Starterguide: **D**ata**b**ionic **T**oolbox

# Was ist die DBT?

Die DBT ist eine Sammlung von Packages für die Skriptsprache R. Sie wird über ein SVN System gewartet und aktualisiert. Einige der Pakete sind parallel auch auf CRAN, einem großen semi-offiziellen Package Hoster für R.

# SVN

Für den Zugang auf die DBT wird in der Regel ein SVN Zugang benötigt. Dieser kann grob mit einer geteilten „Dropbox“ verglichen werden. Sie erhalten Username, Passwort und Adressen auf das SVN System von uns. Mit diesen werden die Daten auf dem Server mit den Dateien auf Ihrem Rechner synchronisiert. Die Wahl Ihrer SVN Software steht Ihnen frei, wobei der Großteil der Arbeitsgruppe TortoiseSVN verwendet.

Mit Ihrem SVN Client können Sie einen „Checkout“ auf die Adressen im SVN System ausführen, wobei die Ordner erstmalig auf Ihren Rechner heruntergeladen werden. Danach können Sie mit „commit“ Ihre aktualisierten Dateien auf den Server laden, oder mittels „update“ die Daten auf Ihrem Rechner auf den Stand des Servers aktualisieren.

Der eigentliche Inhalt der DBT liegt unter PUB/dbt/ wobei jeder Unterordner ein eigenes Package darstellt.­­­­

# Vorbedingungen und Einrichtung

Bevor Sie die DBT installieren, müssen Sie R, Rtools und RStudio­­ auf Ihrem Rechner installieren. R stellt den eigentlichen Interpreter dar, der den Code ausführt. RStudio ist eine Umgebung zum deutlich komfortableren Arbeiten mit R. Die Rtools enthalten unter anderem einen C++ Compiler welcher für einige der Packages benötigt wird.

<https://cran.r-project.org/bin/windows/base/> - R

<https://cran.r-project.org/bin/windows/Rtools/> - Rtools

<https://www.rstudio.com/products/rstudio/download/#download> - RStudio

­

Nachdem Sie den DBT Ordner über das SVN System bezogen haben, müssen Sie zunächst dafür sorgen das Verwaltungsskripte der DBT automatisch geladen werden.

R versucht nach jedem Neustart zuerst die Datei „.Rprofile“ im Library Ordner des Nutzers auszuführen. Welcher dieser Ordner für Sie ist, können Sie in R mittels Sys.getenv ("HOME") herausfinden. Kopieren Sie die Datei „.Rprofile“ aus dem Ordner PUB\8Rservice an diesen Ort. Passen Sie in der Datei den String für die Variable „.pfad“ an, so dass diese auf den DBT Ordner Ihres Systems zeigt.

Nun sollte sich die DBT bei jedem Neustart melden (zunächst mit der Nachricht das die Pakete noch nicht installiert sind).

# DBT Struktur

Die Unterordner in PUB/dbt stellen jeweils den Sourcecode einzelner Packages dar. Die tar.gz Dateien sind unkompilierte komprimierte Formen der Packages welche zur Installation selbiger verwendet werden können.

# DBT Anwendung

Um alle Packages automatisch zu installieren und zu updaten gibt es eigene Funktionen in der DBT.

* installDBTDependencies() installiert automatisch alle Abhängigkeiten unser Pakete von CRAN
* updateDBT() installiert und/oder updated alle Packages der DBT

Viele Probleme lassen sich durch einen Neustart von R sowie dem erneuten ausführen des jeweiligen Befehls lösen. Bei Bedarf können Sie Paket Abhängigkeiten manuell auflösen, indem Sie den install.packages("packageName") Befehl verwenden.

# R Packages

Die Unterordner in PUB/dbt stellen die einzelnen Packages dar. Im obersten Verzeichnis des Packages finden Sie die .rproj Datei, welche das Projekt als Ganzes in RStudio läd. Die DESCRIPTION Datei beschreibt grundsätzliche Informationen des Packages, wie Autoren, Version usw. Die NAMESPACE Datei beschreibt Abhängigkeiten des Packages (welche Pakete müssen vorher geladen werden etc.).

In R Unterordner befinden sich die einzelnen Funktionen bzw. der Code des Packages. Im man Ordner finden Sie die Dokumentationen für die Funktionen im .Rd Format. Weitere Informationen dazu finden Sie hier: <http://r-pkgs.had.co.nz/>

Um ein Package zu packen (die .tar.gz zu generieren) öffnen Sie das Package mittels .rproj Datei und wählen Sie dann oben rechts unter „Build“ die „Build Source Package“ Funktion. A screenshot of a social media post

Description automatically generated

# DBT Konventionen

Jede Funktion in der DBT muss eine Dokumentation in Form von Kommentaren unter dem Funktionskopf enthalten. Die erste Zeile sollte dabei ein Beispielhafter Aufruf sein, welcher das Ergebnis in eine Variable zurückgibt. Danach folgt ein Block für INPUT und einer für OUTPUT. Orientieren Sie sich dabei an diesem Beispiel:

BinProb4Mixtures <- function(Means, SDs, Weights, Breaks, IsLogDistribution = rep(F, length(Means)), LimitsAreFinite=T){

# BinProbabilities <- BinProb4Mixtures(Means, SDs, Weights, Breaks

# calculates the probability of bins/intervals within the dataspace defined by the given breaks between them

# INPUT

# Means[1:m] Means of the GMM Components

# SDs[1:m] Standard Deviations of the GMM Components

# Weights[1:m] Weights of the GMM Components

# Breaks[1:c] Breaks Defining c-1 or c+1 bins (depending on LimitsAreFinite)

# IsLogDistribution If True, the GMM is interpreted as a logarithmic

# LimitsAreFinite If True, there are c+1 Bins, where the first and last bin are of inifinite size

# OUTPUT

# BinProbabilities Probabalities of either c-1 or c+1 bins/intervals (depending on LimitsAreFinite)

# Projekt Struktur

Die einzelnen Ordner Subversion Verzeichnis PRO/Research entsprechen einzelnen Projektarbeiten, welche immer nach dem gleichen Schema aufgebaut sind. 90RawData enthält Datensätze so wie wir sie erhalten und 09Originale enthält dann ihre Entsprechung in unseren Datenformaten. 08AnalyseProgramme enthält Programme welche die Datensätze verarbeiten oder auswerten.

# Datei Formate

Lrn - steht für learn und enthält eine Matrix in welcher die eigentlichen Daten gespeichert sind

Cls - Enthält Klassenzuordnungen für eine lrn Datei.

Wts - Gewichte. Entsprechen dem Grid einer Esom.

Umx - Die berechneten Höhen einer Umatrix (wird auch für die Pmatrix genutzt)

Imx - Maske die zum ausschneiden einer Insel auf einer Umatrix verwendet wird

Bm - Bestmatches. Beispielsweise die Ergebnisse einer Projektion

Weitere Informationen zu den Datei Formaten finden Sie unter PUB/ZfileFormatDocuments

# Erste Hilfe

R Starten

blockDBT()

R neu starten

installDBTDependencies()

updateDBT()

unblockDBT()

R neu starten