Into The Tidyverse

Mike Keating

Load Dependencies

```r
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 becom
```

```r
library(palmerpenguins)
```

```

Attaching package: 'palmerpenguins'

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

    penguins, penguins_raw
```

Task 1

The data for this task is called data.txt and data2.txt. Download these and put them in your data folder before answering the questions below.

We can use read_csv functions to read in data. CSV is a comma-separated file i.e. any text file that uses commas as a delimiter to separate the record values for each field. Therefore, to load data from a text file we can use the read_csv() method (or versions of it), even if the file itself does not have a .csv extension.

In the following question, we are going to read in txt data. Part a has us working with the data.txt file. Part b has you working with the data2.txt file.

Part a

We cannot use read_csv() to read the data in data.txt because it uses a comma (',') as the delimiter (the separating character between values). Instead, we must use read_csv2(), which uses a semicolon (';') as its delimiter. This is helpful in reading data from European countries where a comma may be used as a decimal point and not as a field separator.

```r
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.
```

```r
data
```

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

Part b

Read data delimited by "6" and assign factor, double, and character as datatypes for each column.

```r
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

The Portland Trailblazers are a National Basketball Association (NBA) sports team. These data reflect the points scored by 9 Portland Trailblazers players across the first 10 games of the 2021-2022 NBA season. We are going to use these data to show off our data tidying skills. The data we will be using for this task is called trailblazer, and can be found on Moodle.

Part a

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

```r
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.
```

```r
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 b

Pivot the data so that you have columns for Player, Game, Location, Points. Display the first five rows of your data set. Save your new data set as trailblazer_longer. Your data set should contain 90 rows and 4 columns.

Let's get a glimpse at just the original column names:

```
colnames(trailblazer)
```

```
 [1] "Player"       "Game1_Home"  "Game2_Home"  "Game3_Away"  "Game4_Home"
 [6] "Game5_Home"   "Game6_Away"  "Game7_Away"  "Game8_Away"  "Game9_Home"
[11] "Game10_Home"
```

```
trailblazer_longer <- trailblazer |> pivot_longer("Game1_Home":"Game10_Home", names_to = c("
# Show first 5 rows
print(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
```

```
# And checking dimensions
print(dim(trailblazer_longer))
```

```
[1] 90  4
```

Task 3

Task 4