Lecture 6

2023-08-09

Exercises

Write a function that computes the second smallest element of a vector.

```
second_smallest <- function(v){
  sort(v)[2]
}
second_smallest(c(5, 4, 3, 2, 1))
```

```
## [1] 2
```

Write a function that computes the second largest element of a vector.

```
second_largest <- function(v){
  sort(v, decreasing=TRUE)[2]
}
second_largest(c(1, 2, 3, 4, 5))
```

```
## [1] 4
```

Write a function that takes in a data frame with a column "temperature (Celsius)" and a column "city", and computes a new data frame with a new column "temperature (Fahrenheit).

```
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   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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 errors
```

```
fahrenheit <- function(df){
  mutate(df, Fahrenheit = Celsius * 1.8 + 32)
}
input_data <- data.frame(
  city = c("New York", "New York", "Los Angeles", "Los Angeles", "Chicago", "Chicago"),
  `Celsius` = c(20, 32, 14, 26, 25, 15)
)
fahrenheit(input_data)
```

```
##          city Celsius Fahrenheit
## 1    New York      20       68.0
## 2    New York      32       89.6
## 3 Los Angeles      14       57.2
```

```
## 4 Los Angeles      26       78.8
## 5     Chicago      25       77.0
## 6     Chicago      15       59.0
```

Write a function that takes in a data frame as in the previous problem and computes the second-smallest temperature in the data frame.

```
extract_second_smallest <- function(df){
  second_smallest(df$Celsius)
}
extract_second_smallest(input_data)
```

```
## [1] 15
```

Write a function that takes in a data frame as in the previous problem and computes the average temperature for each city.

```
average_temp <- function(df){
  df %>% group_by(city) %>% summarize(AvgTemp = mean(Celsius))
}
average_temp(input_data)
```

```
## # A tibble: 3 x 2
##   city        AvgTemp
##   <chr>         <dbl>
## 1 Chicago          20
## 2 Los Angeles      20
## 3 New York         26
```

Write a function that takes in a data frame as in the previous problem and computes the city where the coldest day was recorded.

```
coldest <- function(df){
  (df %>% group_by(city) %>% summarize(MinTemp = min(Celsius)) %>% arrange(MinTemp))$city[1]
}
coldest(input_data)
```

```
## [1] "Los Angeles"
```

Make a new data frame which contains the increase in life expectancy per year for each country in gapminder. The increase per year is the difference between the life expectancy in the last year and the first year, divided by the number of years.

```
library(gapminder)
ret <- gapminder %>% group_by(country) %>% arrange(year) %>% mutate(IncreasePerYear = (lifeExp[length(li
```