Programming in Base R

Task 1

(a)

We cannot use `read_csv()` to read this data because the `read_csv()` function assumes the delimiter is a comma (`,`), but the file uses semicolons (`;`).

```
library(tidyverse)
```

```
Warning: package 'ggplot2' was built under R version 4.3.2
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     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 becom
```

```
# Read in the semicolon-delimited file
data <- read_delim(
  file = "/Users/yuhanhu/Documents/Summer2025/ST558/HW/HW3/Data/data.txt",
  delim = ";"
)
```

```
Rows: 2 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ";"
chr (2):  y,  z
dbl (1): x

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
print(data)
```

```
# A tibble: 2 x 3
      x ` y`  ` z`
  <dbl> <chr> <chr>
1     1 " 2"  " 3"
2     5 " 3"  " 8"
```

(b)

```r
# Read in data2.txt using 6 as the delimiter
data2 <- read_delim(
  file = "/Users/yuhanhu/Documents/Summer2025/ST558/HW/HW3/Data/data2.txt",
  delim = "6",
  col_types = cols(
    x = col_factor(),
    y = col_double(),
    z = col_character()
  )
)

print(data2)
```

```
# A tibble: 3 x 3
  x          y z
  <fct> <dbl> <chr>
1 1          2 3
2 5          3 8
3 7          4 2
```

Task 2

(a)

```
library(tidyverse)
trailblazer <- read_csv("/Users/yuhanhu/Documents/Summer2025/ST558/HW/HW3/Data/trailblazer.cs
```

```
Rows: 9 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (1): Player
dbl (10): Game1_Home, Game2_Home, Game3_Away, Game4_Home, Game5_Home, Game6_...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(trailblazer)
```

```
Rows: 9
Columns: 11
$ Player      <chr> "Damian Lillard", "CJ McCollum", "Norman Powell", "Robert ~
$ Game1_Home  <dbl> 20, 24, 14, 8, 20, 5, 11, 2, 7
$ Game2_Home  <dbl> 19, 28, 16, 6, 9, 5, 18, 8, 11
$ Game3_Away  <dbl> 12, 20, NA, 0, 4, 8, 12, 5, 5
$ Game4_Home  <dbl> 20, 25, NA, 3, 17, 10, 17, 8, 9
$ Game5_Home  <dbl> 25, 14, 12, 9, 14, 9, 5, 3, 8
$ Game6_Away  <dbl> 14, 25, 14, 6, 13, 6, 19, 8, 8
$ Game7_Away  <dbl> 20, 20, 22, 0, 7, 0, 17, 7, 4
$ Game8_Away  <dbl> 26, 21, 23, 6, 6, 7, 15, 0, 0
$ Game9_Home  <dbl> 4, 27, 25, 19, 10, 0, 16, 2, 7
$ Game10_Home <dbl> 25, 7, 13, 12, 15, 6, 10, 4, 8
```

(b)

```
colnames(trailblazer)
```

```
 [1] "Player"      "Game1_Home"  "Game2_Home"  "Game3_Away"  "Game4_Home" 
 [6] "Game5_Home"  "Game6_Away"  "Game7_Away"  "Game8_Away"  "Game9_Home" 
[11] "Game10_Home"
```

```
trailblazer_longer <- trailblazer %>%
  pivot_longer(
    cols = -Player,
    names_to = "Game_Location",
    values_to = "Points"
  ) %>%
  separate(Game_Location, into = c("Game", "Location"), sep = "_")

# Show the first 5 rows
head(trailblazer_longer, 5)
```

```
# A tibble: 5 x 4
  Player         Game  Location Points
  <chr>          <chr> <chr>     <dbl>
1 Damian Lillard Game1 Home         20
2 Damian Lillard Game2 Home         19
3 Damian Lillard Game3 Away         12
4 Damian Lillard Game4 Home         20
5 Damian Lillard Game5 Home         25
```

(c)

```
home_vs_away_summary <- trailblazer_longer %>%
  pivot_wider(
    names_from = Location,
    values_from = Points
  ) %>%
  group_by(Player) %>%
  summarise(
    mean_home = mean(Home, na.rm = TRUE),
    mean_away = mean(Away, na.rm = TRUE),
    diff = mean_home - mean_away
  ) %>%
  arrange(desc(diff))

print(home_vs_away_summary)
```

```
# A tibble: 9 x 4
  Player           mean_home mean_away   diff
  <chr>                  <dbl>     <dbl>  <dbl>
1 Jusuf Nurkic           14.2       7.5   6.67 
2 Robert Covington        9.5       3     6.5  
3 Nassir Little           8.33      4.25  4.08 
4 Damian Lillard         18.8      18     0.833
5 Cody Zeller             5.83      5.25  0.583
6 Larry Nance Jr          4.5       5    -0.5  
7 CJ McCollum            20.8      21.5  -0.667
8 Anfernee Simons        12.8      15.8  -2.92 
9 Norman Powell          16        19.7  -3.67 
```

Task 3

(a)

<NULL>: This means that for a given combination, no data exists, e.g., no penguins of that species were observed on that island.

<dbl [52]>: This means a lost-column containing 52 numeric values was created. It occurred because multiple 'bill_length_mm' values exist for that species/island combo, so R stores them in a list.

<list>: This indicates the column is a list-column. Instead of having one value per cell, the cell contains a list of values, which are often numeric vectors like 'dbl [52]'.

Thus, the warning happens because 'pivot_wider()' expects each combination of species and island to map to a single value of 'bill_length_mm', but multiple rows in the original dataset have the same species and island.

(b)

```r
library(tidyverse)
library(palmerpenguins)

penguins_summary <- penguins %>%
  count(species, island) %>%
  pivot_wider(
    names_from = island,
    values_from = n,
    values_fill = 0  # fill missing combinations with 0
  )
```

```r
print(penguins_summary)
```

```
# A tibble: 3 x 4
  species   Biscoe Dream Torgersen
  <fct>      <int> <int>     <int>
1 Adelie        44    56        52
2 Chinstrap      0    68         0
3 Gentoo       124     0         0
```

Task 4

```r
library(tidyverse)
library(palmerpenguins)

penguins_fixed <- penguins %>%
  mutate(
    bill_length_mm = case_when(
      is.na(bill_length_mm) & species == "Adelie" ~ 26,
      is.na(bill_length_mm) & species == "Gentoo" ~ 30,
      TRUE ~ bill_length_mm
    )
  ) %>%
  arrange(bill_length_mm) %>%
  head(10)

print(penguins_fixed)
```

```
# A tibble: 10 x 8
   species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
   <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
 1 Adelie  Torgersen           26            NA                NA          NA
 2 Gentoo  Biscoe              30            NA                NA          NA
 3 Adelie  Dream               32.1          15.5             188        3050
 4 Adelie  Dream               33.1          16.1             178        2900
 5 Adelie  Torgersen           33.5          19               190        3600
 6 Adelie  Dream               34            17.1             185        3400
 7 Adelie  Torgersen           34.1          18.1             193        3475
 8 Adelie  Torgersen           34.4          18.4             184        3325
 9 Adelie  Biscoe              34.5          18.1             187        2900
```

```
10 Adelie  Torgersen           34.6          21.1               198        4400
# i 2 more variables: sex <fct>, year <int>
```