

# EINFÜHRUNG IN R - ERGEBNISSE EXPORTIEREN

Jan-Philipp Kolb

07 Juni, 2019

# ERSTELLEN EINES BEISPIELDATENSATZES

```
A <- c(1,2,3,4)
B <- c("A","B","C","D")

mydata <- data.frame(A,B)
```

```
mydata
```

| A | B |
|---|---|
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | D |

# ÜBERSICHT DATENIMPORT/-EXPORT

- wenn man weiterhin mit R arbeitet ist `.RData` oder `rds` die beste Wahl:

```
save(mydata, file="mydata.RData")  
saveRDS(mydata, "mydata.rds")
```

- Der Datensatz kann mit `load` importiert werden.

```
load("mydata.RData")  
mydata <- readRDS("mydata.rds")
```

- `saveRDS()` speichert nicht das Objekt und seinen Namen, sondern nur eine Darstellung des Objekts.

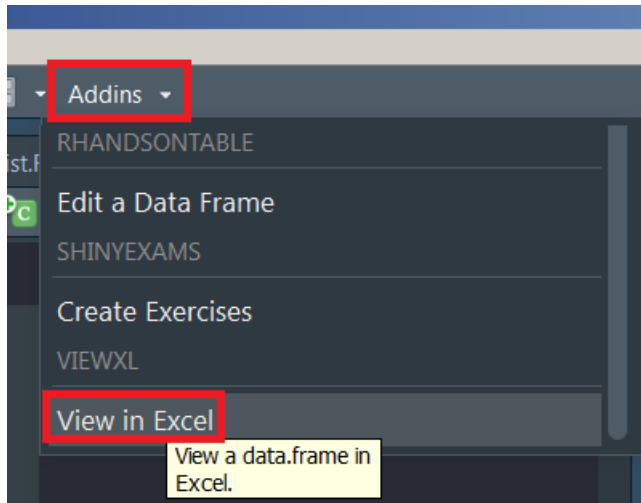
# EXPORT ALS EXCEL-DATENSATZ

```
library(xlsx)
setwd("D:/Daten/GitHub/r_intro_gp18/data")
write.xlsx(mydata,file="mydata.xlsx")
```

# ADDIN UM DATENSATZ IN EXCEL ZU ÖFFNEN

```
devtools::install_github("dreamRs/viewxl")
```

- Wählt man ein `data.frame` im Skript aus wird es in Excel geöffnet.



# DATEN IM .csv FORMAT SPEICHERN

```
write.csv(mydata,file="mydata.csv")
```

- Wenn Sie weiterhin mit deutschem Excel arbeiten wollen, ist es besser, `write.csv2` zu verwenden.

```
write.csv2(mydata,file="mydata.csv")
```

- Ansonsten sieht das Ergebnis so aus:

|   | A        |  |
|---|----------|--|
| 1 | ,"A","B" |  |
| 2 | 1,1,"A"  |  |
| 3 | 2,2,"B"  |  |
| 4 | 3,3,"C"  |  |
| 5 | 4,4,"D"  |  |
| 6 |          |  |

## DEUTSCHE `.csv` FORMATE

- **Verhindern, dass Zeilenamen in eine Datei geschrieben werden, wenn die Datei `write.csv` verwendet wird.**
- In Deutschland ist der Spaltentrenner ein Semikolon, sonst ein Komma

```
write.csv(mydata,file="mydata.csv", row.names=FALSE)
```

- oder für deutsche Daten:

```
write.csv2(mydata,file="mydata.csv", row.names=FALSE)
```

```
install.packages("rio")
```

## Import, Export, and Convert Data Files

The idea behind **rio** is to simplify the process of importing data into R and exporting data from R. This process is, probably unnecessarily, extremely complex for beginning R users. Indeed, R supplies [an entire manual](#) describing the process of data import/export. And, despite all of that text, most of the packages described are (to varying degrees) out-of-date. Faster, simpler, packages with fewer dependencies have been created for many of the file types described in that document. **rio** aims to unify data I/O (importing and exporting) into two simple functions: `import()` and `export()` so that beginners (and experienced R users) never have to think twice (or even once) about the best way to read and write R data.



# DATEN SPEICHERN ALS .sav (SPSS)

```
library("rio")  
# create file to convert  
  
export(mtcars, "data/mtcars.sav")
```

# DATEIFORMATE KONVERTIEREN

```
export(mtcars, "data/mtcars.dta")  
  
# convert Stata to SPSS  
convert("data/mtcars.dta", "data/mtcars.sav")
```

- **Quick R** für den Datenexport



[R Tutorial](#) | [R Interface](#) | [Data Input](#) | [Data Management](#) | [Statistics](#) | [Advanced Statistics](#) | [Graphs](#) | [Advanced Graphs](#)

< [Data Input](#)

Data types

Importing Data

Keyboard Input

## Exporting Data

There are numerous methods for exporting R objects into other formats . For SPSS, SAS and Stata, you will need to load the [foreign](#) packages. For Excel, you will need the [xlsReadWrite](#) package.

- Hilfe zum Export auf dem **CRAN Server**
- **Daten aus R exportieren**

- Laden Sie den Beispieldatensatz `iris`.
- Exportieren Sie den `iris` Datensatzes als Excelfile.