

The Beginnings of Tidyverse

Alise Miller

Task 1

Question a reading in data 1

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.5.2      v tibble     3.2.1
v lubridate  1.9.4      v tidyr      1.3.1
v purrr      1.0.4
-- 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 become
```

```
data_1 <-read_csv( "data/data.txt")
```

```
Rows: 2 Columns: 1
```

```
-- Column specification -----
Delimiter: ","
chr (1): 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_1
```

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

This has a warning, so I immediately felt like I did something wrong. Looking further into the help feature, `read_csv` can't be used because the data was separated with semi-colons, maybe I could try to edit "sep =". So this means the header is messed up along with the actual data presented.

```
data_1a <-read_csv2("data/data.txt") #makes more appropriate
```

```
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_1a
```

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

Question B

```
data_2 <- read_delim("data/data2.txt",
                     col_names =TRUE,
                     delim = "6",
                     col_types ="fdc")
```

```
data_2
```

```
# 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_long <-
  trailblazer %>%
    pivot_longer(cols = starts_with("game"),
                  names_to = "Location1",
                  values_to = "Points")
trailblazer_longer <- separate(trailblazer_long,
                               col= Location1,
                               into = c("Game", "Location"),
                               remove = TRUE)

slice(trailblazer_longer, 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
```

```
trailblazer_longer
```

```
# 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_longer |>
# group_by(Player,Location) |>
# summarize(mean( Points, na.rm = TRUE))
#mutate()
#This is was what I tried to do before reading the bullet points

```

```

trailblazer_wider <- trailblazer_longer |>
pivot_wider(names_from = Location,
             values_from = Points)
trailblazer_wider

```

```

# A tibble: 90 x 4
  Player      Game Home Away
  <chr>      <chr> <dbl> <dbl>
1 Damian Lillard Game1    20    NA
2 Damian Lillard Game2    19    NA
3 Damian Lillard Game3     NA    12
4 Damian Lillard Game4    20    NA
5 Damian Lillard Game5    25    NA
6 Damian Lillard Game6     NA    14
7 Damian Lillard Game7     NA    20
8 Damian Lillard Game8     NA    26
9 Damian Lillard Game9      4     NA
10 Damian Lillard Game10    25     NA
# i 80 more rows

```

```

summary_of_player <- trailblazer_wider |>
group_by(Player) |>
summarise(mean_home = mean(Home, na.rm = TRUE),
           mean_away = mean(Away, na.rm = TRUE))

```

```
)
summary_of_player
```

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

```
summary_of_player |>
  mutate(difference = mean_home - mean_away) |>
  arrange(desc(difference))
```

```
# 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
```

According to my work, the players that scored on average, more points at home than away are: Jusuf NURkic, RObert Covington, Damian Lillard, and Cody Zeller.

Task 3

Question a

```
library(palmerpenguins)
```

Attaching package: 'palmerpenguins'

The following objects are masked from 'package:datasets':

penguins, penguins_raw

```
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::summarise(n = dplyr::n(), .by = c(species, island)) |>
```

```
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]>

The , <dbl[52]>, and mean to me that an error occurred, as if the data is not formatted as my colleague intended. The might indicated that the each column variables are actually lists of numbers. <dbl[52]> means entry: row Adelie and column Torgersen has 52 double or numbers with possible decimals in it. The indicates that the entry for example entry row Chinstrap and col Torgersen is not a list or is it empty?

###Question b

```
#penguins |>select(species, island, bill_length_mm) |>
  #group_by(species, island) |>
  #summarise(mean (test= bill_length_mm, na.rm = TRUE)) |>
  #pivot_wider(names_from = island,
  #values_from = bill_length_mm)
penguins |>
  select(island , species) |>
  count(species, island) |>
  pivot_wider(names_from = island,
              values_from = n,
              values_fill = 0)
```

```
# 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

Question a

```
penguins |>
  select(species, bill_length_mm, island ) |>
  mutate( bill_length_mm =
    case_when(is.na(bill_length_mm) & species == "Adelie" ~ 26,
    is.na(bill_length_mm) & species == "Gentoo"~ 30,
    .default = bill_length_mm
  )) |>
  arrange(bill_length_mm)
```

```
# A tibble: 344 x 3
  species bill_length_mm island
  <fct>         <dbl> <fct>
1 Adelie          26   Torgersen
2 Gentoo          30   Biscoe
3 Adelie         32.1 Dream
4 Adelie         33.1 Dream
```



```
5 Adelie          33.5 Torgersen
6 Adelie          34   Dream
7 Adelie          34.1 Torgersen
8 Adelie          34.4 Torgersen
9 Adelie          34.5 Biscoe
10 Adelie         34.6 Torgersen
# i 334 more rows
```