

Pipe operator

12/05/2020

Necessary packages

```
library(magrittr)
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.0      v purrr  0.3.4
## v tibble  3.0.1      v dplyr  0.8.5
## v tidyr   1.0.3      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.5.0

## -- Conflicts ----- tidyverse_conflicts() --
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
```

```
library(stringr)
```

1. Explore the outputs of the following functions.

```
x <- c(1,2)
sum(x, 3)
```

```
[1] 6
```

```
x %>% sum(3)
```

```
[1] 6
```

```
sum(x, 3) == x %>% sum(3)
```

```
[1] TRUE
```

```
seq(3, 10, 2)
```

```
[1] 3 5 7 9
```

```
3 %>% seq(10, 2)
```

```
[1] 3 5 7 9
```

```
10 %>% seq(3, ., 2)
```

```
[1] 3 5 7 9
```

```
# method 1  
filter(iris, Sepal.Length >= 7.0)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	7.0	3.2	4.7	1.4	versicolor
2	7.1	3.0	5.9	2.1	virginica
3	7.6	3.0	6.6	2.1	virginica
4	7.3	2.9	6.3	1.8	virginica
5	7.2	3.6	6.1	2.5	virginica
6	7.7	3.8	6.7	2.2	virginica
7	7.7	2.6	6.9	2.3	virginica
8	7.7	2.8	6.7	2.0	virginica
9	7.2	3.2	6.0	1.8	virginica
10	7.2	3.0	5.8	1.6	virginica
11	7.4	2.8	6.1	1.9	virginica
12	7.9	3.8	6.4	2.0	virginica
13	7.7	3.0	6.1	2.3	virginica

```
# method 2 - using pipe  
iris %>% filter(Sepal.Length >= 7.0)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	7.0	3.2	4.7	1.4	versicolor
2	7.1	3.0	5.9	2.1	virginica
3	7.6	3.0	6.6	2.1	virginica
4	7.3	2.9	6.3	1.8	virginica
5	7.2	3.6	6.1	2.5	virginica
6	7.7	3.8	6.7	2.2	virginica
7	7.7	2.6	6.9	2.3	virginica
8	7.7	2.8	6.7	2.0	virginica
9	7.2	3.2	6.0	1.8	virginica
10	7.2	3.0	5.8	1.6	virginica
11	7.4	2.8	6.1	1.9	virginica
12	7.9	3.8	6.4	2.0	virginica
13	7.7	3.0	6.1	2.3	virginica

```
# method 1  
ir <- as_tibble(iris)  
select(ir, Species)
```

```
# A tibble: 150 x 1  
  Species
```

```

      <fct>
1 setosa
2 setosa
3 setosa
4 setosa
5 setosa
6 setosa
7 setosa
8 setosa
9 setosa
10 setosa
# ... with 140 more rows

```

```

# method 2 - using pipe
iris %>% as_tibble() %>% select(Species)

```

```

# A tibble: 150 x 1
  Species
  <fct>
1 setosa
2 setosa
3 setosa
4 setosa
5 setosa
6 setosa
7 setosa
8 setosa
9 setosa
10 setosa
# ... with 140 more rows

```

2. Write the following code using the pipe operator.

```

str_c("good", sample(c("health", "food", "work", "day"), 1))

```

```
[1] "goodday"
```

What does the function `str_c` do?

Joins two or more vectors element-wise into a single character vector.

Answer:

```

c("health", "food", "work", "day") %>%
  sample(1) %>%
  str_c("good", .)

```

```
[1] "goodhealth"
```

3. Write the following code using the pipe operator.

```
summarize(filter(iris, Species=="setosa"), median(Sepal.Length))
```

```
  median(Sepal.Length)  
1                5
```

Answer:

```
iris %>%  
  filter(Species=="setosa") %>%  
  summarize(median(Sepal.Length))
```

```
  median(Sepal.Length)  
1                5
```