

In Entwicklung

# WhatsAppR

<https://github.com/BorisLS/WhatsAppR>

# WhatsAppR: Die Idee für das Package



- WhatsApp-Gruppe mit Freunden aus der Studienzeit
- Chatverlauf kann als txt-Datei exportiert und per E-Mail verschickt werden
- „Jahresbericht“ für das alljährliche Weihnachtstreffen erstellt, um zu schauen wer wie häufig etwas schreibt
- WhatsAppR stellt die hierbei verwendeten Funktionen für Import und Analyse als Package zur Verfügung
- Zusätzlich die Funktionalitäten auch als Shiny-Web App umgesetzt

# WhatsAppR: Importieren des Chats

- Exportierter Chat muss in ein Dataframe geladen werden → `wapp_import()`



- Struktur der exportierten Zeile ist nicht einheitlich
  - ❖ Zeilen mit Datumsangaben: Statusmeldung oder Nachricht/Medien-Post
  - ❖ Zeilen ohne Datumsangaben: Leerzeile oder nur Inhalt (Datum, Autor identisch)

```
6209 17.12.16, 19:17 - Boris: <Medien weggelassen> 3212 21.12.16, 21:23:05: Manuel: 🍷🍷
6210 17.12.16, 19:20 - Fabian: 👍 🍷 3213 21.12.16, 21:23:35: Manuel: <Bild weggelassen>

6216 18.12.16, 20:20 - Manuel: So Jungs: matchplan für Freitag:
6217
6218 Am Start von der hüffer WG sind: Saxe, boris, Basti, Stack, manolo
```

- Datumsangaben sind unterschiedlich zwischen englischen und deutschen Systemen
- Unterschiede zwischen iOS und Android in der Trennung von Meta-Daten und Inhalt

# WhatsAppR: Importieren

```
test_chat <- wapp_import(history = 'C:/WhatsApp Chat mit Test-Chat.txt',  
                        os = 'android',  
                        language = 'de',  
                        techuser = FALSE)
```

Im Ergebnis ergibt sich nach dem Import das Dataframe `test_chat` mit sieben Spalten:

- `moment`: POSIXct, Zeitpunkt des Postings
  - `day`: Date, Tag des Postings
  - `author`: chr, Name des Autors
  - `content`: chr, Inhalt des Postings
  - `rawdata`: chr, Ursprüngliche exportierte Zeile des Chatverlaufs
  - `message.length`: int, Anzahl der Zeichen der Nachricht
  - `type`: chr. Kennzeichnung, ob es sich um eine Textnachricht oder ob Dateien gepostet wurden
- Funktion ist nicht „eleganter“, sondern „ergebnisorientiert“ .....

# WhatsAppR: Analysieren

- Nach dem Import kann man verschiedene Analysen und Grafiken durchführen



- WhatsAppR stellt verschiedene Analysen als Funktionen bereit
  - ❖ **wapp\_stat\_author** Welches Gruppenmitglied postet wie viel?
  - ❖ **wapp\_stat\_date** An welchen Tagen wird wie viel gepostet?
  - ❖ **wapp\_stat\_time** In welchen Zeitintervallen wird wie viel gepostet?
  - ❖ **wapp\_stat\_firstpost** Welches Gruppenmitglied macht den ersten Post des Tages
  
  - ❖ **wapp\_stat\_emoticons** Welche Emoticons werden von wem am Meisten verwendet?
  - ❖ **wapp\_stat\_word** Wer hat ein bestimmtes Wort verwendet?
  
  - ❖ **wapp\_prep\_author** Vergabe von “neuen” Namen für die Gruppenmitglieder

# WhatsAppR: Next Steps / Weitere Informationen

## Offene Punkte

- Unterstützung von englischen Systemeinstellungen → **Suche nach anonymisierten WhatsApp-Chats!**
- Erweiterung der ShinyApp (z.B. durch Einstellen des Zeitraums)
- Verbesserung Package Abhängigkeiten (Erfahrungen?)
- evtl. automatisches Erkennen von Land und OS
- evtl. CRAN (Erfahrungen?)

## Weitere Informationen

- GitHub:  
<https://github.com/BorisLS/WhatsAppR>
- Vignette:  
[https://borisls.github.io/whatsappr/WhatsAppR\\_Einfuehrung\\_Deutsch.html](https://borisls.github.io/whatsappr/WhatsAppR_Einfuehrung_Deutsch.html)
- Shiny  
<https://borisls.shinyapps.io/WhatsAppR/>

## Kontakt

E-Mail: Boris.Luetke.Schelhowe@gmail.com

Twitter: @JovlerGeek