Schummelzettel: Textverarbeitung mit R und Sonderzeichen

1. Pakete
2.

stringr: Für die meisten Textmanipulationen. **readr:** Für robusten Datenimport, wichtig für korrektes Encoding.

2. Encoding

UTF-8 als Standard verwenden: Sowohl in RStudio als auch beim Datenimport. **Encoding beim Import angeben:** Besonders wichtig bei Dateien von anderen Betriebssystemen.

- Mit locale Funktion im readr Paket:
- library(readr)
- text <- read_csv("datei.csv", locale = locale(encoding = "ISO-8859-1"))
- Mit fileEncoding bei read.table für Windows:
- read.table("Quelle.csv", sep=",", fileEncoding="UTF-8")

Encoding überprüfen: Encoding(text$spalte) **Encoding ändern:** Encoding(text$spalte) <- "latin1" **Probleme erkennen:** Zeichen wie <fc> oder <e4> deuten auf Encoding-Fehler hin.

3. stringr Funktionen

Groß-/Kleinschreibung:

- str_to_upper(text): Alle Buchstaben groß.
- str_to_lower(text): Alle Buchstaben klein.
- str_to_title(text): Erster Buchstabe jedes Wortes groß (Title Case). **Teile extrahieren/ersetzen:**
- str_sub(text, start, end): Extrahiert Teilstring anhand der Position.
- str_detect(text, muster): Prüft auf Muster (TRUE/FALSE).
- str_match(text, muster): Extrahiert Treffer des Musters.
- str_extract(text, muster): Extrahiert den ersten Treffer des Musters.
- str_replace(text, muster, ersatz): Ersetzt den ersten Treffer.
- str_replace_all(text, muster, ersatz): Ersetzt alle Treffer. **Länge und Bereinigung:**
- str_length(text): Gibt die Länge des Textes zurück.
- str_pad(text, länge, seite, zeichen): Füllt Text auf.
- str_trunc(text, länge): Kürzt Text auf eine bestimmte Länge.
- str_squish(text): Entfernt überflüssige Leerzeichen.

4. Reguläre Ausdrücke (Regex)

Definition: Muster zur Textsuche, -prüfung und -manipulation. **Syntax:**

- **Literale Zeichen:** Die meisten Zeichen matchen sich selbst.
- **Sonderzeichen:** Müssen mit Backslash \ escaped werden (z.B. 1 \+ 1 = 2).
- **Zeichenklassen:**
  - [ae] matcht a oder e.
  - `` matcht eine Ziffer.
  - [^x] matcht alles außer x.
- **Kurzformen:**
  - \d: Ziffer (äquivalent zu ``)
  - \w: Wortzeichen (Buchstaben, Zahlen, Unterstrich).
  - \s: Whitespace (Leerzeichen, Tab, Zeilenumbruch).
- **Quantifizierer:**
  - ?: Optional (0 oder 1 mal).
  - *: 0 oder mehr mal.
  - +: 1 oder mehr mal.
  - {n}: Genau n mal.
  - {n,m}: Zwischen n und m mal.
- **Anker:**
  - ^: Anfang des Strings.
  - $: Ende des Strings.
  - \b: Wortgrenze.
- **Alternation:** | (entspricht "oder").
- **Gruppierung:** (muster) (erfasst Treffer in Gruppen).
- **Lookaround:**
  - (?=muster): Positive Lookahead.
  - (?!muster): Negative Lookahead.
  - (?<=muster): Positive Lookbehind.
  - (?<!muster): Negative Lookbehind.

5. Sonderzeichen in Regex

Umlaute: Können direkt verwendet werden, Encoding beachten!
Häufige Probleme
- Umlaute können als `<fc>` oder `<e4>` angezeigt werden
- Dateien mit Umlauten können Probleme beim Einlesen verursachen

Tipps zur Vermeidung
- Immer UTF-8 verwenden für Deutsch
- RStudio-Einstellungen auf UTF-8 setzen
- Überprüfen der Datentypen beim Import
- Pakete wie `readr` für robusteren Datenimport nutzen
- Umlaute in Variablennamen und Dateinamen vermeiden

Verarbeitung von Umlauten: **Encoding-Einstellungen**
- `Sys.setlocale()` für globale Encoding-Einstellungen
- Für Windows: Encoding bei Dateiimporten explizit angeben

```r
read.table("Quelle.csv", sep=",", fileEncoding="UTF-8")
```

Umlauten ersetzen
- Mit `stringr` können Umlaute gleichzeitig ersetzt werden:

```r
library(stringr)
str_replace_all('üäö', c(
'ü' = 'ue',
'ä' = 'ae',
'ö' = 'oe'
))
```

Andere Sonderzeichen:

- Mit Backslash escapen (z.B. \. für einen Punkt).
- In Zeichenklassen verwenden (z.B. [.,!?] für Satzzeichen).
-

regexr.com: Online-Editor zum Testen von Regex.
rdocumentation.org: stri**ngr-Dokumentation**