

Assessment: String Processing Part 1

Question 1

1/1 point (graded)

Which of the following is NOT an application of string parsing?

- ☐ Removing unwanted characters from text.
- ☐ Extracting numeric values from text.
- ☒ Formatting numbers and characters so they can easily be displayed in deliverables like papers and presentations. ✓
- ☐ Splitting strings into multiple values.

Answer

Correct:

Formatting text and numbers for deliverables is not an application of string parsing. String parsing is used as part of the data wrangling process.

Submit

You have used 1 of 2 attempts

i Answers are displayed within the problem

Question 2

1/1 point (graded)

Which of the following commands would not give you an error in R?

- ☒ `cat(" LeBron James is 6'8\" ")` ✓
- ☐ `cat(' LeBron James is 6'8" ')`

☐ `cat(` LeBron James is 6'8" `)`

☐ `cat(" LeBron James is 6\'8" ")`

Answer

Correct:

This would correctly print out your string. Because the string is enclosed in double quotes, (""), you must use an escape character before the inches symbol (").

Submit

You have used 1 of 2 attempts

 Answers are displayed within the problem

Question 3

1/1 point (graded)

Which of the following are advantages of the stringr package over string processing functions in base R? Select all that apply.

☐ Base R functions are rarely used for string processing by data scientists so it's not worth learning them.

☒ Functions in stringr all start with "str_", which makes them easy to look up using autocomplete. ✓

☒ Stringr functions work better with pipes. ✓

☒ The order of arguments is more consistent in stringr functions than in base R. ✓



Submit

You have used 2 of 2 attempts

 Answers are displayed within the problem

Question 4

1/1 point (graded)

You have a dataframe of monthly sales and profits in R

```
> head(dat)
# A tibble: 5 x 3
  Month      Sales    Profit
<chr>    <chr>    <chr>
January  $128,568  $16,234
February $109,523  $12,876
March    $115,468  $17,920
April    $122,274  $15,825
May      $117,921  $15,437
```

Which of the following commands could convert the sales and profits columns to numeric? Select all that apply.



```
dat %>% mutate_at(2:3, parse_number)
```



```
dat %>% mutate_at(2:3, as.numeric)
```



```
dat %>% mutate_all(parse_number)
```



```
dat %>% mutate_at(2:3, funs(str_replace_all(., c("\\$|,", ""))) %>%
  mutate_at(2:3, as.numeric))
```



Answer

Correct:

You can use the `parse_number` command to remove all non-numeric characters. Combining this with the `mutate_at` command allows you to reformat column two and three (Sales and Profit).

You can use the `str_replace_all` command to replace both the "\$" and "," characters, by specifying these in the "pattern" argument of the command. Combining this function with the `mutate_at` command allows you to reformat both column two and three (Sales and Profit). You then need to use the "as.numeric" command to convert these columns from character strings to numbers.

Submit

You have used 1 of 2 attempts



Answers are displayed within the problem

