Statistics with Recitation: TA Session

Danny Po-Hsien Kang (康柏賢)

November 11, 2025

Today's agenda

1. Data Wrangling
   - `substring()`
   - More on `as.Date()`
   - `seq()`
   - `na.omit()`
   - `grepl()`

Trim String: substring()

- **Syntax:**

```
substring(str, start_index, end_index)
```

- **Example:**

```
str1 <- "Cat_and_Dog"
substr1 <- substring(str1, 1, 3)
print(substr1)
```

- **Output:**

```
[1] "Cat"
```

More on as.Date()

- Remember as.Date() turn specific type of character into date type.
- as.Date() can handle any format that matches the string exactly.

```
as.Date("2025/11/11",  "%Y/%m/%d")
as.Date("2025 11 11",  "%Y %m %d")
as.Date("2025abc11xyz11", "%Yabc%mxzy%d")
```

- You can do arithmetics on date data.

```
d1 <- as.Date("2025-11-11")
d2 <- as.Date("2025-11-20")

d2 - d1 # Time difference of 9 days
d1 + 30 # "2025-12-11"
```

Create Vector in Sequence: seq()

- **Syntax:**

```
seq(from, to, by)
```

- **Example:**

```
seq(0, 100, 7)
```

- **Output:**

```
[1]  0  7 14 21 28 35 42 49 56 63 70 77 84 91 98
```

Create Vector in Sequence: seq()

- seq() also works on Date data.
- **Example:**

```
begin_date <- as.Date("2025-01-01")
end_date <- as.Date("2025-12-31")

# Can also use: "day", "week", "year", "2 months"
print(seq(begin_date, end_date, "1 months"))
```

- **Output:**

```
 [1] "2025-01-01" "2025-02-01" "2025-03-01"
 [4] "2025-04-01" "2025-05-01" "2025-06-01"
 [7] "2025-07-01" "2025-08-01" "2025-09-01"
[10] "2025-10-01" "2025-11-01" "2025-12-01"
```

Remove Missing Values: na.omit()

- na.omit() removes NAs in vectors or dataframes.
- **Syntax:**

```
data_clean <- na.omit(data)
```

- **Example (vector):**

```
x <- c(1, NA, 3, NA, 5)
x_clean <- na.omit(x)
print(x_clean)
```

- **Output:**

```
# keeps an attribute (extra information attached to object)
[1] 1 3 5
attr(,"na.action") # you can use as.vector() to remove these
[1] 2 4
attr(,"class")
[1] "omit"
```

Remove Missing Values: na.omit()

- **Example (dataframe):**

```
df <- data.frame(
    x = c(1, NA, 3, 4, NA),
    y = c(10, 20, NA, 40, 50)
    ) %>%
    na.omit()
print(df)
```

- **Output:**

```
  x  y
1 1 10
4 4 40
```

- Sometimes using na.omit() may be too aggressive...

Find patterns inside element: grepl()

- **Syntax:**

```
grepl(pattern, str)
grepl(pattern, str_vec)
```

- grepl() will return a logical vector: TRUE if the pattern is inside that element, FALSE otherwise.
- **Example:**

```
grepl("app", "apple") # TRUE

fruits <- c("apple", "banana", "pineapple", "pear")
grepl("apple", fruits) # TRUE FALSE TRUE FALSE

strings <- c("abcba", "acbcb", "abbbc", "babcc")
grepl("abc", strings) # TRUE FALSE FALSE TRUE
```