RcppArmadillo



Hanna Zdulska & Dominik Rafacz



Co to jest Rcpp?



- -> Integracja C++ z R
- -> Szybszy niż czysty R
- -> Analogi typów R w C++
- -> Prosta składnia (ale czy na pewno?)
- -> Wsparcie budowy pakietów

```
#R
one ← function() 1L
//C++
int one() {
    return 1;
//Rcpp
Rcpp::cppFunction("int one() {
    return 1;
}")
```



```
#include<Rcpp.h>
\\[Rcpp::export]
Rcpp::NumericVector addNoise(
                         Rcpp::NumericVector a){
   Rcpp::NumericVector b = Rcpp::rnorm(a.size());
   Rcpp::NumericVector c = Rcpp::wrap(1);
   a = a + b + c;
   return Rcpp::List::create(
        Rcpp::Named("noisy") = a,
        Rcpp::Named("infos") = a.size())
```



Co to jest Armadillo?



- -> Biblioteka C++ do algebry liniowej
- -> Ponad 300 funkcji
- -> Szybka i zoptymalizowana

```
#include<armadillo.h>
int main(int argc, const char **argv) {
   arma::Mat<double> A = arma::randu(4, 4);
   arma::Mat<double> B = A * A.t();
   A.row(0) = A.row(1) + A.row(3);
   A.col(3).zeros();
   return 0;
```





Co to jest Rcpp Armadillo?



- -> Mapowanie obiektów Armadillo przez Rcpp do R
- -> Nie wymaga instalacji Armadillo

```
#include <RcppArmadillo.h>
using namespace Rcpp;
// [[Rcpp::depends(RcppArmadillo)]]
// [[Rcpp::export]]
List fastLm_impl(const arma::mat& X, const arma::colvec& y) {
    int n = X.n_rows, k = X.n_cols;
    arma::colvec coef = arma::solve(X, y);
    arma::colvec res = y - X*coef;
    double s2 = std::inner_product(res.begin(), res.end(),
                                   res.begin(), 0.0)/(n - k);
    arma::colvec std_err = arma::sqrt(s2 *
               arma::diagvec(arma::pinv(arma::trans(X)*X)));
    return Rcpp::List::create(Named("coefficients") = coef,
                              Named("stderr") = std err,
                              Named("df.residual") = n - k);
```



Filtr Kalmana





