

Grundläggande strängfunktioner

stringr	base	Användning
str_sub()	substr()	substring, välja ut en del av en sträng
str_c()	paste()	slår ihop strängelement
str_split()	strsplit()	dela upp en sträng i flera element
str_length()	nchar()	beräknar antalet tecken
str_trim()	-	tar bort mellanslag (före/efter textelement)
str_pad()	-	lägger till mellanslag (före/efter textelement)

regular expression: metacharacters

Tecken	Betydelse
.	samtliga tecken (exkl. det "tomma" tecknet "")
^	det "tomma" tecknet i början av en text
\$	det "tomma" tecknet i slutet text
*	föregående tecken 0 eller fler gånger
+	föregående tecken 1 eller fler gånger
?	föregående tecken är valfritt
{n,m}	föregående tecken <i>n</i> eller max <i>m</i>
[]	teckenlista (character list)
	ELLER
()	Gruppering
\	Används för att identifiera metatecken/specialtecken

regular expression: teckenklass

- Med [] skapas en lista över tänkbara tecken, en s.k. teckenklass (eng. character class)
- Används för att identifiera en mängd av tecken
- Inom [] har bara följande meta-tecken en särskild betydelse:

Tecken	Betydelse	Exempel
-	tecken	A-Z a-z 0-9
^	ICKE	^ 0-9
\	specialtecken	\t\n

regular expression: teckenklass

Vanliga fördefinierade klasser är... (kolla ?regexp)

- [:digit:] Nummer
- [:lower:] gemener
- [:upper:] VERSALER
- [:punct:] tecken, ej bokstäver eller siffror
- [:space:] mellanslag, tab, ny rad m.m.

Pattern matchingfunktioner i R

- pattern är ett regular expression

stringr	base	Användning
str_detect()	grepl()	identifierar pattern, returnerar en logisk vektor
str_locate()	gregexpr()	identifierar pattern, returnerar positionen i texten
str_replace()	gsub()	identifierar pattern, och ersätter detta med ny text
str_extract_all	-	Plocka ut alla strängar som uppfyller pattern