# Stats 102A - Homework 3 - Output File

Miles Chen (example)



To receive full credit the functions you write must pass all tests. We may conduct further tests that are not included on this page as well.

## Academic Integrity Statement

By including this statement, I, Yuer Tang, declare that all of the work in this assignment is my own original work.

At no time did I use an AI tool to generate or debug my solutions.

At no time did I look at the code of other students nor did I search for code solutions.

I understand that forbidden AI usage or plagiarism on any single part of this assignment will result in a 0 for the entire assignment.

### Part 1. Basic dplyr exercises

Install the package `fueleconomy` and load the dataset `vehicles`. Answer the following questions.

```
library(fueleconomy) # run install.packages("fueleconomy") if necessary
library(tidyverse)
```

```
Warning: package 'ggplot2' was built under R version 4.4.3
```

```
Warning: package 'tibble' was built under R version 4.4.3
```

```
Warning: package 'tidyr' was built under R version 4.4.3
```

```
Warning: package 'readr' was built under R version 4.4.3

Warning: package 'purrr' was built under R version 4.4.3

Warning: package 'dplyr' was built under R version 4.4.3
```

```r
data(vehicles)
```

a. How many unique vehicle makers (variable `make`) are included in the dataset?

```r
# write your code here, the output displayed should answer the question.
vehicles |> summarize(n = n_distinct(make))
```

```
# A tibble: 1 x 1
      n
  <int>
1   128
```

b. How many vehicles made in 2014 are represented in the dataset?

```r
# write your code here, the output displayed should answer the question.
vehicles |>
  filter(year==2014) |>
  count()
```

```
# A tibble: 1 x 1
      n
  <int>
1  1214
```

c1. For the year 2014, what was the average city mpg (gas mileage) for all `Compact Cars`? What was the average city mpg for `Midsize Cars` in 2014?

```r
# write your code here, the output displayed should answer the question.
vehicles |>
  filter(year == 2014, class=="Compact Cars") |>
  summarise(avg_city_mpg = mean(cty),na.rm = TRUE)
```

```
# A tibble: 1 x 2
  avg_city_mpg na.rm
         <dbl> <lgl>
1         23.9 TRUE
```

c2. For the year 2014, compare makers of `Midsize Cars`.

```
vehicles |>
  filter(year == 2014, class=="Midsize Cars") |>
  summarise(avg_city_mpg = mean(cty),na.rm = TRUE)
```

```
# A tibble: 1 x 2
  avg_city_mpg na.rm
         <dbl> <lgl>
1         23.5 TRUE
```

d. Find the average city mpg of midsize cars for each manufacturer. For example, in 2014, Acura has 5 midsize cars with an average city mpg of 20.6, while Audi has 12 midsize cars with an average city mpg of 19.08. Produce a table showing the city mpg for 2014 midsize cars for the 27 manufacturers represented in the table. Arrange the results in descending order, so that the manufacturer with the highest average mpg will be listed first.

```
# write your code here, the output displayed should answer the question.

vehicles |>
  filter(year == 2014, class == "Midsize Cars")|>
  group_by(make) |>
  summarize(
    avg_city_mpg_make = mean(cty,na.rm=TRUE))|>

  arrange(desc(avg_city_mpg_make))|>
  print(n = Inf, width = Inf)
```

```
# A tibble: 27 x 2
   make          avg_city_mpg_make
   <chr>                     <dbl>
 1 Nissan                     38.9
 2 Toyota                     32.4
 3 Ford                       30.4
 4 Honda                      29.2
```

```
 5 Mazda                 27.6
 6 Hyundai               27.4
 7 Kia                   26.4
 8 Chevrolet             25.6
 9 Lincoln               25.2
10 Volkswagen            24.7
11 Lexus                 21.1
12 Dodge                 21.1
13 Subaru                21
14 Acura                 20.6
15 Buick                 20.2
16 BMW                   20.2
17 Chrysler              20
18 Mercedes-Benz         20
19 Infiniti              19.3
20 Audi                  19.1
21 Volvo                 19
22 Jaguar                16.4
23 Cadillac              16.3
24 Maserati              15
25 Rolls-Royce           13
26 Bentley               11.5
27 Ferrari               11
```

e. Finally, for the years 1994, 1999, 2004, 2009, and 2014, find the average city mpg of `Midsize Cars` for each manufacturer for each year. Use tidyr to transform the resulting output so each manufacturer has one row, and five columns (a column for each year). Print out all the rows of the resulting tibble. You can use `print(tibble, n = 36)` to print 36 rows of a tibble.

```
#             make     1994     1999     2004     2009     2014
# 1          Acura       NA 16.50000 17.33333 17.00000 20.60000
# 2           Audi       NA 15.25000 16.20000 15.83333 19.08333
vehicles |>
  filter(year %in% c(1994, 1999, 2004,2009, 2014), class == "Midsize Cars") |>
  group_by(make,year) |>
  summarise(avg_city_mpg = mean(cty, na.rm = TRUE),.groups = "drop")|>
  pivot_wider(names_from = year, values_from = avg_city_mpg) |>
  print(n=36)
```

```
# A tibble: 36 x 6
   make          `1999` `2004` `2009` `2014` `1994`
```

| | <chr> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> |
|---|---|---|---|---|---|---|
| 1 | Acura | 16.5 | 17.3 | 17 | 20.6 | NA |
| 2 | Audi | 15.2 | 16.2 | 15.8 | 19.1 | NA |
| 3 | BMW | 14.5 | 16.9 | 15.8 | 20.2 | 13.7 |
| 4 | Bentley | 10.5 | 9 | 10 | 11.5 | NA |
| 5 | Buick | 16.5 | 17.3 | 16.5 | 20.2 | 18.2 |
| 6 | Cadillac | 15.3 | 15.8 | 16.2 | 16.3 | 15 |
| 7 | Chevrolet | 17.2 | 18.9 | 20.7 | 25.6 | 16 |
| 8 | Chrysler | 17 | 19.3 | 18.7 | 20 | NA |
| 9 | Daewoo | 18 | 17 | NA | NA | NA |
| 10 | Dodge | 18.8 | 18.8 | 18.7 | 21.1 | 18.8 |
| 11 | Ferrari | NA | NA | 9.5 | 11 | NA |
| 12 | Ford | 16.2 | NA | 18.8 | 30.4 | 16.6 |
| 13 | Honda | 20.2 | 20 | NA | 29.2 | NA |
| 14 | Hyundai | 18.2 | 18.8 | 24.5 | 27.4 | 17.5 |
| 15 | Infiniti | 17.7 | 16.5 | 17.7 | 19.3 | 14.5 |
| 16 | Jaguar | 14.7 | 15.8 | 15.5 | 16.4 | NA |
| 17 | Kia | NA | 19.7 | 22.2 | 26.4 | NA |
| 18 | Lexus | 16.7 | 16.7 | 17.4 | 21.1 | 16 |
| 19 | Lincoln | NA | 16.2 | 17.5 | 25.2 | 16 |
| 20 | Maserati | NA | NA | NA | 15 | NA |
| 21 | Mazda | 19.8 | 18.8 | 19 | 27.6 | 19.2 |
| 22 | Mercedes-Benz | 18.2 | 15 | 16.2 | 20 | NA |
| 23 | Mercury | 16.5 | 17.3 | 18.8 | NA | 17 |
| 24 | Mitsubishi | 18 | 16.8 | 18 | NA | NA |
| 25 | Nissan | 18.5 | 18.4 | 24.4 | 38.9 | 17.3 |
| 26 | Oldsmobile | 17 | NA | NA | NA | 17.8 |
| 27 | Plymouth | 20 | NA | NA | NA | 18.8 |
| 28 | Pontiac | 16.7 | 16.7 | NA | NA | 16 |
| 29 | Rolls-Royce | 10.5 | 11 | 11 | 13 | 9 |
| 30 | Saab | 17.3 | 17.5 | 17.5 | NA | 17 |
| 31 | Saturn | NA | 19.5 | 22.6 | NA | NA |
| 32 | Subaru | NA | NA | NA | 21 | NA |
| 33 | Suzuki | NA | 17 | NA | NA | NA |
| 34 | Toyota | 19 | 25 | 28.4 | 32.4 | 18 |
| 35 | Volkswagen | 17 | 19 | 19 | 24.7 | NA |
| 36 | Volvo | 17.3 | 17.8 | 15.3 | 19 | 17 |

## Part 2. More dplyr

*Make sure your final output shows the desired average number of days between visits.*

```r
load("dr4.Rdata")

# first create a gap table where each row is one customer and each column is the difference l
gap_table <- dr4 |>
  mutate(
    gap12 = visit2 - visit1,
    gap23 = visit3 - visit2,
    gap34 = visit4 - visit3,
    gap45 = visit5 - visit4
  )


# Then average out
overall_avg <- gap_table |>
  summarise(
    avg_days_between_visits = mean(c(gap12, gap23, gap34, gap45), na.rm = TRUE)
  )

print(overall_avg)
```

```
# A tibble: 1 x 1
  avg_days_between_visits
  <drtn>
1 22.4902 days
```

### Part 3. Scrape baseball-reference.com with rvest

```r
library(rvest)
library(polite)

# Open a polite session.
session <- bow("http://www.baseball-reference.com/teams/",
  user_agent = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Ge
  force = TRUE
)

# Scrape the content of the page and store it in an object teampage.
# There's no need to open another session.
teampage <- session |>
  scrape(content = "text/html; charset=UTF-8")
```

```r
# Now that the page content has been scraped, you do not need to request it
# again. Use the object teampage and html_nodes() to extract the desired nodes,
# for example, you'll want to extract the team names among other values.
team_nodes <- teampage |>
  html_nodes("#teams_active .left a")

team_names <- html_text(team_nodes, trim = TRUE)
team_paths <- html_attr(team_nodes, "href")

team_df <- tibble(
  team_name = team_names,
  team_path = team_paths
)


# Write a loop to visit each of the active franchise team pages.
# To change what page you are visiting, use nod("url of updated location")

baseball_list <- vector("list", nrow(team_df))

for (i in seq_len(nrow(team_df))) {

  team_name <- team_df$team_name[i]
  team_path <- team_df$team_path[i]

  team_html <- scrape(nod(session, path = team_path))
  tbl_node <- team_html |> html_node("#franchise_years")

  if (is.na(tbl_node) || length(tbl_node) == 0) {
    baseball_list[[i]] <- NULL
    next
  }

  df <- tbl_node |> html_table(fill = TRUE) |> mutate(across(everything(), as.character))
  df$current_name <- team_name
  baseball_list[[i]] <- df

  Sys.sleep(1)
}

# Combine all the data into a single tibble called baseball that contains all
# of the teams' franchise histories
```

```r
baseball <- bind_rows(baseball_list)

# at the end, be sure to print out the dimensions of your baseball table
dim(baseball)
```

```
[1] 2864   22
```

```r
# Print the first 10 rows of the tibble. Do not use head, as that will truncate
# the table.
print(baseball, n = 10, width = Inf)
```

```
# A tibble: 2,864 x 22
   Year  Tm                   Lg      G     W     L     Ties  `W-L%` `pythW-L%`
   <chr> <chr>                <chr>   <chr> <chr> <chr> <chr> <chr>  <chr>
 1 2025  Arizona Diamondbacks NL West 162   80    82    0     0.494  0.503
 2 2024  Arizona Diamondbacks NL West 162   89    73    0     0.549  0.553
 3 2023  Arizona Diamondbacks NL West 162   84    78    0     0.519  0.491
 4 2022  Arizona Diamondbacks NL West 162   74    88    0     0.457  0.476
 5 2021  Arizona Diamondbacks NL West 162   52    110   0     0.321  0.377
 6 2020  Arizona Diamondbacks NL West 60    25    35    0     0.417  0.458
 7 2019  Arizona Diamondbacks NL West 162   85    77    0     0.525  0.541
 8 2018  Arizona Diamondbacks NL West 162   82    80    0     0.506  0.533
 9 2017  Arizona Diamondbacks NL West 162   93    69    0     0.574  0.594
10 2016  Arizona Diamondbacks NL West 162   69    93    0     0.426  0.424
   Finish   GB    Playoffs          R     RA    Attendance  BatAge PAge  `#Bat`
   <chr>    <chr> <chr>             <chr> <chr> <chr>       <chr>  <chr> <chr>
 1 4th of 5 13.0  ""                791   785   "2,393,973" 27.8   29.8  65
 2 3rd of 5 9.0   ""                886   788   "2,341,876" 28.6   28.6  51
 3 2nd of 5 16.0  "Lost WS (4-1)"   746   761   "1,961,182" 27.4   28.5  54
 4 4th of 5 37.0  ""                702   740   "1,605,199" 26.5   30    57
 5 5th of 5 55.0  ""                679   893   "1,043,010" 28.9   28.5  64
 6 5th of 5 18.0  ""                269   295   ""          29.1   27.7  45
 7 2nd of 5 21.0  ""                813   743   "2,135,510" 28.7   28.6  45
 8 3rd of 5 9.5   ""                693   644   "2,242,695" 29.2   29.6  49
 9 2nd of 5 11.0  "Lost NLDS (3-0)" 812   659   "2,134,375" 28.3   28.7  45
10 4th of 5 22.0  ""                752   890   "2,036,216" 26.7   26.4  50
   `#P`  `Top Player`       Managers           current_name
   <chr> <chr>              <chr>              <chr>
 1 42    G.Perdomo (7.0)    T.Lovullo (80-82)  Arizona Diamondbacks
 2 33    K.Marte (6.8)      T.Lovullo (89-73)  Arizona Diamondbacks
 3 34    C.Carroll (5.3)    T.Lovullo (84-78)  Arizona Diamondbacks
```

```
 4 33    Z.Gallen (5.3)      T.Lovullo (74-88)  Arizona Diamondbacks
 5 41    M.Kelly (2.3)       T.Lovullo (52-110) Arizona Diamondbacks
 6 26    Z.Gallen (2.5)      T.Lovullo (25-35)  Arizona Diamondbacks
 7 27    K.Marte (6.9)       T.Lovullo (85-77)  Arizona Diamondbacks
 8 30    P.Goldschmidt (5.8) T.Lovullo (82-80)  Arizona Diamondbacks
 9 23    Z.Greinke (6.1)     T.Lovullo (93-69)  Arizona Diamondbacks
10 29    J.Segura (6.6)      C.Hale (69-93)     Arizona Diamondbacks
# i 2,854 more rows
```

**Some light text clean up**

```
[1] "Lengths (21, 20) differ (comparison on first 20 components)"
[2] "13 element mismatches"
```

```
[1] TRUE
```

## Part 4. dplyr to summarize the baseball data

```
baseball_summary <- baseball |>
  filter(Year >= 2001, Year <=2025) |>
  mutate(
    W = as.numeric(W),
    L = as.numeric(L),
    R = as.numeric(R),
    RA = as.numeric(RA)
  )|>
  group_by(current_name) |>
  summarise(TW = sum(W, na.rm = TRUE),
            TL= sum(L, na.rm = TRUE),
            TR = sum(R, na.rm = TRUE),
            TRA = sum (RA, na.rm = TRUE),
            win_percent = TW/ (TW+TL),
             .groups = "drop")|>
  arrange(desc(win_percent))

print(baseball_summary, n = 30,  width = Inf)
```

```
# A tibble: 30 x 6
   current_name             TW    TL    TR   TRA win_percent
```

| | <chr> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> |
|---|---|---|---|---|---|---|
| 1 | New York Yankees | 2293 | 1652 | 20042 | 17125 | 0.581 |
| 2 | Los Angeles Dodgers | 2246 | 1702 | 18292 | 15610 | 0.569 |
| 3 | St. Louis Cardinals | 2162 | 1783 | 18325 | 16692 | 0.548 |
| 4 | Boston Red Sox | 2149 | 1798 | 20122 | 18038 | 0.544 |
| 5 | Atlanta Braves | 2133 | 1811 | 18273 | 16596 | 0.541 |
| 6 | Philadelphia Phillies | 2050 | 1897 | 18190 | 17638 | 0.519 |
| 7 | Cleveland Guardians | 2044 | 1901 | 18200 | 17441 | 0.518 |
| 8 | Houston Astros | 2044 | 1902 | 17956 | 17039 | 0.518 |
| 9 | San Francisco Giants | 2036 | 1909 | 17063 | 16708 | 0.516 |
| 10 | Los Angeles Angels | 2024 | 1924 | 17997 | 17773 | 0.513 |
| 11 | Athletics | 2015 | 1932 | 17741 | 17258 | 0.511 |
| 12 | Chicago Cubs | 1996 | 1950 | 17766 | 17139 | 0.506 |
| 13 | Toronto Blue Jays | 1988 | 1959 | 18607 | 18053 | 0.504 |
| 14 | Minnesota Twins | 1985 | 1964 | 18205 | 18209 | 0.503 |
| 15 | Milwaukee Brewers | 1984 | 1964 | 17646 | 17678 | 0.503 |
| 16 | Seattle Mariners | 1974 | 1974 | 17190 | 17420 | 0.5 |
| 17 | Tampa Bay Rays | 1967 | 1980 | 17575 | 17637 | 0.498 |
| 18 | New York Mets | 1965 | 1981 | 17261 | 17167 | 0.498 |
| 19 | Texas Rangers | 1947 | 2002 | 19103 | 19247 | 0.493 |
| 20 | Arizona Diamondbacks | 1917 | 2031 | 17931 | 18333 | 0.486 |
| 21 | San Diego Padres | 1902 | 2047 | 16602 | 17366 | 0.482 |
| 22 | Washington Nationals | 1877 | 2069 | 17108 | 18011 | 0.476 |
| 23 | Chicago White Sox | 1876 | 2072 | 17520 | 18356 | 0.475 |
| 24 | Cincinnati Reds | 1859 | 2089 | 17485 | 18698 | 0.471 |
| 25 | Detroit Tigers | 1849 | 2094 | 17520 | 18815 | 0.469 |
| 26 | Miami Marlins | 1831 | 2114 | 16562 | 18132 | 0.464 |
| 27 | Baltimore Orioles | 1806 | 2140 | 17626 | 19441 | 0.458 |
| 28 | Pittsburgh Pirates | 1769 | 2174 | 16264 | 18468 | 0.449 |
| 29 | Colorado Rockies | 1770 | 2179 | 18808 | 20640 | 0.448 |
| 30 | Kansas City Royals | 1742 | 2206 | 17010 | 19262 | 0.441 |

## 5. Regular expressions to extract values in the Managers Column

```
pattern <- "([A-Z]\\.[a-zA-ZÀ-ÿ' ]+) \\(([0-9]+)-([0-9]+)\\)"
manager_matches <- str_match_all(baseball$Managers, pattern)
manager_matrix <- do.call(rbind, manager_matches)
manager_df <- tibble(
  manager = manager_matrix[, 2],
  wins    = as.numeric(manager_matrix[, 3]),
  losses  = as.numeric(manager_matrix[, 4])
```

```r
)

manager_summary <- manager_df |>
  group_by(manager) |>

  summarize(
    games   = sum(wins + losses),
    wins    = sum(wins),
    losses  = sum(losses),
    win_pct = wins / games,
    .groups = "drop"
  ) |>
  arrange(desc(games))

print(manager_summary, n = 15, width = Inf)
```

```
# A tibble: 616 x 5
   manager    games  wins losses win_pct
   <chr>      <dbl> <dbl>  <dbl>   <dbl>
 1 C.Mack      7679  3731   3948   0.486
 2 T.La Russa  5383  2884   2499   0.536
 3 D.Baker     4824  2602   2222   0.539
 4 B.Bochy     4518  2252   2266   0.498
 5 B.Cox       4505  2504   2001   0.556
 6 B.Harris    4377  2158   2219   0.493
 7 J.McGraw    4373  2583   1790   0.591
 8 J.Torre     4323  2326   1997   0.538
 9 S.Anderson  4028  2194   1834   0.545
10 G.Mauch     3939  1902   2037   0.483
11 T.Francona  3784  2033   1751   0.537
12 C.Stengel   3747  1905   1842   0.508
13 L.Durocher  3717  2008   1709   0.540
14 W.Alston    3653  2040   1613   0.558
15 L.Piniella  3548  1835   1713   0.517
# i 601 more rows
```