# Untitled

Jan-Philipp Kolb

21 Mai, 2019

# Introduction to R

## Introduction round

### Please tell us shortly...

- Where are you from? What are you studying/working?
- What are your expectations of this course?
- Where do you think you can use Machine Learning in the future?

## Preliminaries

- This topic is huge - we concentrate on presenting the applications in R
- Usually we have big differences in knowledge and abilities of the participants - please tell, if it is too fast or slow.
- We have many **exercises** because at the end you can only learn on your own
- We have many **examples** - try them!
- If there are questions - always ask
- R is more fun together - ask your neighbor

## Content of the course - day 1

- The first section is about laying the foundations in R. We will need all things covered later on.
- The second section is an introduction to the field of machine learning.
- The third part is on regression and classification.

## Why R is a good choice . . .

- . . . because it is an **open source language**
- . . . outstanding graphs - **graphics**, **graphics**, **graphics**
- . . . relates to other languages - **R can be used in combination with other programs** - e.g. **data linking**
- . . . R can be used **for automation**
- . . . Vast Community - **you can use the intelligence of other people ;-)** and new statistical methodologies are implemented quite fast
- Because R can be combined with other programs like `PostgreSQL` or `Python`

## Constraints

### Newer modules in Python

- Machine learning is a field that changes rapidly.
- Some new tools are first developed in Python.
- The package `reticulate` offers the possibility to use these modules from an R environment.
- Good news - Python is also Open Source

### Big Data

- Especially if you work with web data, you quickly have to deal with large amounts of data.
- Therefore one must fall back on databases and parallelization strategies, which can be used in R.

## Import data

```
?read.csv
?read.csv2
```

```
path <- "https://raw.githubusercontent.com/thomaspernet/data_csv
dname <- "titanic_csv.csv"
titanic <- read.csv(paste0(path,dname))
```

### Using a path to import data

## The download the data from UCI.

```
url<-'http://archive.ics.uci.edu/ml/machine-learning-databases/0
Yacht_Data <- readr::read_table(file = url)
```

## Built in datasets

- A sample dataset is often provided to demonstrate the functionality of a package.
- These records can be loaded using the `data` command.

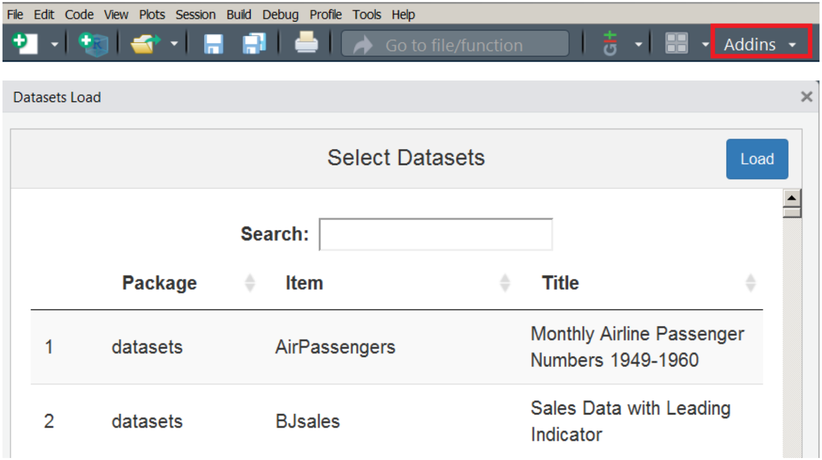
```
data(iris)
```

- There is also a **RStudio Add-In** that helps to find a built-in dataset.

```
install.packages("datasets.load")
```

# Exkurs RStudio Addins

- Oben rechts befindet sich ein Button Addins


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Datasets Load
Select Datasets
Load
Search:

| | Package | Item | Title |
|---|---|---|---|
| 1 | datasets | AirPassengers | Monthly Airline Passenger Numbers 1949-1960 |
| 2 | datasets | BJsales | Sales Data with Leading Indicator |

## The titanic dataset

```
kable(head(titanic))
```

## Exercise

Load the the built-in dataset `swiss` and answer the following questions:

- How many observations and variables are available?
- What is the scale level of the variables?

Create an interactive data table

## The function scan to import data

- scan has an easy way to distinguish comments from data

```
?scan
```

### Example dataset

```
cat("TITLE extra line", "# a comment","2 3 5 7", "11 13 17",
    file = "../data/ex.data", sep = "\n")
```

### Import data and skip the first line

```
pp <- scan("../data/ex.data", skip = 1, quiet = TRUE)
```

```
pp <- scan("../data/ex.data", comment.char="#", skip = 1, quiet
```

## The R-package data.table

### Get an overview

```
data(airquality)
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

### Overview with data.table

```
library(data.table)
airq <- data.table(airquality)
airq
```

```
##       Ozone Solar.R Wind Temp Month Day
```

## Exercise

```
x <- c(0.109, 0.359, 0.63, 0.996, 0.515, 0.142, 0.017, 0.829, 0.
```

- Compute the logarithm of `x`, return suitably lagged and iterated differences,
- compute the exponential function and round the result

```
## [1]  3.3  1.8  1.6  0.5  0.3  0.1 48.8  1.1
```

## The pipe operator

```
library(magrittr)

# Perform the same computations on `x` as above
x %>% log() %>%
    diff() %>%
    exp() %>%
    round(1)
```

```
## [1]  3.3  1.8  1.6  0.5  0.3  0.1 48.8  1.1
```

## How to deal with missing values

```
?na.omit
```

```
airq
```

```
##      Ozone Solar.R Wind Temp Month Day
##   1:    41     190  7.4   67     5   1
##   2:    36     118  8.0   72     5   2
##   3:    12     149 12.6   74     5   3
##   4:    18     313 11.5   62     5   4
##   5:    NA      NA 14.3   56     5   5
##  ---
## 149:    30     193  6.9   70     9  26
## 150:    NA     145 13.2   77     9  27
## 151:    14     191 14.3   75     9  28
## 152:    18     131  8.0   76     9  29
## 153:    20     223 11.5   68     9  30
```

```
na.omit(airq)
```

```
##      Ozone Solar.R Wind Temp Month Day
```

## Clean the titanic data set

```
library(dplyr)
library(magrittr)
# Drop variables
clean_titanic <- titanic %>%    mutate(pclass = factor(pclass, l
    survived = factor(survived, levels = c(0, 1), labels = c('No
na.omit()
```

- `pclass = factor(pclass, levels = c(1,2,3), labels= c('Upper', 'Middle', 'Lower'))`: Add label to the variable pclass. 1 becomes Upper, 2 becomes MIddle and 3 becomes lower
- `factor(survived, levels = c(0,1), labels = c('No', 'Yes'))`: Add label to the variable survived. 1 Becomes No and 2 becomes Yes
- `na.omit()`: Remove the NA observations

## Get an overview of the data

```
glimpse(clean_titanic)
```

```
## Observations: 1,045
## Variables: 13
## $ X         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
## $ pclass    <fct> Upper, Upper, Upper, Upper, Upper, Upper, U
## $ survived  <fct> Yes, Yes, No, No, No, Yes, Yes, No, Yes, No
## $ name      <fct> "Allen, Miss. Elisabeth Walton", "Allison,
## $ sex       <fct> female, male, female, male, female, male, f
## $ age       <dbl> 29.0000, 0.9167, 2.0000, 30.0000, 25.0000,
## $ sibsp     <int> 0, 1, 1, 1, 1, 0, 1, 0, 2, 0, 1, 1, 0, 0, 0
## $ parch     <int> 0, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
## $ ticket    <fct> 24160, 113781, 113781, 113781, 113781, 1995
## $ fare      <dbl> 211.3375, 151.5500, 151.5500, 151.5500, 151
## $ cabin     <fct> B5, C22 C26, C22 C26, C22 C26, C22 C26, E12
## $ embarked  <fct> S, S, S, S, S, S, S, S, S, C, C, C, C, S, S
## $ home.dest <fct> "St Louis, MO", "Montreal, PQ / Chestervill
```

## Example Data - Housing Values in Suburbs of Boston

```
library(MASS)
data <- Boston
```

```
kable(head(data))
```

| crim | zn | indus | chas | nox | rm | age | dis | rad | tax | ptratio | black |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.00632 | 18 | 2.31 | 0 | 0.538 | 6.575 | 65.2 | 4.0900 | 1 | 296 | 1 | |
| 0.02731 | 0 | 7.07 | 0 | 0.469 | 6.421 | 78.9 | 4.9671 | 2 | 242 | 1 | |
| 0.02729 | 0 | 7.07 | 0 | 0.469 | 7.185 | 61.1 | 4.9671 | 2 | 242 | 1 | |
| 0.03237 | 0 | 2.18 | 0 | 0.458 | 6.998 | 45.8 | 6.0622 | 3 | 222 | 1 | |
| 0.06905 | 0 | 2.18 | 0 | 0.458 | 7.147 | 54.2 | 6.0622 | 3 | 222 | 1 | |
| 0.02985 | 0 | 2.18 | 0 | 0.458 | 6.430 | 58.7 | 6.0622 | 3 | 222 | 1 | |

## Normalize your data

```
maxs <- apply(data, 2, max)
mins <- apply(data, 2, min)
scaled <- as.data.frame(scale(data, center = mins, scale = maxs
```

## The command sample

- We can use this command to draw a sample.
- We need the command later to split our dataset into a test and a training dataset.

```
sample(1:10,3,replace=T)
```

```
## [1]  6 10  1
```

```
sample(1:10,3,replace=T)
```

```
## [1] 9 9 5
```

## Set a seed

- set.seed is the recommended way to specify seeds.
- If we set a seed, we get the same result for random events.
- This function is mainly required for simulations.

```
set.seed(234)
sample(1:10,3,replace=T)
```

```
## [1] 1 2 2
```

```
set.seed(234)
sample(1:10,3,replace=T)
```

```
## [1] 1 2 2
```

## Time measurement

```
start_time <- Sys.time()
ab <- runif(10000000)
end_time <- Sys.time()

end_time - start_time
```

```
## Time difference of 0.791045 secs
```

## How many cores are available

```
library(doParallel)
detectCores()
```

```
## [1] 4
```

## Make cluster

```
cl <- makeCluster(detectCores())
registerDoParallel(cl)
```

```
start_time <- Sys.time()
ab <- runif(10000000)
end_time <- Sys.time()
```

```
end_time - start_time
```

```
## Time difference of 0.9540551 secs
```

```
stopCluster(cl)
```

```
?parallel::makeCluster
```

## The swirl package

```r
install.packages("swirl")
```

```r
library("swirl")
swirl()
```

## Resources

- Course materials for the Data Science Specialization

## Introduction to machine learning

## Modern Machine Learning Algorithms

Categorizing machine learning algorithms is tricky, and there are several reasonable approaches; they can be grouped into generative/discriminative, parametric/non-parametric, supervised/unsupervised, and so on.

## Machine Learning - Components

- Feature Extraction + Domain knowledge
- Feature Selection
- Choice of Algorithm

Naive Bayes, Support Vector Machines, Decision Trees, k-Means Clustering, . . .

- Training
- Choice of Metrics/Evaluation Criteria
- Testing

## Feature selection

In machine learning, feature selection is the process of selecting a subset of relevant features (variables, predictors) for use in model construction.

Feature selection techniques are used for four reasons:

- simplification of models to make them easier to interpret by researchers/users,
- shorter training times,
- to avoid the curse of dimensionality,
- enhanced generalization by reducing overfitting (formally, reduction of variance)

## Supervised vs unsupervised learning

### Supervised Learning

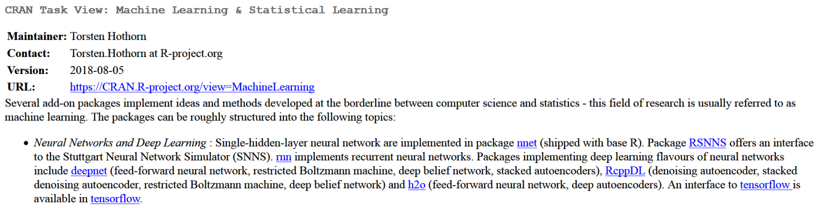
- we have prior knowledge of what the output values for our samples should be.

## Task: Find R-packages

Go to https://cran.r-project.org/ and search for packages that,...

- can be used for lasso regression

## Task View Machine Learning



**CRAN Task View: Machine Learning & Statistical Learning**

|  |  |
|---|---|
| **Maintainer:** | Torsten Hothorn |
| **Contact:** | Torsten.Hothorn at R-project.org |
| **Version:** | 2018-08-05 |
| **URL:** | https://CRAN.R-project.org/view=MachineLearning |

Several add-on packages implement ideas and methods developed at the borderline between computer science and statistics - this field of research is usually referred to as machine learning. The packages can be roughly structured into the following topics:

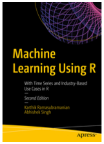
- *Neural Networks and Deep Learning* : Single-hidden-layer neural network are implemented in package nnet (shipped with base R). Package RSNNS offers an interface to the Stuttgart Neural Network Simulator (SNNS). rnn implements recurrent neural networks. Packages implementing deep learning flavours of neural networks include deepnet (feed-forward neural network, restricted Boltzmann machine, deep belief network, stacked autoencoders), RcppDL (denoising autoencoder, stacked denoising autoencoder, restricted Boltzmann machine, deep belief network) and h2o (feed-forward neural network, deep autoencoders). An interface to tensorflow is available in tensorflow.

## Install all packages of a task view

```
install.packages("ctv")
ctv::install.views("MachineLearning")
```

## Prediction vs. Causation in Regression Analysis

## Literature for machine learning


Machine Learning Using R

With Time Series and Industry-Based Use Cases in R

Second Edition

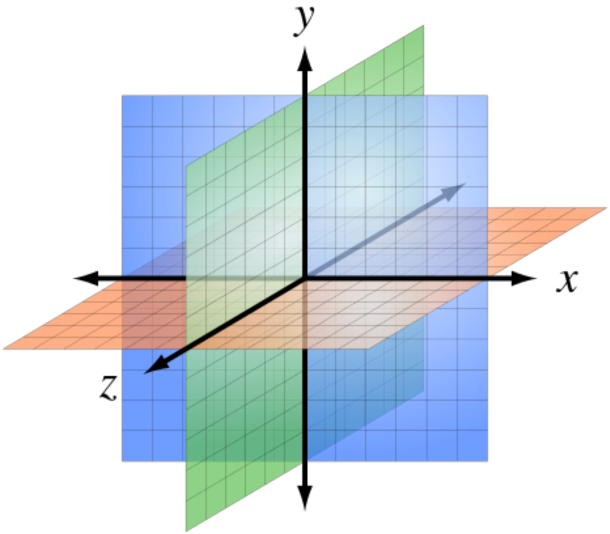
Karthik Ramasubramanian
Abhishek Singh

Apress

## Introduction to machine learning with R

- Your First Machine Learning Project in R Step-By-Step
- chapter about machine learning in awesome R
- Shiny App for machine learning

## The Curse of Dimensionality



## What is supervised learning?

Supervised learning includes tasks for “labeled” data (i.e. you have a target variable).

- In practice, it’s often used as an advanced form of predictive modeling.
- Each observation must be labeled with a “correct answer.”
- Only then can you build a predictive model because you must tell the algorithm what’s “correct” while training it (hence, “supervising” it).
- Regression is the task for modeling continuous target variables.
- Classification is the task for modeling categorical (a.k.a. “class”) target variables.

## Random Forest

*Random forest aims to reduce the previously mentioned correlation issue by choosing only a subsample of the feature space at each split. Essentially, it aims to make the trees de-correlated and prune the trees by setting a stopping criteria for node splits, which I will cover in more detail later.*

## What are the advantages and disadvantages of decision trees?

Advantages: Decision trees are easy to interpret, nonparametric (which means they are robust to outliers), and there are relatively few parameters to tune.

Disadvantages: Decision trees are prone to be overfit.

- This can be addressed by ensemble methods like random forests or boosted trees.

## Ensembling

Ensembles are machine learning methods for combining predictions from multiple separate models.

### Bagging

attempts to reduce the chance overfitting complex models.

- It trains a large number of "strong" learners in parallel.
- A strong learner is a model that's relatively unconstrained.
- Bagging then combines all the strong learners together in order to "smooth out" their predictions.

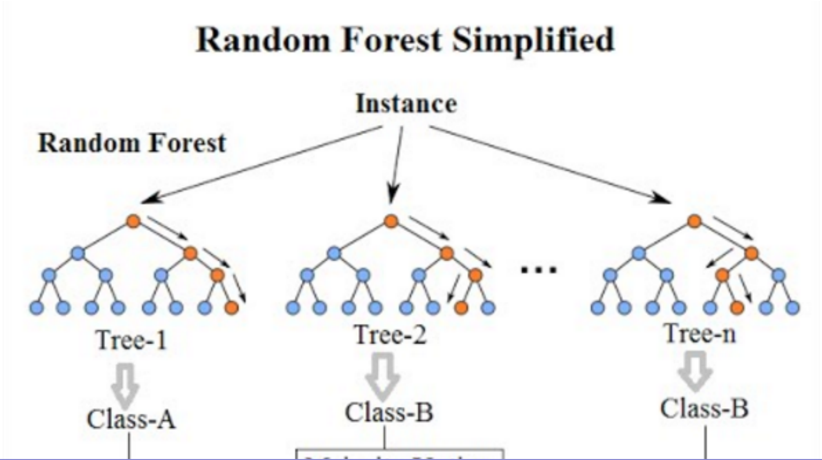
### Boosting

attempts to improve the predictive flexibility of simple models.

- It trains a large number of "weak" learners in sequence.
- A weak learner is a constrained model (i.e. you could limit the max depth of each decision tree).
- Each one in the sequence focuses on learning from the mistakes of

## Random forest

- Ensemble learning method - multitude of decision trees
- Random forests correct for decision trees' habit of overfitting to their training set.



## Gradient boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

The idea of gradient boosting originated in the observation by Leo Breiman that boosting can be interpreted as an optimization algorithm on a suitable cost function.

Breiman, L. (1997). "Arcing The Edge". Technical Report 486. Statistics Department, University of California, Berkeley.

## Advantages of gradient boosting

- Often provides predictive accuracy that cannot be beat.
- Lots of flexibility - can optimize on different loss functions and provides several hyperparameter tuning options that make the function fit very flexible.
- No data pre-processing required - often works great with categorical and numerical values as is.
- Handles missing data - imputation not required.

## Disadvantages of gradient boosting

- GBMs will continue improving to minimize all errors. This can overemphasize outliers and cause overfitting. Must use cross-validation to neutralize.
- Computationally expensive - GBMs often require many trees ($>$1000) which can be time and memory exhaustive.
- The high flexibility results in many parameters that interact and influence heavily the behavior of the approach (number of iterations, tree depth, regularization parameters, etc.). This requires a large grid search during tuning.
- Less interpretable although this is easily addressed with various tools (variable importance, partial dependence plots, LIME, etc.).

## Two types of errors for tree methods

### Bias related errors

- Adaptive boosting
- Gradient boosting

### Variance related errors

- Bagging
- Random forest

## Links

- Presentations on ‘Elements of Neural Networks & Deep Learning’
- Understanding the Magic of Neural Networks
- Neural Text Modelling with R package ruimtehol
- Feature Selection using Genetic Algorithms in R
- Lecture slides: Real-World Data Science (Fraud Detection, Customer Churn & Predictive Maintenance)
- Automated Dashboard for Credit Modelling with Decision trees and Random forests in R
- Looking Back at Google’s Research Efforts in 2018
- Selecting ‘special’ photos on your phone
- Open Source AI, ML & Data Science News
- Google‘s Machine Learning Crash Course
- A prelude to machine learning