

# SAL 413 HW5

Hunter Geise

2023-11-09

## R Markdown

This homework will work on data manipulation primarily using dplyr and lubridate, but other tidyverse packages may also be used. Do not use any Base R functions to work with any Date data. Only rely on lubridate functions (lubridate has the function `as_date()` if needed). Each question will build on the results from the previous question. Read each question carefully as steps to solving the question may be laid out in the question itself. Download the `Fixed_CFB_Scores_21.csv` file from Blackboard (this is a sizeable file and may take time to download and/or load into your R session).

```
library(tidyverse)
library(dplyr)
library(lubridate)
theme_set(theme_bw())

data <- read_csv("cfb_scores.csv")

data %>%
  select(-c(2, 3, 4, 8, 10, 11, 17, 18, 19, 20,
            21, 22, 23, 24, 25, 26, 27, 28, 29,
            30, 31, 32, 36, 37, 38, 39, 40, 41,
            42, 43, 44, 47, 48, 49, 50, 51, 52,
            53, 54, 55, 56, 57, 58, 59)) -> data
```

1. Using lubridate and other tidyverse packages, create a logical variable in the data that determines if each observation occurred between 12/17/21 and 1/4/22 (these are the dates bowl games were played). Then limit the data to only contain observations where this variable is true.

```
data %>%
  mutate(is_bowl_game = as.Date(wallclock) %within%
         (ymd("21/12/17") %--% ymd("22-1-5"))) -> data

data %>%
  filter(is_bowl_game == TRUE) -> bowl_game

bowl_game %>%
  select(c(game_id, homeScore, awayScore, wallclock,
          type.text, is_bowl_game)) -> new_bowl_game

new_bowl_game
```

```
## # A tibble: 6,455 x 6
##   game_id homeScore awayScore wallclock      type.text is_bowl_game
##   <dbl>     <dbl>     <dbl> <dtm>         <chr>         <lgl>
## 1 401331235         0         0 2021-12-31 20:40:52 Kickoff    TRUE
## 2 401331235         0         0 2021-12-31 20:40:52 Rush       TRUE
## 3 401331235         0         0 2021-12-31 20:42:02 Rush       TRUE
## 4 401331235         0         0 2021-12-31 20:42:29 Rush       TRUE
## 5 401331235         0         0 2021-12-31 20:43:19 Rush       TRUE
## 6 401331235         0         0 2021-12-31 20:43:58 Rush       TRUE
## 7 401331235         0         0 2021-12-31 20:44:38 Rush       TRUE
## 8 401331235         0         0 2021-12-31 20:44:54 Rush       TRUE
## 9 401331235         0         0 2021-12-31 20:45:18 Rush       TRUE
## 10 401331235         0         0 2021-12-31 20:45:43 Penalty  TRUE
## # i 6,445 more rows
```

2. For each game in the data, find whether the “home” team won the game by checking the final scores for each game. What proportion of bowl games did the “home” team win?

```
new_bowl_game %>%  
  filter(type.text == "End of Game") -> game_over  
  
game_over %>%  
  summarise(home_win = homeScore > awayScore) -> hw
```

```
## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in  
## dplyr 1.1.0.  
## i Please use 'reframe()' instead.  
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()'   
##   always returns an ungrouped data frame and adjust accordingly.  
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was  
## generated.
```

```
mean(hw$home_win)
```

```
## [1] 0.5142857
```

3. Create a dataset with variables `game_id`, `home_team`, `away_team`, and `total_points` that lists the total points scored in the two bowl games with the most combined points scored by the two teams and the fewest points scored. The end result will have 4 columns and just 2 rows: 1 row for the game with the most points scored and 1 row for the game with the fewest points scored.

```
bowl_game %>%
  filter(type.text == "End of Game") %>%
  mutate(total_points = awayScore + homeScore) -> points

points %>%
  select(c(game_id, home_team, away_team, total_points)) %>%
  tibble() -> points

points %>%
  arrange(total_points) -> lowest_pts

lowest_pts %>% slice(1) -> lowest_pts

points %>%
  arrange(desc(total_points)) -> highest_pts

highest_pts %>% slice(1) -> highest_pts

final_points_table <- bind_rows(highest_pts, lowest_pts)

final_points_table
```

```
## # A tibble: 2 x 4
##   game_id home_team away_team total_points
##   <dbl>   <dbl>   <dbl>   <dbl>
## 1 401331166     2026       98       97
## 2 401331224      135      277      24
```