Lecture Assignment 16

Viraj Vijaywargiya

2022-05-26

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Part 14.2.5

Question 1 The paste() function by default separates strings by space, whereas, the paste0() function does not separate strings by space by default. They are equivalent to the str_c() function, however, it is more like paste0() as it does not separate strings with spaces by default. While the paste() and paste0() functions convert NA to the string "NA" and treat it as a character vector, str_c() returns NA if any argument is a missing value.

Question 3

```
x <- c("Apple", "Banana", "Pear")
len <- str_length(x)
f <- ceiling(len / 2)
str_sub(x, f, f)
```

```
## [1] "p" "n" "e"
```

If the string has an even number of characters, I choose ceiling(length/2) as it also accounts for cases when the length of string is one.

Question 4

The str_wrap() function wraps text, fitting within a certain width. We might want to use this function when wrapping long texts to be typeset.

Part 14.3.1.1

Question 1

"" will escape the next character in R string."\" will resolve to in regular expression, escaping the next character in the regular expression. For "\", the first 2 \ will resolve to in regular expression, and the third escaping the next character.

Question 2

```
str_view("\"'\\", "\"'\\\\", match = TRUE)
```

Part 14.3.2.1

Question 1

```
str_view(c("$^$", "ab$^$sfas"), "^\\$\\^\\$$", match = TRUE)
```

Question 2

1) Words starting with "y",

```
str_view(stringr::words, "^y", match = TRUE)
```

2) Words ending with "x",

```
str_view(stringr::words, "x$", match = TRUE)
```

3) Words that are exactly three letters long,

```
str_view(stringr::words, "^...$", match = TRUE)
```

4) Words having seven letters or more,

```
str_view(stringr::words, ".......", match = TRUE)
```