Time Series Prediction

Jaakko Hollmén

Helsinki Institute for Information Techhnology
Aalto University, Department of Computer Science
Espoo, Finland
Web: http://users.ics.aalto.fi/jhollmen/

e-mail: Jaakko.Hollmen@aalto.fi

April 3, 2016

Acknowledgements, Part I

Collaborative work:

- Jarkko Tikka, Mikko Korpela and Jaakko Hollmén

Based on two publications by the authors:

- Jarkko Tikka, Jaakko Hollmén (2008). Sequential input selection algorithm for long-term prediction of time series. Neurocomputing, 71(13-15), pp. 2604-2615. ISSN 0925-2312, `http://doi.org/10.1016/j.neucom.2007.11.037`
- Mikko Korpela (2015). sisal: Sequential Input Selection Algorithm. R package version 0.46. `http://cran.r-project.org/package=sisal`

Acknowledgements, Part II

Collaborative work:

- Indrė Žliobaitė, Heikki Junninen and Jaakko Hollmén

Based on two publications by the authors:

- Indrė Žliobaitė, Jaakko Hollmén. Optimizing regression models for data streams with missing values. Machine Learning, 99(1), 47-73, April 2015. `http://dx.doi.org/10.1007/s10994-014-5450-3`
- Indrė Žliobaitė, Jaakko Hollmén, Heikki Junninen. Regression models tolerant to massively missing data: a case study in solar radiation nowcasting. Atmospheric Measurement Techniques Discussions, 7, 7137-7174, 2014. `http://dx.doi.org/10.5194/amtd-7-7137-2014`

Machine Learning and Data Mining

Research Interests

- Artificial Intelligence (Deep belief networks etc.)
- Machine Learning
- Data Mining
- Computer Science
- Applications in environmental informatics and health

Contents of the Lecture, Part I

Topics on Time Series Prediction:

- Introduction and background
- Minitopics: Curse of dimensionality, Bootstrap, Generalization, Cross-Validation
- Variable Selection in Time Series prediction models
- Missing data in Time Series Prediction
- Hands-on exercise with R SISAL package

Time Series Prediction: Introduction

Some useful methods for time series analysis and prediction:

- Wavelets
- Fourier analysis, FFT, DFT, Goertzel algorithm
- Dynamical models
- Probabilistic models: Hidden Markov Models, Kalman filters, Dynamic Bayesian Networks
- Empirical mode demposition, SAX (Symbolic Aggregate Approximation)

How to choose an appropriate method for your problem?

Time Series Prediction: Introduction

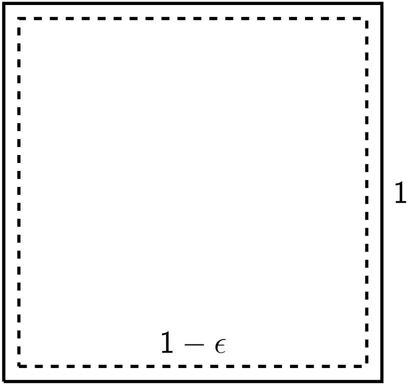
Two roles in data analysis:

- Users of data analysis: tools, understanding of methods
- Developers of data analysis: understanding of theory, making tools

Interdisciplinary research:

- Experts in the domain, like space physics
- Experts in data analysis
- Data analysis is not a service, but a collaboration!
- Think what you can achieve together, before the experiment!

Curse of Dimensionality



Curse of Dimensionality

Curse of dimensionality is a fundamental law in data analysis

- Assume a d-dimensional unit hybercube (side equals 1), with Volume $V_1 = 1^d$.
- Internal points are points if they are within a cube, side equals $1 - \epsilon$, with $\epsilon > 0$, with Volume $V_{1-\epsilon} = (1-\epsilon)^d$
- Data is uniformly distributed in the cube
- Ratio of internal points to all points is $R = \frac{V_{1-\epsilon}}{V_1} = \frac{(1-\epsilon)^d}{1^d} = (1-\epsilon)^d$
- If dimensions grow without bound: $\lim_{d \to \infty} (1-\epsilon)^d \to 0$.

This means (no matter how small our ϵ is) that in very high dimensions all the points are *on the surface of the cube*!

Boostrapping for Uncertainty Estimation

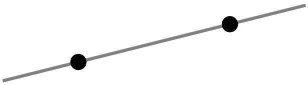
The average of the data set:

- Data Set: $X = \{1.0, 1.3, 2.7, 4.9, 5.1\}$
- Sum of the data points: $\sum_{i=1}^{5} x_i = 15$
- Average value: $\frac{1}{5} \sum_{i=1}^{5} x_i = 3.0$

Can we quantify the uncertainty of the average value?

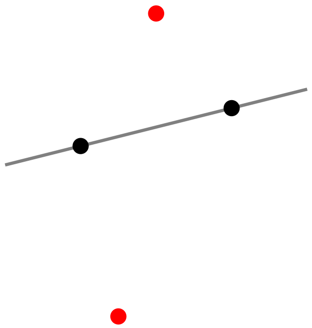
- Answer: Bootstrapping, sampling with replacement
- Sample several data sets (N=5) with replacement
- Example 1: $X^* = \{1.0, 1.0, 2.7, 4.9, 5.1\}$
- Example 2: $X^* = \{1.0, 1.3, 2.7, 4.9, 5.1\}$
- Example 3: $X^* = \{1.3, 1.3, 2.7, 4.9, 4.9\}$
- and calculate the average for each data set to get a empirical distribution of the average value

Generalization



- Generalization refers to the ability to generalize to *unseen data points* measured in the future
- The aim of predictive modeling is to generalize, not to describe the data set at hand
- A perfect fit?

Generalization



- Generalization refers to the ability to generalize to *unseen data points* measured in the future
- Overfitting: fitting to training data too well, not being able to generalize
- New data arrives..

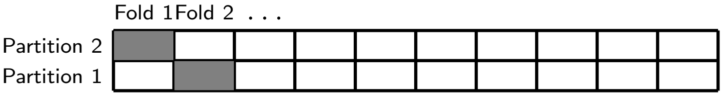
Cross-validation for model assessment

- Anticausality: we can not optimize with regard to future, unseen data points
- We can simulate this situation: cross-validation!
- Divide the data into a training data and hold-out data, that is kept hidden from the data analyst
- Measure the model performance: training data set
- Measure the model performance: hold-out data set, or sometimes called the validation set, or the test set

Cross-validation for model assessment

Example: 10-fold cross-validation repeated 2 times

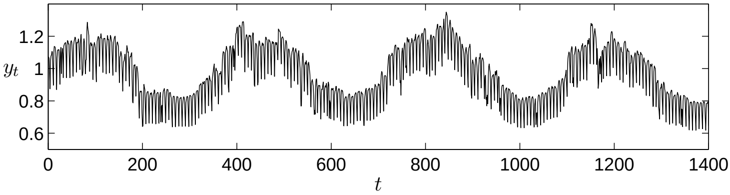
- Divide, or partition the data into ten parts
- Use nine parts for training, one part is a hold-out set, repeat 10 times for each choice of a hold-out set
- repeat twice, second time with a new partition

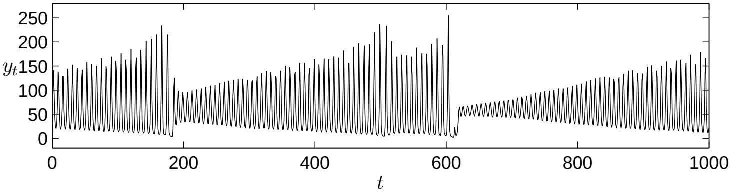


You can estimate the errors based on 20 modeling efforts:

- 20 estimates for the training set, 20 for the hold-out set
- The hold-out sets emulate or mimic the *future, unseen data sets*

Time Series: Some Examples





Strategies: Time Series Prediction

- Turning the time series prediction problem into (a kind of) a static regression problem
- Autoregressive time series prediction model
- $x_{t+1} = f(x_t, x_{t-1}, x_{t-2}, \ldots, x_{t-d-1})$, f linear
- Takens theorem

Take a look at an example:

- Consider a time series: $X = \{1, 2, 3, 4, 5, 6, 7, 8\}$
- `library(sisal)`
- `laggedData(1:8, 0:3, 1)`
- `laggedData(sunspot.month, 0:10, 1)`

Strategies: Time Series Prediction

Choices to implement or use the regression model:

- Recursive Prediction Strategy
- Direct Prediction Strategy
- And variants

Recursive Prediction Strategy

Predictions are made one step-ahead at the time:

- $\hat{x}_{t+1} = f(x_t, x_{t-1}, x_{t-2}, \ldots x_{t-d+1})$
- $\hat{x}_{t+2} = f(\hat{x}_{t+1}, x_t, x_{t-1}, x_{t-2}, \ldots x_{t-d})$
- Benefits: Only one prediction model f to estimate
- Disadvantages: Accumulation of errors in each step

Direct Prediction Strategy

Predictions are made k steps ahead at once:

- $\hat{x}_{t+k} = f_k(x_t, x_{t-1}, x_{t-2}, \ldots x_{t-d+1})$
- Benefits: The problem of k steps ahead prediction is solved directly
- Disadvantages: Must train a model f_k for each k

Take a look at an example:

- Consider a time series: $X = \{1, 2, 3, 4, 5, 6, 7, 8\}$
- `library(sisal)`
- `laggedData(1:8, 0:3, 3)`
- `laggedData(sunspot.month, 0:10, 6)`

Time Series Prediction: Long-term Prediction

What is long-term prediction depends on the context!

- Interesting phenomena vary from milliseconds to centuries
- Prediction further into the future is more difficult
- Direct Prediction Strategy is preferred

Sequential Input Selection Algorithm (SISAL)

Let us assume that there are N measurements available from a time series $x_t,\ t = 1, \ldots, N$. Future values of time series x_t are predicted using the previous values $x_{t-i},\ i = 1, \ldots, l$. If the dependency between the output x_t and the inputs x_{t-i} is assumed to be linear it can be written as

$$x_t = \sum_{i=1}^{l} \beta_i x_{t-i} + \varepsilon_t, \tag{1}$$

which is a linear autoregressive process of order l or briefly AR(l). The errors ε_t are supposed to beindependently normally distributed with zero mean and common finite variance $\varepsilon_t \sim \mathrm{N}(0, \sigma^2)$.

Sequential Input Selection Algorithm (SISAL)

Linear model as a predictor:

- Using linear prediction models implicity implies linearization of the system
- Validity of assumptions of the linear model?
- Simple, too simple?
- You can build non-linearity on top of linearity afterwards

Input Variable Selection in Time Series Prediction

Start with a time series model with a lot of variables:

- You don't really know which ones are the correct model variables
- You want to reduce complexity (curse of dimensionality)
- Perform *Variable Selection* to reduce the number of variables
- SISAL implements input variable selection in time series models

Input Variable Selection in Time Series Prediction

Input Variable Selection: Search Strategies

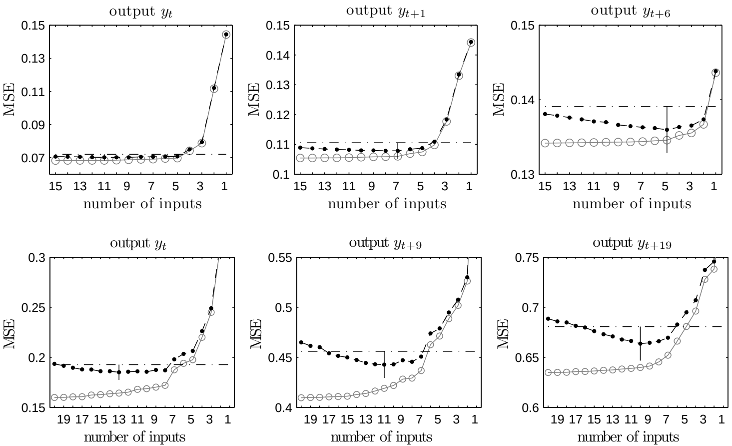
- Forward-selection: greedily add variables
- Example: $\{\} \to \{x_1\} \to \{x_1, x_5\} \ldots$
- Backward selection: greedily remove variables
- Example: $\ldots \to \{x_1, x_4, x_6\} \to \{x_4, x_6\} \to \{x_4\} \to \{\}$
- And a lot of variants ...

Input Variable Selection in Time Series Prediction

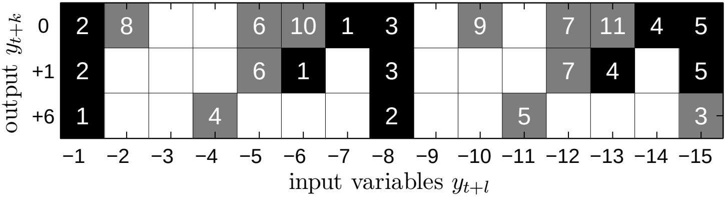
SISAL uses Backward Selection Type of Search Strategy

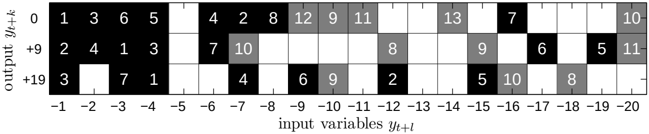
- Start with a full model, remove variables
- Important Point: take uncertainty into account (by bootstrapping)
- Advantage: you include all the variables in the beginning
- Disadvantage: you may end up with large models in the beginning (use regularization)

Input Variable Selection in Time Series Prediction

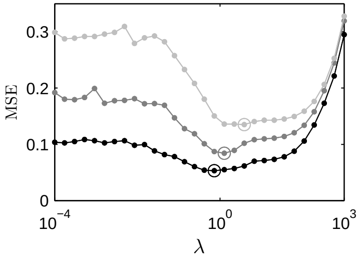


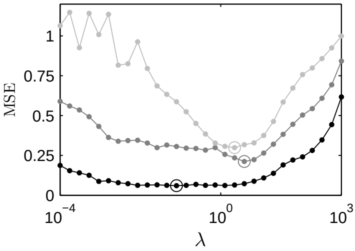
Input Variable Selection in Time Series Prediction



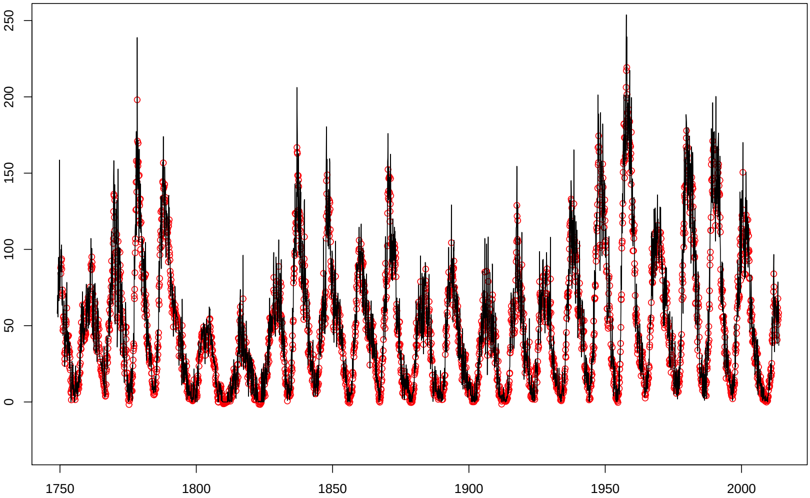


Input Variable Selection in Time Series Prediction





Predicting monthly sunspots: 1 month ahead

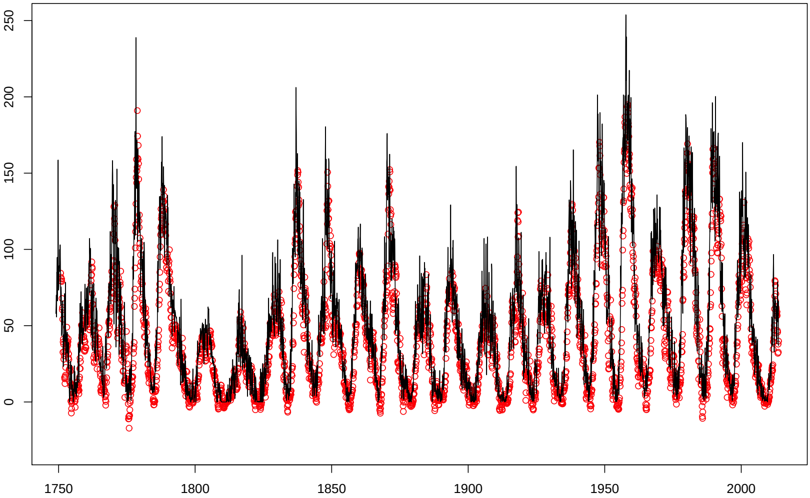


Predicting monthly sunspots: 1 month ahead

Future values can be predicted with the following equation:

$$
\begin{aligned}
x_t = 0.00 + 0.56x_{t-1} + 0.11x_{t-2} + 0.10x_{t-3} \\
+ 0.09x_{t-4} + 0.04x_{t-5} + 0.07x_{t-6} \\
+ 0.10x_{t-9} - 0.03x_{t-13} - 0.10x_{t-16}
\end{aligned}
$$

Predicting monthly sunspots: 6 months ahead

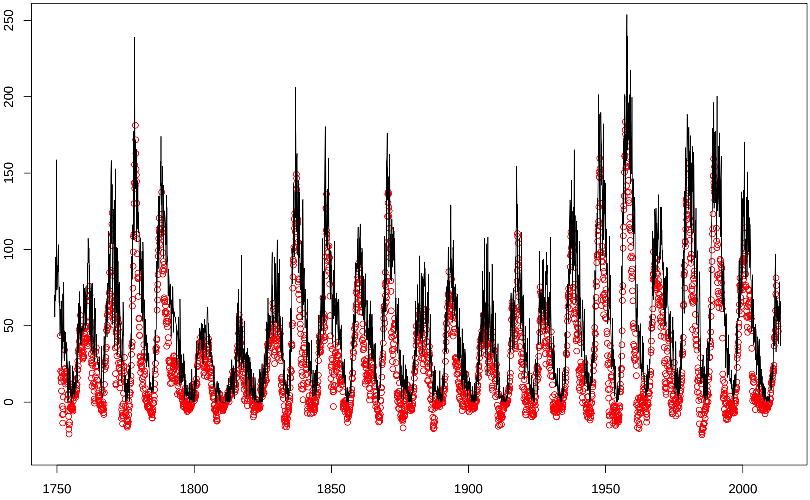


Predicting monthly sunspots: 6 months ahead

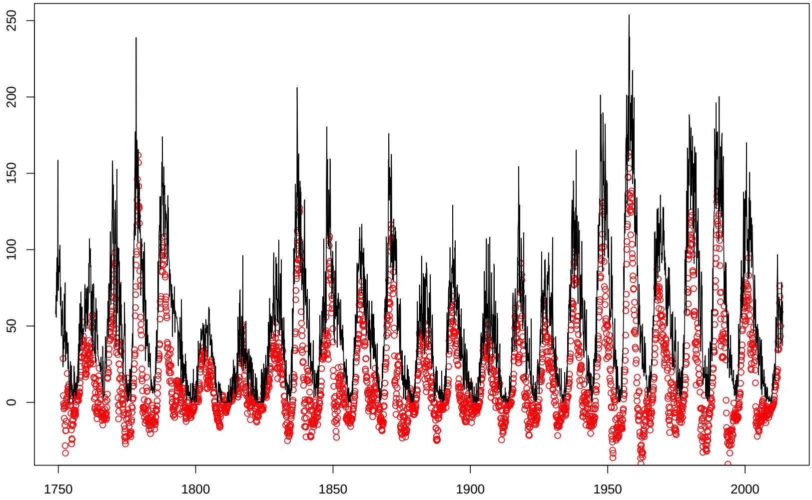
Future values can be predicted with the following equation:

$$x_t = 0.00 + 0.40x_{t-1} + 0.16x_{t-2} + 0.13x_{t-3} \\ + 0.19x_{t-4} + 0.12x_{t-5} + 0.11x_{t-6} + 0.84x_{t-7} \\ + 0.07x_{x-9} - 0.11x_{t-13} - 0.06x_{t-14} \\ - 0.09x_{t-15} - 0.2x_{t-16}$$

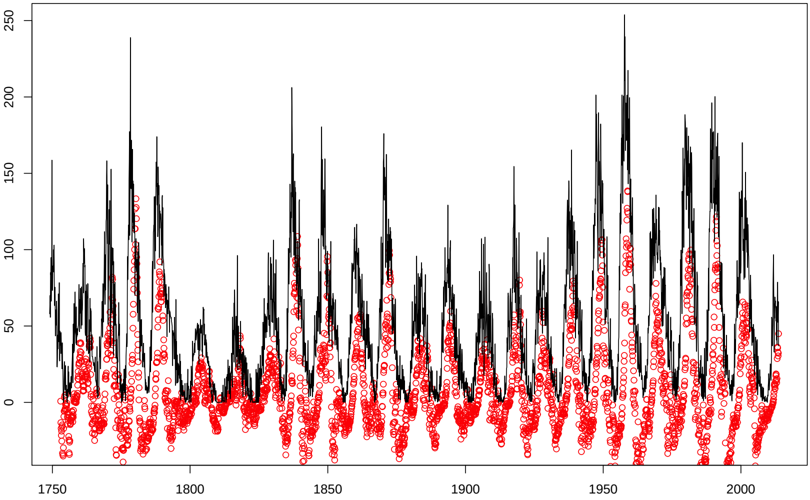
Predicting monthly sunspots: 12 months ahead



Predicting monthly sunspots: 18 months ahead



Predicting monthly sunspots: 24 months ahead



Predicting monthly sunspots with SISAL

Take a look at an example:

- `library(sisal)`
- `sunsp <- laggedData(sunspot.month, 0:10, 1)`
- `sunsp$X[1:5,]`
- `sunsp$y[1:5]`
- `spmodel <- sisal(sunsp$X, sunsp$y, Mtimes=10, kfold=5)`
- `names(spmodel)`
- `plotSelected(spmodel)`

Linear prediction with missing data

Brief summary of the surprising results:

- Indrė Žliobaitė, Jaakko Hollmén. Optimizing regression models for data streams with missing values. Machine Learning, 99(1), 47-73, April 2015. http://dx.doi.org/10.1007/s10994-014-5450-3

Linear prediction with missing data

Brief summary of one particular problem in missing data:

- Think of the problem, when you *train* your prediction model by regression with *full data* (no missing data)
- In deployment, you have *missing data* in prediction
- Scope of this work: On-line analysis, model-based imputation is not possible (limitations on energy or computational power)
- Surprising result: predictions are very soon useless, with very little missing data

Linear prediction with missing data

Estimation according to the principle of least-squares

$$\hat{\vec{\beta}}_{\mathsf{OLS}} = \arg\min_{\vec{\beta}} \left((\vec{y} - \mathbf{X}\vec{\beta})^{\mathrm{T}} (\vec{y} - \mathbf{X}\vec{\beta}) \right) = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\ \vec{y}$$

With Regularization: Ridge Regression, Weight Decay

$$\hat{\vec{\beta}}_{\mathsf{RR}} = \arg\min_{\vec{\beta}} \left((\vec{y} - \mathbf{X}\vec{\beta})^{\mathrm{T}} (\vec{y} - \ \mathbf{X}\vec{\beta}) + \lambda \vec{\beta}^{\mathrm{T}} \vec{\beta} \right)$$

$$= (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\vec{y}$$

Linear prediction with missing data

Assume a single probability of any variable missing: p

- Then: $\hat{\beta}_{ROB} = \left((1-p)\mathbf{X}^T\mathbf{X} + pn\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}$

Probabilities of ith variable missing:
$\mathbf{p} = (p_i) = (p_1, p_2, \ldots, p_r)^T$.

- $\hat{\beta}_{ROB} = \left(\mathbf{X}^T\mathbf{X}(\mathbf{I} - diag(\mathbf{p})) + diag(\mathbf{p})n\right)^{-1}\mathbf{X}^T\mathbf{y}$

Hands-on exercise with R package SISAL

Sequential Input Variable Selection Algorithm

- Long-term time-series prediction:
  $\hat{x}_{t+k} = f(x_t, x_{t-1}, x_{t-2}, \ldots x_{t-d+1})$
- Select Input variables in the model simultaneously
- Bootstrapping for uncertainty estimation
- Make informed choices taking uncertainty into account
- Parsimionious, or sparse models

Hands-on exercise with R package SISAL

The R Project for Statistical Computing

- R is a free software environment for statistical computing and graphics
- `https://www.r-project.org`
- Active ecosystem, widely used

The Comprehensive R Archive Network

- Network of servers that store identical, up-to-date, versions of code and documentation for R
- `https://cran.r-project.org/`
- Currently, 8178 available packages
- "Climate", 23 packages
- "Solar", 8 packages

Hands-on exercise with R package SISAL

Sequential Input Selection Algorithm (SISAL)

- Available from CRAN
- `http://CRAN.R-project.org/package=sisal`

Hands-on exercise with R package SISAL

Basic commands in R

- `quit()`
- `hello <- "World"`
- `a <- 3.14`
- `a <- a + 1`
- `vec <- c(1,2,3)`
- `print(hello)`
- List all variables: `ls()`
- Remove all variables: `rm(list=ls())`

Hands-on exercise with R package SISAL

Useful commands for the exercise:

- Load package SISAL: `library(sisal)`
- Load package SISAL: `library("sisal")`
- Help with the SISAL package: `help("sisal-package")`
- Run a simple test run: `sisalTest()`