Homework 8

Ocean Fan

```r
library(dplyr)
```

```

   'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.0     v readr     2.1.5
v ggplot2   4.0.0     v stringr   1.5.2
v lubridate 1.9.4     v tibble    3.3.0
v purrr     1.1.0     v tidyr     1.3.1

-- 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 becom
```

```r
library(rvest)
```

```
    'rvest'

The following object is masked from 'package:readr':

    guess_encoding
```

1.

```r
compute_sumsq <- function(n){
  sum <- 0
  for (i in 1:n) {
    sum <- sum + i^2
  }
  return(sum)
}
compute_sumsq(15)
```

```
[1] 1240
```

```r
compute_sumsq(27)
```

```
[1] 6930
```

2.

a.

```r
scrape_bomojo <- function(url){
  movie_table <- read_html(url) |>
  html_elements("table") |>
  html_table() |>
  pluck(1) |>
  mutate(
    Gross = parse_number(Gross),
    Theaters = parse_number(Theaters),
    `Total Gross` = parse_number(`Total Gross`)
  ) |>
  select(-Genre, -Budget, -`Running Time`, -Estimated)
  movie_table <- janitor::clean_names(movie_table)
  df <- data.frame(movie_table)
  return(df)
}
scrape_bomojo("https://www.boxofficemojo.com/year/2024/")|>
  head()
```

```
  rank                 release     gross theaters total_gross release_date
1    1            Inside Out 2 652980194     4440   652980194       Jun 14
2    2    Deadpool & Wolverine 636745858     4330   636745858       Jul 26
3    3                  Wicked 432943285     3888   473231120       Nov 22
4    4                 Moana 2 404017489     4200   460405297       Nov 27
5    5       Despicable Me 4 361004205     4449   361004205        Jul 3
6    6 Beetlejuice Beetlejuice 294100435     4575   294100435        Sep 6
                           distributor
1 Walt Disney Studios Motion Pictures
2 Walt Disney Studios Motion Pictures
3                   Universal Pictures
4 Walt Disney Studios Motion Pictures
5                   Universal Pictures
6                         Warner Bros.
```

the release date part is a bit easier to apply in part b so I took it off in part a. If we have to do it, then we will do a regex to filter out the only numeric part, which is year, from the url string. b.

```
scrape_bomojo2 <- function(url){
  url <- str_glue("https://www.boxofficemojo.com/year/{url}")
  movie_table <- read_html(url) |>
  html_elements("table") |>
  html_table() |>
  pluck(1) |>
  mutate(
    Gross = parse_number(Gross),
    Theaters = parse_number(Theaters),
    `Total Gross` = parse_number(`Total Gross`),
    `Release Date` = lubridate::mdy(str_glue("{`Release Date`} {url}"))
  ) |>
  select(-Genre, -Budget, -`Running Time`, -Estimated)
  movie_table <- janitor::clean_names(movie_table)
  df <- data.frame(movie_table)
  return(df)
}
scrape_bomojo2(2003) |>
  head()
```

```
  rank                                                release     gross
1    1                                           Finding Nemo 339714184
2    2 Pirates of the Caribbean: The Curse of the Black Pearl 305398779
3    3                                    The Matrix Reloaded 281576461
4    4          The Lord of the Rings: The Return of the King 249445927
5    5                                         Bruce Almighty 242829261
6    6                                       X2: X-Men United 214949694
  theaters total_gross release_date                         distributor
1     3425   339714978   2003-05-30 Walt Disney Studios Motion Pictures
2     3416   305413918   2003-07-09 Walt Disney Studios Motion Pictures
3     3603   281576461   2003-05-15                        Warner Bros.
4     3703   377027325   2003-12-17                     New Line Cinema
5     3549   242829261   2003-05-23                  Universal Pictures
6     3749   214949694   2003-05-02               Twentieth Century Fox
```

3.

a.

```
library(nycflights13)
flights <- nycflights13::flights
filter_severe <- function(flight){
  flight|>
    filter(is.na(arr_time) | (dep_delay > 1))
}
flights |> filter_severe()
```

```
# A tibble: 128,787 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     1     1      517            515         2      830            819
 2  2013     1     1      533            529         4      850            830
 3  2013     1     1      542            540         2      923            850
 4  2013     1     1      608            600         8      807            735
 5  2013     1     1      611            600        11      945            931
 6  2013     1     1      613            610         3      925            921
 7  2013     1     1      623            610        13      920            915
 8  2013     1     1      632            608        24      740            728
 9  2013     1     1      644            636         8      931            940
10  2013     1     1      702            700         2     1058           1014
# i 128,777 more rows
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

b.

```
summarize_severe <- function(flight){
  num_cancel <- NA
  num_delay <- NA
  num_cancel <- flight|>
    filter(is.na(arr_time)) |>
    nrow()
  num_delay <- flight|>
    filter(dep_delay > 1) |>
    nrow()
  summary_table <- tibble(num_delay, num_cancel)
  return(summary_table)
}
flights |> group_by(dest) |> summarize_severe()
```

```
# A tibble: 1 x 2
  num_delay num_cancel
      <int>      <int>
1    120382       8713
```

c.

```
filter_severe <- function(flight, hours){
  flight|>
    filter(is.na(arr_time) | (dep_delay > hours))
}
flights |> filter_severe(hours = 2)
```

```
# A tibble: 122,559 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
```

```
 1  2013     1     1      533            529        4     850            830
 2  2013     1     1      608            600        8     807            735
 3  2013     1     1      611            600       11     945            931
 4  2013     1     1      613            610        3     925            921
 5  2013     1     1      623            610       13     920            915
 6  2013     1     1      632            608       24     740            728
 7  2013     1     1      644            636        8     931            940
 8  2013     1     1      709            700        9     852            832
 9  2013     1     1      732            729        3    1041           1039
10  2013     1     1      732            645       47    1011            941
# i 122,549 more rows
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

d.

```
weather <- nycflights13::weather
summarize_weather <- function(weather, temp){
  weather |>
  summarize(
    min = min({{temp}}, na.rm = TRUE),
    mean = mean({{temp}}, na.rm = TRUE),
    max = max({{temp}}, na.rm = TRUE)
    )
}
weather |> summarize_weather(temp)
```

```
# A tibble: 1 x 3
    min  mean   max
  <dbl> <dbl> <dbl>
1  10.9  55.3  100.
```

e.

```
standardize_time <- function(flight, time){
  flight|>
    mutate(new_time = {{time}}%/%60 + {{time}}%/%60/60)
}
flights |> standardize_time(sched_dep_time)
```

```
# A tibble: 336,776 x 20
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     1     1      517            515         2      830            819
 2  2013     1     1      533            529         4      850            830
 3  2013     1     1      542            540         2      923            850
 4  2013     1     1      544            545        -1     1004           1022
 5  2013     1     1      554            600        -6      812            837
 6  2013     1     1      554            558        -4      740            728
 7  2013     1     1      555            600        -5      913            854
 8  2013     1     1      557            600        -3      709            723
 9  2013     1     1      557            600        -3      838            846
10  2013     1     1      558            600        -2      753            745
# i 336,766 more rows
# i 12 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>, new_time <dbl>
```

4.

a.

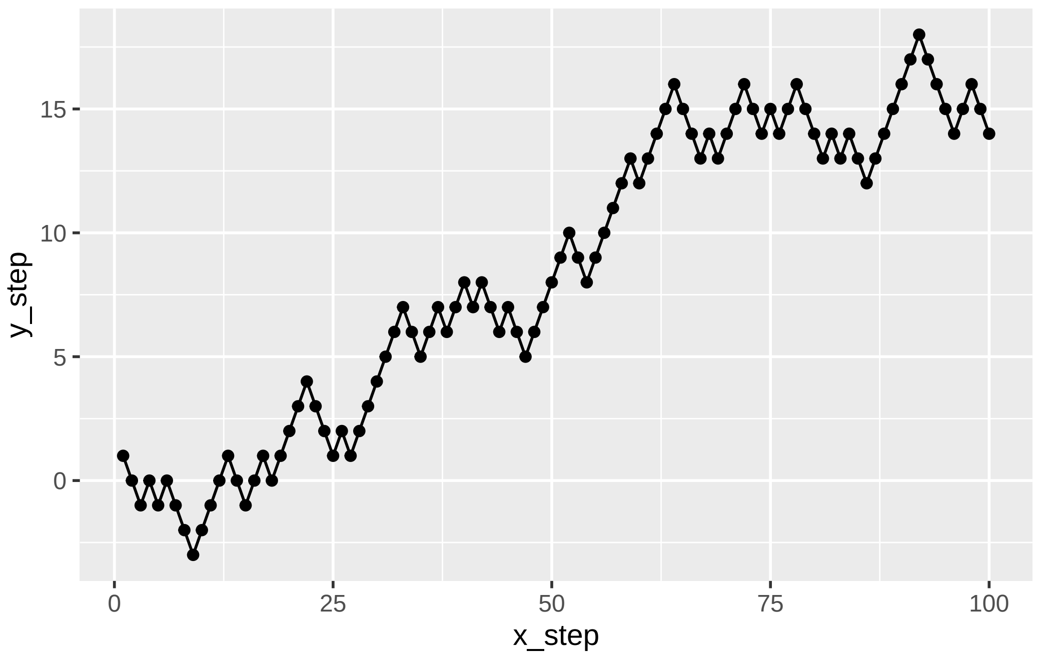
```r
commute <- read.csv("http://aloy.rbind.io/data/CommuteAtlanta.csv")
```

```r
slice_then_mean <- function(commute){
  n_row <- nrow(commute)
  mean <- commute |>
    slice_sample(n = n_row, replace = TRUE) |>
    summarize(mean_time = mean(Time))
  mean <- mean$mean_time[1]
  return(mean)
}

bootstrap_1000 <- vector(length = 1000)

for(i in c(1:1000)){
  bootstrap_1000[i] <- slice_then_mean(commute)
}

quantile(bootstrap_1000, probs = c(0.025, 0.975))
```

```
    2.5%    97.5% 
27.40145 30.86430 
```

5.

```r
x <- 0
y <- 0
x_step <- vector(length = 100)
y_step <- vector(length = 100)
for (i in c(1:100)){
  coin <- sample(c("heads", "tails"), size = 1)
  if (coin == "heads") {
    y <- y + 1
  }
  else{
    y <- y - 1
  }
  x <- x + 1
  x_step[i] <- x
  y_step[i] <- y
}
steps <- data.frame(x_step, y_step)
ggplot(data = steps, aes(x = x_step, y = y_step))+
  geom_point() +
  geom_line()
```


15
10
5
0
y_step
0
25
50
75
100
x_step