In-class Assignment 10

Andrew Shao

2024-11-06

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
df_a <- tibble(id = c(1, 2, 3), value = c("X", "Y", "Z"))
df_b <- tibble(id = c(2, 3, 4), value = c("Y", "Z", "W"))
```

7.8.8 Q1

```
intersect(df_a, df_b)
```

```
## # A tibble: 2 x 2
##      id value
##   <dbl> <chr>
## 1     2 Y
## 2     3 Z
```

7.8.8 Q2

```
union(df_a, df_b)
```

```
## # A tibble: 4 x 2
##      id value
##   <dbl> <chr>
## 1     1 X
## 2     2 Y
## 3     3 Z
## 4     4 W
```

```
union_all(df_a, df_b)
```

```
## # A tibble: 6 x 2
##      id value
##   <dbl> <chr>
## 1     1 X
## 2     2 Y
## 3     3 Z
## 4     2 Y
## 5     3 Z
## 6     4 W
```

`union_all` includes duplicate rows while `union` doesn't.

7.8.8 Q3

```
setdiff(df_a, df_b)
```

```
## # A tibble: 1 x 2
##      id value
##   <dbl> <chr>
## 1     1 X
```

```
setdiff(df_b, df_a)
```

```
## # A tibble: 1 x 2
##      id value
##   <dbl> <chr>
## 1     4 W
```

7.8.8 Q4

```
union(setdiff(df_a, intersect(df_a, df_b)), setdiff(df_b, intersect(df_a, df_b)))
```

```
## # A tibble: 2 x 2
##      id value
##   <dbl> <chr>
## 1     1 X
## 2     4 W
```

```
symdiff(df_a, df_b)
```

```
## # A tibble: 2 x 2
##      id value
##   <dbl> <chr>
## 1     1 X
## 2     4 W
```

7.8.8 Q5

```
df_c <- tibble(id = c('a', 'b', 'c'), value = c(1, 2, 3))
df_d <- tibble(id = c('c', 'a', 'b'), value = c(3, 1, 2))
setequal(df_c, df_d)
```

```
## [1] TRUE
```