# BASE R STRING OPERATIONS CHEATSHEET

As people start realizing the importance of EDA nowadays, R becomes one of the most popular languages in the field of data analysis. However, switching from one language to another is painful in most cases: different syntax, unknown data types, inconsistent functions, etc. In addition to numeric values, **Strings** are the most common data type with which people deal. For this reason, I collect a series of **String Operations** supported by **Base R** into this cheat sheet and use Python as a reference.

## STRING BASICS

### Create

| R | Python |
|---|---|
| **x** <- 'my string' | **x** = 'my string' |

### Measure Length

| R | Python |
|---|---|
| *nchar*(**x**) | *len*(**x**) |

### Access

| R | *substring*(**x**, 1, 1) |
|---|---|
| Python | **x**[0] |
| Find the character in the string **x** by index. | |
| R | *substring*(**x**, 3, 5) |
| Python | **x**[2:5] |
| Find a substring of the string **x** by the start and end indices. | |

## STRING CHECK

### Start & End with

| R | *startsWith*(**x**, 'R') |
|---|---|
| | *endsWith*(**x**, 'R') |
| Python | **x**.*startswith*('R') |
| | **x**.*endswith*('R') |
| Check if a string starts or ends with a certain character. | |

### Contain

| R | Python |
|---|---|
| *grepl*('target', **x**) | 'target' in **x** |
| Check if **x** contains a substring. | |
| !*grepl*('\\D', **x**) | **x**.*isnumeric*() |
| Check if a string is numeric only with regular expressions. | |

## STRING MALNIPULATION

### Clean

| R | Python |
|---|---|
| *trimws*(**x**) | **x**.*strip*() |
| Remove any leading and trailing spaces in the string. | |

### Replace

| R | Python |
|---|---|
| *sub*(old, new, **x**) | **x**.*replace*(old, new) |

### Update

| R | *substring*(**x**, 1, 1) <- 'R' |
|---|---|
| Python | **x**[0] = 'R' |
| Change the character in the string **x** by index. | |
| R | *substring*(**x**, 3, 5) <- 'new' |
| Python | **x** = **x**[:2] + 'new' + x[5:] |
| Change a substring of the string **x** by the start and end indices. | |

### Convert Cases

| R | Python |
|---|---|
| *tolower*(**x**) | **x**.*lower*() |
| *toupper*(**x**) | **x**.*upper*() |
| Convert between lower cases and upper ones. | |

### Split & Concatenate

| R | Python |
|---|---|
| *strsplit*(**x**, split = '-') | **x**.*split*('-') |
| Split **x** into several parts based on a certain separator. | |
| *paste*(**lst_x**, collapse = '-') | '-'.*join*(**lst_x**) |
| Concatenate a list of strings into one by a certain connector. | |