satRday Joburg 2019:

Three Essential REGEX Hacks

“Some people, when confronted with a problem,
think, I know, I’ll use regular expressions.
Now they have two problems.”
- Jamie Zawinskie, August 12, 1997

Marinus Louw

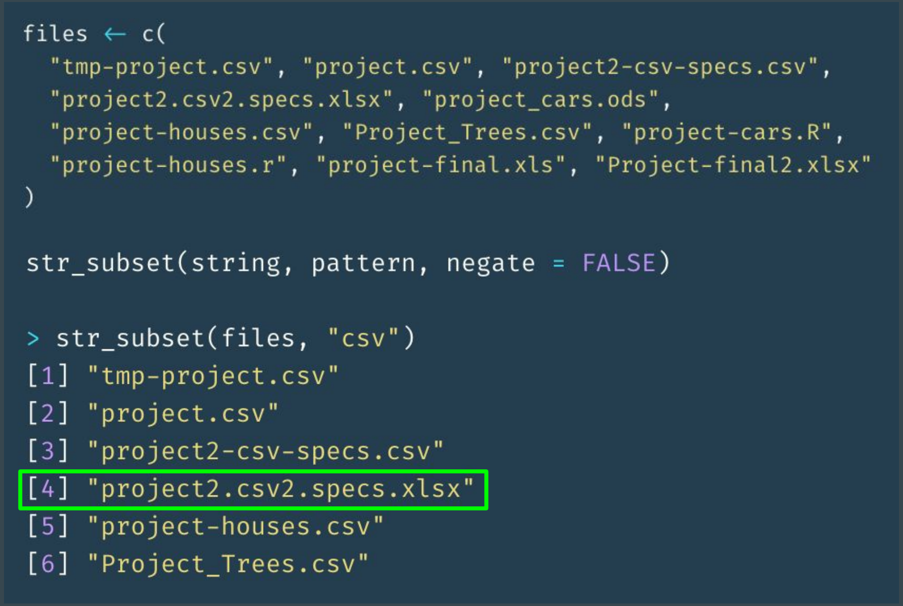
Vanilla Search

Definition:

What are regex?

- Regular expressions are patterns used to find text or find and replace text.

Example:



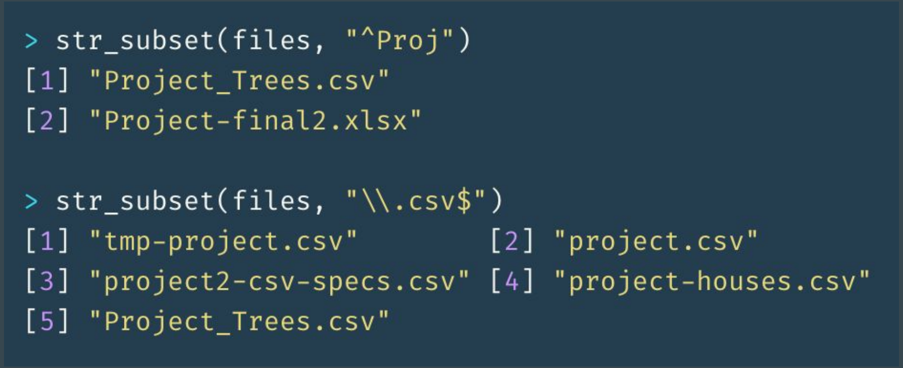
```
files ← c(
  "tmp-project.csv", "project.csv", "project2-csv-specs.csv",
  "project2.csv2.specs.xlsx", "project_cars.ods",
  "project-houses.csv", "Project_Trees.csv", "project-cars.R",
  "project-houses.r", "project-final.xls", "Project-final2.xlsx"
)

str_subset(string, pattern, negate = FALSE)

> str_subset(files, "csv")
[1] "tmp-project.csv"
[2] "project.csv"
[3] "project2-csv-specs.csv"
[4] "project2.csv2.specs.xlsx"
[5] "project-houses.csv"
[6] "Project_Trees.csv"
```

1. Carets and Dollars - ^ and $

Definition:

The ‘^’ and ‘$’ are called anchors and are used to match a **position** before, after or between characters.

Example:



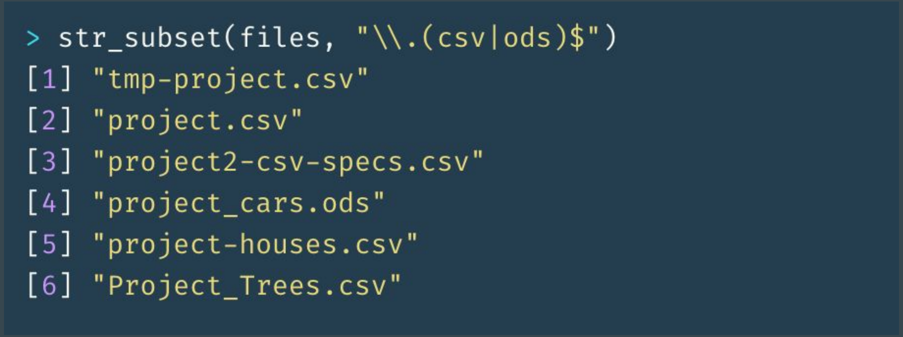
```
> str_subset(files, "^Proj")
[1] "Project_Trees.csv"
[2] "Project-final2.xlsx"

> str_subset(files, "\\.csv$")
[1] "tmp-project.csv"          [2] "project.csv"
[3] "project2-csv-specs.csv" [4] "project-houses.csv"
[5] "Project_Trees.csv"
```

2. Groups and Pipes - ( ) and |

Definition:

‘( )’ and ‘|’ fall under metacharacters. ‘( )’ are used to capture groups, where the order within them matter. ‘|’ allow for alternative matches where you want to specify *x* **or** *y*.

Example:



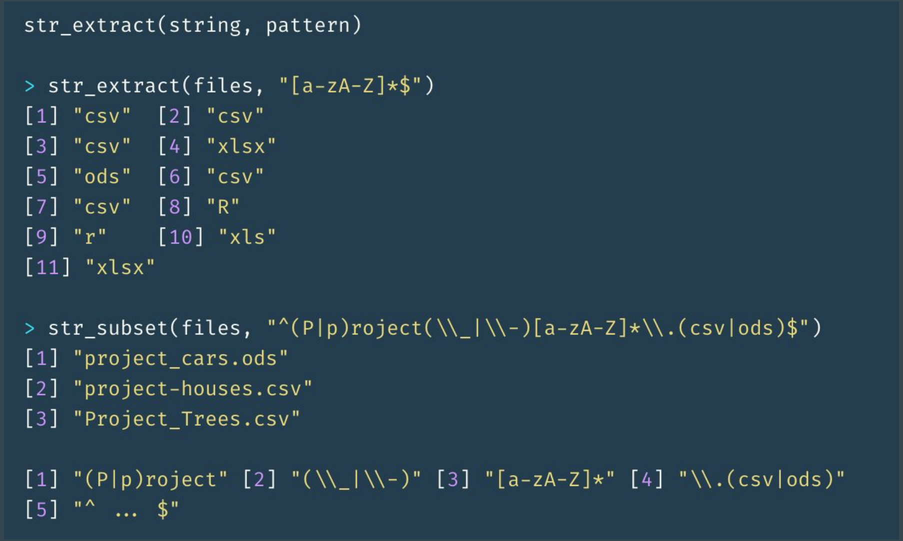
```
> str_subset(files, "\\.(csv|ods)$")
[1] "tmp-project.csv"
[2] "project.csv"
[3] "project2-csv-specs.csv"
[4] "project_cars.ods"
[5] "project-houses.csv"
[6] "Project_Trees.csv"
```

3. Square brackets and Asterisks - [ ] and *

Definition:

‘[ ]’ and ‘*’ are also characterised as meta characters. ‘[ ]’ are used to create a character set where the order of the characters don’t matter. ‘*’ are a type of quantifier as it specifies that the preceding character must match 0 or more times.

Example:



```
str_extract(string, pattern)

> str_extract(files, "[a-zA-Z]*$")
[1] "csv"  [2] "csv"
[3] "csv"  [4] "xlsx"
[5] "ods"  [6] "csv"
[7] "csv"  [8] "R"
[9] "r"    [10] "xls"
[11] "xlsx"

> str_subset(files, "^(P|p)roject(\\_|\\-)[a-zA-Z]*\\.(csv|ods)$")
[1] "project_cars.ods"
[2] "project-houses.csv"
[3] "Project_Trees.csv"

[1] "(P|p)roject" [2] "(\\_|\\-)" [3] "[a-zA-Z]*" [4] "\\.(csv|ods)"
[5] "^ … $"
```

Thank you!

@marinuslouw

EXEGETIC