Assignment 7, Part II

Ellen Bledsoe

2024-02-29

Assignment Details

Purpose

The goal of this assignment is to work with dates and times using the `lubridate` package.

Task

Write R code to successfully answer each question below.

Criteria for Success

- Code is within the provided code chunks or new code chunks are created where necessary
- Code chunks run without errors
- Code chunks have brief comments indicating which code is answering which part of the question
- Code will be assessed as follows:
  - Produces the correct answer using the requested approach: 100%
  - Generally uses the right approach, but a minor mistake results in an incorrect answer: 90%
  - Attempts to solve the problem and makes some progress using the core concept, but returns the wrong answer and does not demonstrate comfort with the core concept: 50%
  - Answer demonstrates a lack of understanding of the core concept: 0%
- Any questions requiring written answers are answered with sufficient detail

Due Date

March 11 at midnight MST

Assignment Exercises

The assignment for week 7 is divided into 2 parts:

- Part 1: `lubridate`
- Part 2: `stringr`

This is Part 2, using `stringr`

1. Set-Up (5 pts)

Load in the `tidyverse`.

```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   3.4.4     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 become error
```

2. Vectors (10 pts)

Let's ease into our practice working with strings with some lyrics from the first Black woman to ever hit the top of the Country music charts.

No need to save your outputs.

```
lyrics <- c("This ain't Texas (woo),",
           "ain't no hold 'em (hey)/",
           "So lay your cards down, down, down, down")
```

a. Use `str_length` to determine how many characters are in each string in the vector.

```
str_length(lyrics)
```

```
## [1] 23 24 40
```

b. Use `str_count` to count the number of times the word "down" occurs in each string.

```
str_count(lyrics, "down")
```

```
## [1] 0 0 4
```

c. Use `str_detect` to find the strings which have the word "Texas" in them.

```
str_detect(lyrics, "Texas")
```

```
## [1]  TRUE FALSE FALSE
```

d. Use `str_subset` to select the string that have words in parentheses. Use the following regex as the pattern to match: `\\((.*?)\\)`. Remember that regex patterns need to go inside quotation marks.

```
str_subset(lyrics, "\\((.*?)\\)")
```

```
## [1] "This ain't Texas (woo),"  "ain't no hold 'em (hey)/"
```

e. Use `str_extract` to pull out the parentheticals themselves. Use the same regex as in (d)

```
str_extract(lyrics, "\\((.*?)\\)")
```

```
## [1] "(woo)" "(hey)" NA
```

3. Dugout Data (15 pts)

Dugouts are human-made water reservoirs on the landscape, often used for cattle or other ranching ventures.

Here is another example of data from my postdoc lab that I was asked to clean up. I'm going to go ahead and clean up the column names for the columns we will be using in the following questions.

```
dugout <- read_csv("elevation_2017.csv") %>%
  rename(SoilSalinity = `Soil Salinity`,
         SoilZone = `Soil Zone`,
         MajorSalts = `Major Salts in groundwater`)
```

```
## Rows: 102 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): Site_ID, Date, Soil Salinity, pH, Soil Zone, Location of nearest o...
## dbl  (7): latitude, longitude, Elevation.m, ion Concentration in groundwater...
## time (1): Time
##
## 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.
```

```
dugout
```

```
## # A tibble: 102 x 16
##    Site_ID Date      Time   latitude longitude SoilSalinity pH          SoilZone
##    <chr>   <chr>     <time>    <dbl>     <dbl> <chr>        <chr>       <chr>
##  1 5       24-Aug-17 10:03      51.4     -103. moderate     alkaline    dark gr~
##  2 20      24-Jul-17 11:41      50.1     -102. very slight  unclassifi~ black
##  3 36      10-Aug-17 15:05      52.5     -105. very slight  alkaline    dark gr~
##  4 49      24-Jul-17 13:15      50.0     -102. slight       unclassifi~ black
##  5 51      24-Jul-17 16:19      50.0     -102. slight       unclassifi~ black
##  6 52      25-Jul-17 11:27      49.9     -102. slight       unclassifi~ black
##  7 65      11-Aug-17 11:50      52.6     -110. very slight  slightly a~ dark br~
##  8 68      8-Aug-17  09:30      50.6     -105. very slight  alkaline    brown
##  9 10A     24-Aug-17 12:25      51.8     -103. slight       alkaline    dark gr~
## 10 10B     24-Aug-17 13:14      51.8     -103. slight       alkaline    dark gr~
## # i 92 more rows
## # i 8 more variables: Elevation.m <dbl>,
## #   `Location of nearest observation well` <chr>,
## #   `ion Concentration in groundwater (mg/L)` <dbl>, MajorSalts <chr>,
## #   Anion <chr>, `2017 Well groundwater depth` <dbl>,
## #   `dugout elevation above groundwater` <dbl>, Surface_Sal.ppt <dbl>
```

You do not need to save any of the outputs for this question.

a. Using `filter` and `str_detect`, return rows that have "slight" in the values in the `SoilSalinity` column.

```
dugout %>%
  filter(str_detect(SoilSalinity, "slight"))
```

```
## # A tibble: 85 x 16
##    Site_ID Date      Time   latitude longitude SoilSalinity pH          SoilZone
##    <chr>   <chr>     <time>    <dbl>     <dbl> <chr>        <chr>       <chr>
##  1 20      24-Jul-17 11:41      50.1     -102. very slight  unclassifi~ black
##  2 36      10-Aug-17 15:05      52.5     -105. very slight  alkaline    dark gr~
##  3 49      24-Jul-17 13:15      50.0     -102. slight       unclassifi~ black
##  4 51      24-Jul-17 16:19      50.0     -102. slight       unclassifi~ black
##  5 52      25-Jul-17 11:27      49.9     -102. slight       unclassifi~ black
##  6 65      11-Aug-17 11:50      52.6     -110. very slight  slightly a~ dark br~
##  7 68      8-Aug-17  09:30      50.6     -105. very slight  alkaline    brown
##  8 10A     24-Aug-17 12:25      51.8     -103. slight       alkaline    dark gr~
##  9 10B     24-Aug-17 13:14      51.8     -103. slight       alkaline    dark gr~
## 10 10C     24-Aug-17 10:30      51.8      103. very slight  alkaline    dark gr~
## # i 75 more rows
## # i 8 more variables: Elevation.m <dbl>,
## #   `Location of nearest observation well` <chr>,
## #   `ion Concentration in groundwater (mg/L)` <dbl>, MajorSalts <chr>,
## #   Anion <chr>, `2017 Well groundwater depth` <dbl>,
## #   `dugout elevation above groundwater` <dbl>, Surface_Sal.ppt <dbl>
```

b. Using `filter` and `str_detect`, return rows that have a letter in the values in the `Site_ID` column. The regex pattern to match is `"[A-Z]+"`.

```
dugout %>%
  filter(str_detect(Site_ID,"[A-Z]+"))
```

```
## # A tibble: 94 x 16
##    Site_ID Date      Time   latitude longitude SoilSalinity pH          SoilZone
##    <chr>   <chr>     <time>    <dbl>     <dbl> <chr>        <chr>       <chr>
##  1 10A     24-Aug-17 12:25      51.8     -103. slight       alkaline    dark gr~
##  2 10B     24-Aug-17 13:14      51.8     -103. slight       alkaline    dark gr~
##  3 10C     24-Aug-17 10:30      51.8      103. very slight  alkaline    dark gr~
##  4 10D     24-Aug-17 11:39      51.8     -103. very slight  alkaline    dark gr~
##  5 14A     12-Jul-17 10:15      51.0     -105. very slight  alkaline    brown
##  6 14B     12-Jul-17 12:50      51.0     -105. very slight  alkaline    black
##  7 15A     3-Aug-17  11:41      49.6     -102. slight       neutral to~ dark gr~
##  8 15B     3-Aug-17  14:15      49.5     -102. slight       neutral to~ dark gr~
##  9 22B     8-Aug-17  12:28      51.1      106. very slight  alkaline    brown
## 10 24A     14-Aug-17 14:15      49.9     -110. slight       neutral to~ brown
## # i 84 more rows
## # i 8 more variables: Elevation.m <dbl>,
## #   `Location of nearest observation well` <chr>,
## #   `ion Concentration in groundwater (mg/L)` <dbl>, MajorSalts <chr>,
## #   Anion <chr>, `2017 Well groundwater depth` <dbl>,
## #   `dugout elevation above groundwater` <dbl>, Surface_Sal.ppt <dbl>
```

c. Using `mutate` and `str_replace`, replace the word “acid” with “acidic” in the `pH` column.

```
dugout %>%
  mutate(pH = str_replace(pH, "acid", "acidic"))
```

```
## # A tibble: 102 x 16
##    Site_ID Date      Time   latitude longitude SoilSalinity pH          SoilZone
##    <chr>   <chr>     <time>    <dbl>     <dbl> <chr>        <chr>       <chr>
##  1 5       24-Aug-17 10:03      51.4     -103. moderate     alkaline    dark gr~
##  2 20      24-Jul-17 11:41      50.1     -102. very slight  unclassifi~ black
##  3 36      10-Aug-17 15:05      52.5     -105. very slight  alkaline    dark gr~
##  4 49      24-Jul-17 13:15      50.0     -102. slight       unclassifi~ black
##  5 51      24-Jul-17 16:19      50.0     -102. slight       unclassifi~ black
##  6 52      25-Jul-17 11:27      49.9     -102. slight       unclassifi~ black
##  7 65      11-Aug-17 11:50      52.6     -110. very slight  slightly a~ dark br~
##  8 68      8-Aug-17  09:30      50.6     -105. very slight  alkaline    brown
##  9 10A     24-Aug-17 12:25      51.8     -103. slight       alkaline    dark gr~
## 10 10B     24-Aug-17 13:14      51.8     -103. slight       alkaline    dark gr~
## # i 92 more rows
## # i 8 more variables: Elevation.m <dbl>,
## #   `Location of nearest observation well` <chr>,
## #   `ion Concentration in groundwater (mg/L)` <dbl>, MajorSalts <chr>,
## #   Anion <chr>, `2017 Well groundwater depth` <dbl>,
## #   `dugout elevation above groundwater` <dbl>, Surface_Sal.ppt <dbl>
```

4. Santa Cruz Rodents (20 pts)

Remember the rodent data from the Santa Cruz that we used in our assignment for Week 6? There were quite a few columns that had messy data, and we used a combination of `replace` and `na_if` to address the issues.

Let's use `stringr` functions to complete the same tasks.

First, read in the `capture_data.csv` file.

```r
rodents <- read_csv("capture_data.csv")
```

```
## Rows: 51 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (10): Site, Trap ID, Species, Status (R/N), Sex, Tail length, Hair samp...
## dbl   (4): Total Weight, Bag weight, Animal Weight, Hind foot length
## date  (1): Date
##
## 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.
```

The column names in the dataset have not been cleaned. You can either clean up the column names before working through the questions or you can use the column names in backticks throughout the rest of the question–up to you!

We will be using the `Species`, `Tail length`, `Hair sample (Y/N)`, and `Position (R/L)` columns.

Also, you do not need to save any of the outputs from this question, though you can in b-d, if you would like to.

```r
rodents <- rodents %>%
  rename("TailLength" = `Tail length`,
         "HairSample" = `Hair sample (Y/N)`,
         "Position" = `Position (R/L)`)
```

a. Species codes should be exactly 4 characters long, not more and not less. Filter the dataframe (but do not save it) to show rows that have species codes that do not fit that requirement (hint: use `!=`).

```r
rodents %>%
  filter(str_length(Species) != 4)
```

```
## # A tibble: 2 x 15
##   Date       Site  `Trap ID` Species `Status (R/N)` Sex   `Total Weight`
##   <date>     <chr> <chr>     <chr>   <chr>          <chr>          <dbl>
## 1 2022-11-14 <NA>  4J        SIOC?   N              <NA>              NA
## 2 2022-11-18 <NA>  D6        DIME?   N              F                 44
## # i 8 more variables: `Bag weight` <dbl>, `Animal Weight` <dbl>,
## #   `Hind foot length` <dbl>, TailLength <chr>, HairSample <chr>,
## #   Position <chr>, Handler <chr>, Notes <chr>
```

b. Use `str_remove` to remove the ~ from the `Tail Length` column (it is in the last row).

```r
rodents %>%
  mutate(TailLength = str_remove(TailLength, "~"))
```

```
## # A tibble: 51 x 15
##    Date       Site     `Trap ID` Species `Status (R/N)` Sex   `Total Weight`
##    <date>     <chr>    <chr>     <chr>   <chr>          <chr>          <dbl>
##  1 2022-11-14 Heritage 4C        SIOC    N              F                134
##  2 2022-11-14 <NA>     4D        SIOC    N              M                 136
##  3 2022-11-14 <NA>     4I        SIOC    N              <NA>               90
##  4 2022-11-14 <NA>     2H        REME    N              M                  38
##  5 2022-11-14 <NA>     4J        SIOC?   N              <NA>               NA
##  6 2022-11-14 <NA>     2F        REME    N              F                  22
##  7 2022-11-15 <NA>     4C        SIOC    R              <NA>               NA
##  8 2022-11-15 <NA>     4H        SIOC    N              F                  95
##  9 2022-11-15 <NA>     1H        REME    N              <NA>               26
## 10 2022-11-15 <NA>     1B        REME    N              F                  35
## # i 41 more rows
## # i 8 more variables: `Bag weight` <dbl>, `Animal Weight` <dbl>,
## #   `Hind foot length` <dbl>, TailLength <chr>, HairSample <chr>,
## #   Position <chr>, Handler <chr>, Notes <chr>
```

c. Use `str_remove` to remove the `?` from the `Species` column.

   Because `stringr` by default expects regex in the "pattern" argument and `?` is a special regex character, we need to use the pattern `"\\?"`. The \\ is an "escape," telling regex to treat the ? as a regular ?, not as a regex symbol.

```
rodents %>%
  mutate(Species = str_remove(Species, "\\?"))
```

```
## # A tibble: 51 x 15
##    Date       Site     `Trap ID` Species `Status (R/N)` Sex   `Total Weight`
##    <date>     <chr>    <chr>     <chr>   <chr>          <chr>          <dbl>
##  1 2022-11-14 Heritage 4C        SIOC    N              F                 134
##  2 2022-11-14 <NA>     4D        SIOC    N              M                 136
##  3 2022-11-14 <NA>     4I        SIOC    N              <NA>               90
##  4 2022-11-14 <NA>     2H        REME    N              M                  38
##  5 2022-11-14 <NA>     4J        SIOC    N              <NA>               NA
##  6 2022-11-14 <NA>     2F        REME    N              F                  22
##  7 2022-11-15 <NA>     4C        SIOC    R              <NA>               NA
##  8 2022-11-15 <NA>     4H        SIOC    N              F                  95
##  9 2022-11-15 <NA>     1H        REME    N              <NA>               26
## 10 2022-11-15 <NA>     1B        REME    N              F                  35
## # i 41 more rows
## # i 8 more variables: `Bag weight` <dbl>, `Animal Weight` <dbl>,
## #   `Hind foot length` <dbl>, TailLength <chr>, HairSample <chr>,
## #   Position <chr>, Handler <chr>, Notes <chr>
```

d. Use `str_replace` to replace the `?` in hair sample and position with `NA`. Remember to use `"\\?"`.

   `stringr` expects a character value, and NA is not a character value–it is a NULL value. To get around this, we need to use `NA_character_` in place of `NA`, a special work around.

```
rodents %>%
  mutate(HairSample = str_replace(HairSample, "\\?", NA_character_),
         Position = str_replace(Position, "\\?", NA_character_))
```

```
## # A tibble: 51 x 15
##    Date       Site     `Trap ID` Species `Status (R/N)` Sex   `Total Weight`
##    <date>     <chr>    <chr>     <chr>   <chr>          <chr>          <dbl>
##  1 2022-11-14 Heritage 4C        SIOC    N              F                 134
##  2 2022-11-14 <NA>     4D        SIOC    N              M                 136
##  3 2022-11-14 <NA>     4I        SIOC    N              <NA>               90
##  4 2022-11-14 <NA>     2H        REME    N              M                  38
##  5 2022-11-14 <NA>     4J        SIOC?   N              <NA>              NA
##  6 2022-11-14 <NA>     2F        REME    N              F                 22
##  7 2022-11-15 <NA>     4C        SIOC    R              <NA>              NA
##  8 2022-11-15 <NA>     4H        SIOC    N              F                 95
##  9 2022-11-15 <NA>     1H        REME    N              <NA>              26
## 10 2022-11-15 <NA>     1B        REME    N              F                 35
## # i 41 more rows
## # i 8 more variables: `Bag weight` <dbl>, `Animal Weight` <dbl>,
## #   `Hind foot length` <dbl>, TailLength <chr>, HairSample <chr>,
## #   Position <chr>, Handler <chr>, Notes <chr>
```