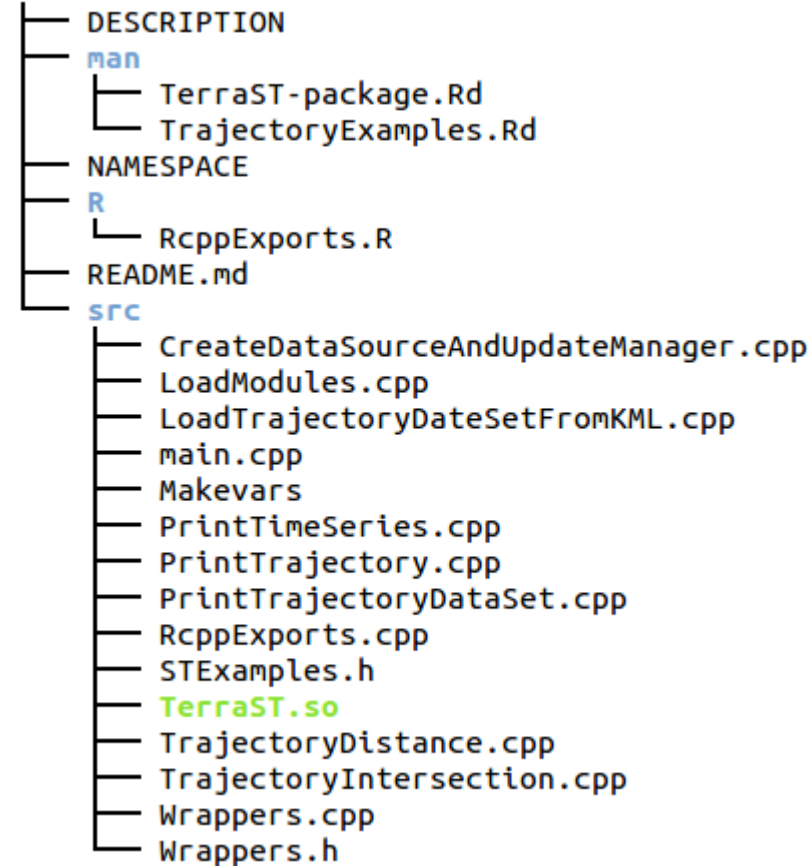
    return datetime;
}
```

Pacote TerraST

- Dependências:
 - pacotes Rcpp, spacetime.
 - TerraLib 5.

Estrutura do Pacote

TerraST



Resultados

- Pacote TerraST
- Arquivos KML – trajetória de leões marinhos
- Dados carregados em memória
- TerraLib – TrajectoryDataSet -> R – STI

Conclusão