HW3_BaseRProgramming

Task 1

Part A: Read in data.txt file

```
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   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
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 beco
```

```
library(palmerpenguins)

data_v1 <- read_csv("/Users/alexiskolecki/Repos/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_v1
```

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

We can not use this specific read_csv() to read this data since the data is seperated by semicolons and not commas or tabs.

```
data <- read_csv2("/Users/alexiskolecki/Repos/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
```

Part 2 : Read in data2.txt

```
data2 <- read_delim("/Users/alexiskolecki/Repos/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

Part 1: Take a glimpse of the trailblazer data set to show that you have read in the data correctly

```
trailblazer <- read_csv("/Users/alexiskolecki/Repos/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.
```

```
trailblazer
```

```
# A tibble: 9 x 11
  Player       Game1_Home Game2_Home Game3_Away Game4_Home Game5_Home Game6_Away
  <chr>             <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
1 Damian Lill~         20         19         12         20         25         14
2 CJ McCollum          24         28         20         25         14         25
3 Norman Powe~         14         16         NA         NA         12         14
4 Robert Covi~          8          6          0          3          9          6
5 Jusuf Nurkic         20          9          4         17         14         13
6 Cody Zeller           5          5          8         10          9          6
7 Anfernee Si~         11         18         12         17          5         19
8 Larry Nance~          2          8          5          8          3          8
9 Nassir Litt~          7         11          5          9          8          8
# i 4 more variables: Game7_Away <dbl>, Game8_Away <dbl>, Game9_Home <dbl>,
#   Game10_Home <dbl>
```

Part 2 : Pivot the data so that you have columns for player, game, location, points. Display the first five rows of your data set.

```
trailblazer_longer <- pivot_longer(trailblazer, cols = 2:11,names_to = "Games", values_to = '
trailblazer_longer <- separate_wider_delim(trailblazer_longer, Games, delim ="_" ,
                                names = c("Games", "Location"))

trailblazer_longer
```

```
# A tibble: 90 x 4
   Player         Games  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
```

Part 3 : Which players scored more on average when playing at home verus away?

```
trailblazer_longer|> pivot_wider(names_from = Location, values_from = Points) |>
  group_by(Player)|>
  summarize(mean_home = mean(Home, na.rm = TRUE),
            mean_away = mean(Away,
           na.rm =TRUE), 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
```

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

Task 3

Part A : Example the Errors in the code that co worker sent you

Whats going on here? Explain the , <dbl [52]> and means.

*There is an error being caused because thre are duplicate rows in the data file.

*The means that that spot in the table is empty and there are no values for this spot in the table.

*The refers to a list object in R.

*The <dbl[52]> means a numeric data type that can be up to 52 values.

Part B : Recreate the Table

```
penguins
```

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

```
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]>
```

Task 4

Replace NA with correct bill values