Rkundung – Einführung in R Studio

Dominik Bahlburg, Charlotte Kunze

**Übersicht einiger wichtiger R-Befehle**

**Skript**

|  |  |  |
| --- | --- | --- |
| Befehl | Bedeutung | Beispiel |
| setwd() | Arbeitsverzeichnis setzen | setwd("~/Desktop/Uni/MASTER/WS 2018:2019/Rkundung") |
| read\_csv()  read.csv()  # | Einlesen eines Datensatzes als Tibble  Einlesen eines Datensatzes als dataframe  Kommentare schreiben | data <- read\_csv("~/Desktop/Uni/MASTER/WS 2018:2019/Rkundung/Datensatz/rkurs\_data02.csv")  data <- read.csv('..’)  #Skript zur… |
|  |  |  |

**Hilfefunktionen (Session 2)**

|  |  |  |
| --- | --- | --- |
| Befehl | Bedeutung | Beispiel |
| help() | Hilfeseite zu der Funktion in R | help(plot) |
| ? | Hilfeseite zu der Funktion in R | ?plot  read.csv('..’) |

**R-Pakete**

Hiermit können Funktionen geladen und genutzt werden, die nicht in BaseR enthalten sind. Generell gilt, wenn die packages einmal installiert sind, müssen sie nur noch geladen werden am Anfang eines Skriptes.

|  |  |  |
| --- | --- | --- |
| Befehl | Bedeutung | Beispiel |
| install.packages() | R-Pakete installieren | install.packages(‘tidyverse’) |
| library() | R-Pakete laden | library(‘tidyverse’) |

**Rechnen**

|  |  |  |
| --- | --- | --- |
| Befehl | Bedeutung | Beispiel |
| +, -, +, ^, / | Grundlegende Operationen | 5\*4+(3-pi)^4 |
| sum(), prod()  mean()  sd()  sqrt()  round(), floor(), ceiling()  exp() | Summe, Produkt  Mittelwert berechnen  Standard deviation berechnen  Wurzel  Runden, Abrunden, Aufrunden  Exponentialfunktion | sum(1,2) = 3, prod(1,2)=2  v=c(1,2,3,5), mean(v)  v=c(1,2,3,5), sd(v)  sqrt(9)  round(577.89) |
| log() | Logarithmus von x | log(x) |

**Struktur erkunden**

|  |  |  |
| --- | --- | --- |
| Befehl | Bedeutung | Beispiel |
| str() | gibt die Struktur sowie Format der einzelnen variables | str(data) |
| summary()  View() | fasst den Datensatz in der Console  Anzeigen der eingelesenen data in einem neuen tab | summary(data)  View(data) |

**Variablen**

|  |  |  |
| --- | --- | --- |
| Befehl | Bedeutung | Beispiel |
| = oder <- | zuweisen eines Wertes oder Datensatzes | neue\_daten <- data;  a <-1 oder a =1 |
| rm()  rm(list=ls()) | eine Variable löschen  Löschen aller Variablen | rm(a)  rm(list=ls())) |

**Logische Abfragen (Session 3)**

|  |  |  |
| --- | --- | --- |
| Befehl | Bedeutung | Beispiel |
| TRUE, FALSE | wahr, falsch | w=c(T, F, F) |
| == , !=  >, >=  <, <=  &, I  %in% | gleich, ungleich  größer, größer gleich  kleiner, kleiner gleich  und, oder  sowohl als auch | filter(data, region != ‘Irish Sea’)  5 >3  7 < 5  filter(data, region == ‘Irish Sea’ & HAB ==’1’)  filter(data, region %in% c('English Channel', 'Elbe Estuary')) |

**Nützliche Funktionen aus dem tidyverse package**

|  |  |  |
| --- | --- | --- |
| Befehl | Bedeutung | Beispiel |
| filter() | Dient dazu values aus variables zu filter mit Hilfe der logischen Abfragen | filter(data, region == ‘Irish Sea’) |
| subset()  gather()  group\_by()  mutate()  select()  summarise()  distinct()  arrange() | ähnlich der Filterfunktion, erstellt ein subset nach bestimmten Kriterien  Funktion, die Datenstrukturen verändern und tidy machen kann. Anhand von key= kann man spezifizieren wie die neue zusammengefasste Spalte heißen soll, mit values gibt man an, wie sie gefüllt wird. Mit “-“ spezifiziert man, was nicht zusammengefasst werden soll.  gruppieren von Daten, um nach bestimmten variables zusammenfassen zu können  Erstellt neue Variable mit Funktionen bereits existierender Variablen  mit select können einzelne Spalten ausgewählt oder rausgeschmissen werden  fasst die Daten zusammen, ähnlich wie bei mutate können hier mean, sd o.ä. berechnet werden, allerdings wird der Datensatz danach in seiner Größe reduziert.  entfernt die duplicate in einem Datensatz. In unserem df z.B Temperaturduplicate an jeder Station  Sortierung der variables/Spalten (alphabetisch, numerisch auf-/ oder absteigend). automatisch Default: Aufsteigend, mit descendent: absteigend | subset(data, region != ‘Irish Sea’)  Erstellen einer Nutrientspalte aus 3 Einzelspalten:  gather(data, key = ‘nutrients’, values=’concentration’, -region, -time, -HAB, -toxicity, -spec\_biomass)  group\_by(data, region, date)  mutate(data, mean\_si = mean(silicate))  select(data, -time)  summarise(data, mean\_temp = mean(temperature))  distinct(data, region, temperature)  arrange(data, region) |

Ggplot:

Um ggplot und die o.g. Funktionen zu nutzen, sollte tidyverse geladen werden.

Hier gibt es den Schummelzettel dazu: