Programming in Base R

```
library(tidyverse)
```

```
-- 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 becor
```

```
library(palmerpenguins)
```

Task 1

Question 1

```
?read_csv
```

We cannot use read_csv specifically because our delimiter for the given data is a semicolon, not a comma. There are 2 other functions described in the help file called read_csv2 and read_delim which allow for more control over the delimiter, and read_csv2 specifically uses ; as the delimiter which we will use.

```
data <- read_csv2("data/data.txt")
```

```
i Using "',' as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.

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

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.
```

```
data
```

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

Question 2

```
data2 <- read_delim("data/data2.txt", delim = '6', col_types = "fdc")
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 1

```
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(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 2

```
trailblazer_longer <- trailblazer |>
  pivot_longer(cols = 2:11,
               names_to = c('Game', 'Location'),
               names_sep = '_',
               values_to = 'Points')
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
```

Question 3

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

```
# 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
```

Jusuf Nurkic, Robert Covington and Nassir Little were the top three players who scored more points at home than away throughout the first 10 games of the season.

Task 3

Question 1

```
?penguins
penguins |>
  select(species, island, bill_length_mm) |>
  pivot_wider(
    names_from = island, values_from = bill_length_mm
```

```
    )
```

```
Warning: Values from `bill_length_mm` are not uniquely identified; output will contain
list-cols.
* Use `values_fn = list` to suppress this warning.
* Use `values_fn = {summary_fun}` to summarise duplicates.
* Use the following dplyr code to identify duplicates.
  {data} %>%
  dplyr::group_by(species, island) %>%
  dplyr::summarise(n = dplyr::n(), .groups = "drop") %>%
  dplyr::filter(n > 1L)
```

```
# A tibble: 3 x 4
  species   Torgersen  Biscoe      Dream
  <fct>     <list>     <list>      <list>
1 Adelie    <dbl [52]> <dbl [44]>  <dbl [56]>
2 Gentoo    <NULL>     <dbl [124]> <NULL>
3 Chinstrap <NULL>     <NULL>      <dbl [68]>
```

We did not group by the species and island, so there are errors when we run this code since there are multiple values for bill_length_mm. means that there are no penguins for that combination of species and island. <dbl [52]> means that for the Adelie species on Torgersen, there are 52 dbl values, so there are 52 different values of bill_length_mm for that group. means that the column is a list since it is comprised of many dbls or null values.

Question 2

```
penguins |>
  group_by(species, island) |>
  summarise(count = n()) |>
  pivot_wider(names_from = island, values_from = count, values_fill = 0)
```

```
`summarise()` has grouped output by 'species'. You can override using the
`.groups` argument.
```

```
# 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

Question 1

```r
penguins_nona <- penguins |>
  mutate(bill_length_mm = case_when(
    (species == 'Adelie') & is.na(bill_length_mm) ~ 26,
    (species == 'Gentoo') & is.na(bill_length_mm) ~ 30,
    TRUE ~ bill_length_mm
  ))

head(penguins_nona, 10)
```

```
# 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           39.1          18.7               181        3750
 2 Adelie  Torgersen           39.5          17.4               186        3800
 3 Adelie  Torgersen           40.3          18                 195        3250
 4 Adelie  Torgersen           26            NA                  NA          NA
 5 Adelie  Torgersen           36.7          19.3               193        3450
 6 Adelie  Torgersen           39.3          20.6               190        3650
 7 Adelie  Torgersen           38.9          17.8               181        3625
 8 Adelie  Torgersen           39.2          19.6               195        4675
 9 Adelie  Torgersen           34.1          18.1               193        3475
10 Adelie  Torgersen           42            20.2               190        4250
# i 2 more variables: sex <fct>, year <int>
```