Homework 03 CSCI 036 Solutions

Lucas Welch

Due: Friday, 2022-09-23

Instructions

Please box your answers. For numerical answers, this can be done using something like $\boxed{34}$. For text answers, this can be done using something like $\boxed{\text{My answer}}$. The output of a code chunk is automatically boxed, so no need to do more.

This homework uses four files. Each of the files should be placed in a subdirectory called `datasets`. The four files needed are:

- `annual-enterprise-survey-2019-financial-year-provisional-csv.csv`
- `CMC_Sequence_Declarations_2020-09-23.csv`
- `example.xlsx`
- `cancer.dta`

The *newline character* `\n` can be used to tell R that a new line is starting in a string. That allows us to directly create a comma separated file to play with. Consider the following code.

```
df <-
  read_csv(
    "1, Mole, Moly\n
     2, Rat, Ratty\n
     3, Mr. Toad, Toady\n
     4, Mr. Badger, Badger"
  )
```

```
## Rows: 3 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (2): Mole, Moly
## dbl (1): 1
##
## 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.
```

Modify this code by changing parameters to read_csv so that the first row of data does not get turned into header names, and remove the first column of data.

```
df <-
  read_csv(
    "1, Mole, Moly\n
     2, Rat, Ratty\n
     3, Mr. Toad, Toady\n
     4, Mr. Badger, Badger",
    col_names = FALSE
  ) |>
  select(-1)
```

```
## Rows: 4 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (2): X2, X3
## dbl (1): X1
##
## 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.
```

```
df
```

```
## # A tibble: 4 × 2
##   X2         X3
##   <chr>      <chr>
## 1 Mole       Moly
## 2 Rat        Ratty
## 3 Mr. Toad   Toady
## 4 Mr. Badger Badger
```

What command would you give R to read a file called `abc.txt` where values in the data are separated with “|”?

I would give the command delim = "|" to abc.txt

Explain for each of the following strings why it does not give a valid inline CSV file.

```
read_csv("a,b\n1,2,3\n4,5,6")
read_csv("a,b,c\n1,2\n1,2,3,4")
read_csv("a,b\n\"1")
read_csv("a,b\n1,2\na,b")
read_csv("a;b\n1;3")
```

```
#1 read_csv("a,b\n1,2,3\n4,5,6")
#2 read_csv("a,b,c\n1,2\n1,2,3,4")
#3 read_csv("a,b\n\"1")
#4 read_csv("a,b\n1,2\na,b")
#5 read_csv("a;b\n1;3")
```

#1 Revised

```
read_csv("a,b\n1,2\n3,4\n5,6")
```

```
## Rows: 3 Columns: 2
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (2): a, b
##
## 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.
```

```
## # A tibble: 3 × 2
##       a     b
##   <dbl> <dbl>
## 1     1     2
## 2     3     4
## 3     5     6
```

#1 Explanation: The rows in the csv were not shifted to the next row with "\n" command making the numbers combine instead of seperating

#2 Revised

```
read_csv("a,b,c\n1,2,1\n2,3,4")
```

```
## Rows: 2 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (3): a, b, c
##
## 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.
```

```
## # A tibble: 2 × 3
##       a     b     c
##   <dbl> <dbl> <dbl>
## 1     1     2     1
## 2     2     3     4
```

#Explanation: Given that this read_csv file has 3 columns(abc), there were only 2 numbers in in row1 while there are 4 numbers in row2. By shifting t

#3 Revised

```
read_csv("a,b\n1")
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 1 Columns: 2
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (1): a
## lgl (1): b
##
## 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.
```

```
## # A tibble: 1 × 2
##       a b
##   <dbl> <lgl>
## 1     1 NA
```

#Explanation: There was an uneven amount of quotations(3) and an extra "\" making the csv file unreadable. If just the quotations were taken out, the

#4 Revised

```
read_csv("a,b\n1,2")
```

```
## Rows: 1 Columns: 2
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (2): a, b
##
## 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.
```

```
## # A tibble: 1 × 2
##       a     b
##   <dbl> <dbl>
## 1     1     2
```

#Explanation: There was another row added to the csv file after \n1,2 stating \na,b which is the same as the column name. This is not a value, its a col

#5 Revised

```
read_csv("a,b\n1,3")
```

```
## Rows: 1 Columns: 2
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (2): a, b
##
## 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.
```

```
## # A tibble: 1 × 2
##       a     b
##   <dbl> <dbl>
## 1     1     3
```

#Explanation: This csv file seperated its numbers with semicolons instead of commas. This made col_a, row_1 "1:3", which is incorrect because there

Consider the following vector of numbers.

```
x <- c(15, 20, 14, 8, 7)
```

a. Write an R command that returns a vector of boolean values that are TRUE if and only if the component of the vector equals 8 exactly.

b. Write an R command that returns a vector of boolean values that are TRUE if and only if the component of the vector is greater than 9 and less than 16.

c. Write an R command that returns a vector of boolean values that are TRUE if and only if the component of the vector is at most 7 or at least 14.

A.

```
c(8) == x
```

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

B.

```
c(9) < x & c(16) > x
```

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

C.

```
c(7) > x | c(14) < x
```

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

Consider the CSV file `datasets/CMC_Sequence_Declarations_2020-09-23.csv`.

a. Load this into a data set `sequences`, dealing with any comments in the file and labeling the first variable `sequence` and the second `number_of_records`.

b. Filter out (remove) those sequences with fewer than 10 records.

A.

```
sequences <- read_csv("datasets/CMC_Sequence_Declarations_2020-09-23.csv", col_names = c("sequence", "number_of_r
ecords"), skip = 2)
```

```
## Rows: 11 Columns: 2
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): sequence
## dbl (1): number_of_records
##
## 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.
```

```
sequences
```

```
## # A tibble: 11 × 2
##    sequence                 number_of_records
##    <chr>                                <dbl>
##  1 Asian Amer. Studies                      2
##  2 Computer Science                        45
##  3 Data Science                            66
##  4 Ethics                                   5
##  5 Financial Economics                     70
##  6 Gender/Sexuality Studies                 7
##  7 Holocaust & Human Rights                10
##  8 Leadership                              66
##  9 Legal Studies                          142
## 10 Public Policy                           15
## 11 Scientific Modeling                      2
```

B.

```
sequences <- read_csv("datasets/CMC_Sequence_Declarations_2020-09-23.csv", col_names = c("sequence", "number_of_r
ecords"), skip = 2) |>
  filter(number_of_records < 10)
```

```
## Rows: 11 Columns: 2
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): sequence
## dbl (1): number_of_records
##
## 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.
```

```
sequences
```

```
## # A tibble: 4 × 2
##   sequence                 number_of_records
##   <chr>                                <dbl>
## 1 Asian Amer. Studies                      2
## 2 Ethics                                   5
## 3 Gender/Sexuality Studies                 7
## 4 Scientific Modeling                      2
```

Consider the following command:

```
aes <- read_csv("datasets/annual-enterprise-survey-2019-financial-year-provisional-csv.csv")
```

```
## Rows: 32445 Columns: 10
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): Industry_aggregation_NZSIOC, Industry_code_NZSIOC, Industry_name_NZ...
## dbl (1): Year
##
## 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.
```

a. What type of variable was `Year` parsed as?

b. Modify the above command to read in the `Year` variable as an integer.

A.

```
aes <- read_csv("datasets/annual-enterprise-survey-2019-financial-year-provisional-csv.csv")
```

```
## Rows: 32445 Columns: 10
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): Industry_aggregation_NZSIOC, Industry_code_NZSIOC, Industry_name_NZ...
## dbl (1): Year
##
## 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.
```

```
aes
```

```
## # A tibble: 32,445 × 10
##     Year Industry_…¹ Indus…² Indus…³ Units Varia…⁴ Varia…⁵ Varia…⁶ Value Indus…⁷
##    <dbl> <chr>       <chr>   <chr>   <chr> <chr>   <chr>   <chr>   <chr> <chr>
##  1  2019 Level 1     99999   All in… Doll… H01     Total … Financ… 728,… ANZSIC…
##  2  2019 Level 1     99999   All in… Doll… H04     Sales,… Financ… 643,… ANZSIC…
##  3  2019 Level 1     99999   All in… Doll… H05     Intere… Financ… 62,9… ANZSIC…
##  4  2019 Level 1     99999   All in… Doll… H07     Non-op… Financ… 21,5… ANZSIC…
##  5  2019 Level 1     99999   All in… Doll… H08     Total … Financ… 634,… ANZSIC…
##  6  2019 Level 1     99999   All in… Doll… H09     Intere… Financ… 35,2… ANZSIC…
##  7  2019 Level 1     99999   All in… Doll… H10     Indire… Financ… 7,458 ANZSIC…
##  8  2019 Level 1     99999   All in… Doll… H11     Deprec… Financ… 20,9… ANZSIC…
##  9  2019 Level 1     99999   All in… Doll… H12     Salari… Financ… 112,… ANZSIC…
## 10  2019 Level 1     99999   All in… Doll… H13     Redund… Financ… 206   ANZSIC…
## # … with 32,435 more rows, and abbreviated variable names
## #   ¹Industry_aggregation_NZSIOC, ²Industry_code_NZSIOC, ³Industry_name_NZSIOC,
## #   ⁴Variable_code, ⁵Variable_name, ⁶Variable_category, ⁷Industry_code_ANZSIC06
```

Year is being parsed as a "dbl"

B.

```
aes <- read_csv("datasets/annual-enterprise-survey-2019-financial-year-provisional-csv.csv", col_types = cols("Ye
ar" = col_integer()))
```

Consider the file `datasets/example.xlsx` downloadable from the course website.

a. Give a command to read this into the tibble `data` using `read_excel` from the `readxl` library.

b. How many data points are there in the resulting dataset?

A.

```
library(readxl)
data <- read_excel("datasets/example.xlsx")
data
```

```
## # A tibble: 500 × 9
##    `First Name` `Last Name` Company Na…¹ Address City  State Phone…² Email Web
##    <chr>        <chr>       <chr>        <chr>   <chr> <chr> <chr>   <chr> <chr>
##  1 Rebbecca     Didio       Brandt, Jon… 171 E … Leith TAS   03-817… rebb… http…
##  2 Stevie       Hallo       Landrum Tem… 22222 … Pros… QLD   07-999… stev… http…
##  3 Mariko       Stayer      Inabinet, M… 534 Sc… Hamel WA    08-555… mari… http…
##  4 Gerardo      Woodka      Morris Down… 69206 … Talm… NSW   02-604… gera… http…
##  5 Mayra        Bena        Buelt, Davi… 808 Gl… Lane… NSW   02-145… mayr… http…
##  6 Idella       Scotland    Artesian Ic… 373 La… Cart… WA    08-786… idel… http…
##  7 Sherill      Klar        Midway Hotel 87 Syl… Nyam… WA    08-652… skla… http…
##  8 Ena          Desjardiws  Selsor, Rob… 60562 … Bend… NSW   02-522… ena_… http…
##  9 Vince        Siena       Vincent J P… 70 S 1… Purr… QLD   07-318… vinc… http…
## 10 Theron       Jarding     Prentiss, P… 8839 V… Blan… SA    08-689… tjar… http…
## # … with 490 more rows, and abbreviated variable names ¹`Company Name`,
## #   ²`Phone No`
```

B. 500

Continue with the dataset from the last problem. This data comes from Australia. Filter to only keep the clients from New South Wales, abbreviated NSW.

```
data <- read_excel("datasets/example.xlsx") |>
  filter(State == "NSW")
data
```

```
## # A tibble: 125 × 9
##    `First Name` `Last Name` Company Na…¹ Address City  State Phone…² Email Web
##    <chr>        <chr>       <chr>        <chr>   <chr> <chr> <chr>   <chr> <chr>
##  1 Gerardo      Woodka      Morris Down… 69206 … Talm… NSW   02-604… gera… http…
##  2 Mayra        Bena        Buelt, Davi… 808 Gl… Lane… NSW   02-145… mayr… http…
##  3 Ena          Desjardiws  Selsor, Rob… 60562 … Bend… NSW   02-522… ena_… http…
##  4 Reita        Tabar       Cooper Myer… 79620 … Arth… NSW   02-351… rtab… http…
##  5 Camellia     Pylant      Blackley, W… 570 W … Tugg… NSW   02-517… came… http…
##  6 Hayley       Taghon      Biltmore Te… 72 Wyo… Eugo… NSW   02-163… htag… http…
##  7 Norah        Daleo       Gateway Ref… 754 Sa… Kota… NSW   02-532… ndal… http…
##  8 Ben          Majorga     Voyager Tra… 13904 … Wher… NSW   02-817… ben.… http…
##  9 Oren         Lobosco     Vei Inc      1585 S… Dang… NSW   02-504… olob… http…
## 10 Keena        Rebich      Affilated C… 3713 P… Sawt… NSW   02-497… kreb… http…
## # … with 115 more rows, and abbreviated variable names ¹`Company Name`,
## #   ²`Phone No`
```

Continue with the dataset from the last problem. Arrange the data so that each observation is in alphabetical order first by State, and then by City within that state.

```
data <- read_excel("datasets/example.xlsx") |>
  arrange(State, City)
data
```

```
## # A tibble: 500 × 9
##    `First Name` `Last Name` Company Na…¹ Address City  State Phone…² Email Web
##    <chr>        <chr>       <chr>        <chr>   <chr> <chr> <chr>   <chr> <chr>
##  1 Soledad      Mockus      Sinclair Ma… 75 Elm… Bart… ACT   02-129… sole… http…
##  2 Annamae      Lothridge   Highland Me… 584 Me… Civi… ACT   02-191… alot… http…
##  3 Katheryn     Lamers      Sonoco Prod… 62171 … Fysh… ACT   02-488… kath… http…
##  4 Roy          Nybo        Phoenix Pho… 823 Fi… Red … ACT   02-531… rnyb… http…
##  5 Jamie        Kushnir     Bell Electr… 3216 W… Tugg… ACT   02-462… jami… http…
##  6 Dana         Vock        Fried, Mont… 49 Wal… Yarr… ACT   02-668… dana… http…
##  7 Santos       Wisenbaker  Brattleboro… 67729 … Allw… NSW   02-295… swis… http…
##  8 Emmanuel     Avera       Bank Of New… 3883 N… Appin NSW   02-198… emma… http…
##  9 Reita        Tabar       Cooper Myer… 79620 … Arth… NSW   02-351… rtab… http…
## 10 Princess     Saffo       Asian Jewel… 12398 … Aubu… NSW   02-265… prin… http…
## # … with 490 more rows, and abbreviated variable names ¹`Company Name`,
## #   ²`Phone No`
```

The haven package contains functions for loading in datasets from commercial packages such as STATA.

```
library(haven)
```

a. Use the read_dta function to read in the file `cancer.dta` from the `datasets` subdirectory.

b. How many participants are there of age 60 or greater?

A.

```
read_dta("datasets/cancer.dta")
```

```
## # A tibble: 48 × 4
##    studytime  died  drug   age
##        <dbl> <dbl> <dbl> <dbl>
##  1         1     1     1    61
##  2         1     1     1    65
##  3         2     1     1    59
##  4         3     1     1    52
##  5         4     1     1    56
##  6         4     1     1    67
##  7         5     1     1    63
##  8         5     1     1    58
##  9         8     1     1    56
## 10         8     0     1    58
## # … with 38 more rows
```

B.

```
read_dta("datasets/cancer.dta") |>
  filter(age >= 60)
```

```
## # A tibble: 13 × 4
##    studytime  died  drug   age
##        <dbl> <dbl> <dbl> <dbl>
##  1         1     1     1    61
##  2         1     1     1    65
##  3         4     1     1    67
##  4         5     1     1    63
##  5        12     1     1    62
##  6         6     1     2    67
##  7         6     0     2    65
##  8        11     0     2    61
##  9        13     1     2    62
## 10        16     1     2    67
## 11        17     0     3    60
## 12        33     1     3    60
## 13        34     0     3    62
```

13