

tidyverse

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

Task 1

Question A

```
?read_csv()
```

starting httpd help server ... done

In 1-2 sentences, explain why we can not use specifically the `read_csv()` to read in these data.

`read_csv()` can read in files with comma separated values, and `read_csv2()` can read in files with semicolons as delimiters instead. Because `data.txt` and `data2.txt` contain semicolons and not commas, we must use `read_csv2()` instead of `read_csv()` to read in the data.

```
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 B

```
six <- read_delim("Data/data2.txt",
                  delim = "6",
                  col_types = "fdc")
```

```
six
```

```
# 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

```
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 B

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

head(trailblazer_longer, n=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
```

```
trailblazer_longer # showing that there are 90 rows and 4 cols
```

```
# A tibble: 90 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
6	Damian Lillard	Game6	Away	14
7	Damian Lillard	Game7	Away	20
8	Damian Lillard	Game8	Away	26
9	Damian Lillard	Game9	Home	4
10	Damian Lillard	Game10	Home	25

i 80 more rows

Question C

```

trailblazer_wider <- trailblazer_longer |>
  pivot_wider(names_from = Location,
              values_from = Points) |>
  group_by(Player) |>
  summarize(mean_home = mean(Home, na.rm = T),
            mean_away = mean(Away, na.rm = T),
            points_diff = mean_home - mean_away) |>
  arrange(desc(points_diff))

trailblazer_wider

```

```

# A tibble: 9 x 4
  Player      mean_home mean_away points_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

```

In 1 sentence, state which players scored, on average, more points at home through the first 10 games of the season than away.

On average, player Jusuf Nurkic scored more points at home through the first 10 games of the season than at away games.

Task 3

Question A

Written answer to Task 3, Question A

When trying to pivot the penguins data set wider, there is an error that is thrown due to non-unique values of bill length. This means that there are some data points that are common across penguins/islands and cannot be distinguished from one another, so there is not a way to show the values distinctly with the wider data set.

The NULL column value is showing that the indicated list is empty - there are no data for Gentoo penguins on the islands Torgersen and Dream, for example, but there ARE data for these penguins on island Biscoe (the only island for Gentoo that did not display NULL).

The <dbl [52]> text in the Adelie row, Torgersen column indicates that there is a list of 52 double numeric values present for this species/island pair.

Finally, “list” appears under each island name, indicating that the data type for all these is list (as opposed to double, character, etc.).

Question B

```
penguins |>
  group_by(species, island) |>
  summarize(n = as.numeric(n())) |>
  pivot_wider(names_from = island,
              values_from = n,
              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>      <dbl> <dbl>      <dbl>
1 Adelie      44    56         52
2 Chinstrap    0    68         0
3 Gentoo     124    0         0
```

Task 4

```
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) |>
  print(n = 10)
```

```
# A tibble: 344 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 334 more rows
# i 2 more variables: sex <fct>, year <int>
```