Using Tidyverse

Calista Harris

Load require libraries

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

Task 1

Question a

We cannot use the `read_csv()` function specifically to read this data because it expects the data to be comma-separated. The data in `data.txt` is separated by a different delimiter (semicolon). `read_csv()` does not allow specifying a different delimiter, so we must use `read_delim()` instead.

```
#read in the data file
data <- read_delim("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.
```

```
#have all variables have type "dbl"
data <- data |>
  mutate(across(everything(), as.double))
```

```
#display the data
data
```

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

Question b

```
#read in the data file
data2 <- read_delim("data/data2.txt", delim = "6",
                    col_types = "fdc") #'f' for fct, 'd' for dbl, 'c' for chr

#display the data
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

Question a

```
#read in the data file
trailblazer <- read_csv("data/trailblazer.csv")
```

```
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 of the data to confirm it was read in correctly
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
```

Question b

```
#pivot the data to long format
trailblazer_longer <- trailblazer |>
  pivot_longer(cols = -Player, #use all columns except Player
               names_to = c("Game", "Location"),
               names_sep = "_",
               values_to = "Points"
               )

#display the first 5 rows with slice()
trailblazer_longer |>
  slice(1:5)
```

```
# 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
```

Question c

```
avg_trailblazer <- trailblazer_longer |>
  pivot_wider(names_from = Location,
              values_from = Points
              ) |> #creates the 90 x 4 tibble
  group_by(Player) |> #turns into 9 x 4 tibble with each row for the 9 players
  summarise(mean_home = mean(Home, na.rm = TRUE),
            mean_away = mean(Away, na.rm = TRUE)
            ) |>
  mutate(difference = mean_home - mean_away) |>
  arrange(desc(difference))

#display the data
avg_trailblazer
```

```
# A tibble: 9 x 4
  Player           mean_home mean_away difference
  <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
```

From the tibble above, player **Jusuf Nurkic** scored more on average at home through the first 10 games of the season than away.

Task 3

Question a

- <NULL>: means that there are no values for that particular cell in the data table where there is a missing or empty entry. This is common in hierarchical data when certain groups don't have data for some variables
- <dbl[52]>: means that the cell contains a list-column with 52 numeric values, indicating repeated or nested data. This is an example of hierarchical data stored in a rectangular format
- <list>: indicates that the cell contains a list-column that could contain any type of object. From other lists, this is common in tibbles when data is too complex or nested to fit into a single vector

Question b

```
penguins |>
  count(species, island) |>
  pivot_wider(names_from = island, values_from = n,
              values_fill = 0 #from tibble, missing combinations are value 0
              ) |>
  group_by(species)
```

```
# A tibble: 3 x 4
# Groups:   species [3]
  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
complete_bill_length <- penguins |>
  mutate(bill_length_mm = if_else(
      is.na(bill_length_mm) & species == "Adelie", 26,
      if_else(is.na(bill_length_mm) & species == "Gentoo", 30, bill_length_mm)
  )) |>
  arrange(bill_length_mm) |>
  slice(1:10)

#display the data
complete_bill_length
```

```
# 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>
```