Data Classes - Exercises

Gbemisola Talabi

21 September 2022

Part 1

Load all the libraries we will use in this lab.

```
library(readr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v dplyr   1.0.9
## v tibble  3.1.8     v stringr 1.4.1
## v tidyr   1.2.0     v forcats 0.5.2
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

0. Write you name in the YAML header above
1. Create some data to work with.

First, create a vector that has class integer called `int_vect` that starts at 1 and goes up to 10 and repeats this sequence 3 times using `rep` (hint use `seq()`).

```
int_vect <- as.integer(rep(seq(from =1, to = 10, by = 1), times =3))
is.integer(int_vect)
```

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

```
class(int_vect)
```

```
## [1] "integer"
```

2. Next, create a random vector of 30 values from a set of integers from 1 to 30 called `rand_vect` (hint use `sample()` and set the replace argument to `TRUE`).

Because we are using a random vector let's use the `set.seed()` function to make sure we all have the same result - this code is already in the code chunk for you. Simply create the vector below the `set.seed` line.

```
set.seed(1234)
rand_vect <- sample(
  x = seq(from = 1, to = 30, by = 1),
  size = 30, replace = TRUE
)

# this also works... need to set the seed right before
set.seed(1234)
rand_vect <- sample(x = 1:30, size = 30, replace = TRUE)
```

3a. Create a vector that repeats `c(TRUE, TRUE, FALSE)` 10 times called `TF_vect`. Also create a vector that repeats `c("TRUE", "TRUE", "FALSE")` 10 times called `TF_vect2`.

```
TF_vect <- rep(c(TRUE, TRUE, FALSE), times =10)
TF_vect2 <- rep(c("TRUE", "TRUE", "FALSE"), times = 10)
```

3. Create a tibble combining these vectors together called `vect_data` using the following code.

```
vect_data <- tibble(int_vect, rand_vect, TF_vect, TF_vect2)
```

4. Take a look at 5 random rows using the `slice_sample()` function. Try this a few times to see how the results change.

```
slice_sample(vect_data, n = 5)
```

```
## # A tibble: 5 x 4
##   int_vect rand_vect TF_vect TF_vect2
##      <int>     <int> <lgl>   <chr>
## 1        3        26 FALSE   FALSE
## 2        4        22 TRUE    TRUE
## 3        6         4 TRUE    TRUE
## 4        5         5 TRUE    TRUE
## 5        2        16 TRUE    TRUE
```

5. Take a look at 5 random rows using the `slice_sample()` function again but this time with `set.seed(1234)` as the first line of the chunk. Try this a few times to see the results. (Don't forget to not copy the backticks.)

```
set.seed(1234)
slice_sample(vect_data, n = 5)
```

```
## # A tibble: 5 x 4
##   int_vect rand_vect TF_vect TF_vect2
##      <int>     <int> <lgl>   <chr>
## 1        8         8 TRUE    TRUE
## 2        6        26 TRUE    TRUE
## 3        6         4 TRUE    TRUE
## 4        2        30 TRUE    TRUE
## 5        5         5 TRUE    TRUE
```

6. Check to see if the `TF_vect` is logical. Check to see if `TF_vect2` is logical. Why are the results what they are?

```
class(TF_vect)
```

```
## [1] "logical"
```

```
class(TF_vect2)
```

```
## [1] "character"
```

```
# because TF_vect is not in quotes, it is a logical element.
#because TF_vect2 is in quotes, it is a character.
```

7. Use `mutate()` function to create a new variable in `vect_data` named `type_fact` that is of class `factor` made from the `int_vect` variable. Take a look at the data and observe how the class is different for the new variable compared to `int_vect`.

Part 2

1. Read in the Charm City Circulator data file "circulator_ridership.csv". We have used this data before and should be in your data folder, but you can also download in again from Day1. Remember that a .RMD file automatically makes its working directory where the .RMD file is located. So you will likely need to use `../data/circulator_ridership.csv` as the relative path to move out the subdirector with `../` and then back into the data folder with `data/`

```
charmcity<- read_csv("../data/circulator_ridership.csv")
```

```
## Rows: 13752 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): day, line, type
## dbl  (2): daily, number
## 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.
```

Use the `str()` function to take a look at the data and learn about the column types.

```
str(charmcity)
```

```
## spec_tbl_df [13,752 x 6] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ day   : chr [1:13752] "Monday" "Tuesday" "Wednesday" "Thursday" ...
##  $ date  : Date[1:13752], format: "2010-01-11" "2010-01-12" ...
##  $ daily : num [1:13752] 952 796 1212 1214 1644 ...
##  $ line  : chr [1:13752] "orange" "orange" "orange" "orange" ...
##  $ type  : chr [1:13752] "Boardings" "Boardings" "Boardings" "Boardings" ...
##  $ number: num [1:13752] 877 777 1203 1194 1645 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   day = col_character(),
##   ..   date = col_date(format = ""),
##   ..   daily = col_double(),
##   ..   line = col_character(),
##   ..   type = col_character(),
##   ..   number = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

2. Use the `mutate()` function to create a new column named `date_formatted` that is of `Date` class. The new variable is created from `date` column. Hint: use `mdy()` function.

```
charmcity <- charmcity %>% mutate(
              date_formatted = ymd(date),
              date = as.numeric(date))
glimpse(charmcity)
```

```
## Rows: 13,752
## Columns: 7
## $ day            <chr> "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",~
## $ date           <dbl> 14620, 14621, 14622, 14623, 14624, 14625, 14626, 14627,~
## $ daily          <dbl> 952.0, 796.0, 1211.5, 1213.5, 1644.0, 1490.5, 888.5, 99~
## $ line           <chr> "orange", "orange", "orange", "orange", "orange", "oran~
## $ type           <chr> "Boardings", "Boardings", "Boardings", "Boardings", "Bo~
## $ number         <dbl> 877, 777, 1203, 1194, 1645, 1457, 839, 999, 1023, 1375,~
## $ date_formatted <date> 2010-01-11, 2010-01-12, 2010-01-13, 2010-01-14, 2010-0~
```

3. Move the `date_formatted` variable to be before `date` using the `relocate` function. Take a look at the data using `glimpse()`. Note the difference between `date` and `date_formatted` columns.

```
charmcity <- charmcity %>% relocate(date_formatted, .before = date)
glimpse(charmcity)
```

```
## Rows: 13,752
## Columns: 7
## $ day            <chr> "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",~
## $ date_formatted <date> 2010-01-11, 2010-01-12, 2010-01-13, 2010-01-14, 2010-0~
## $ date           <dbl> 14620, 14621, 14622, 14623, 14624, 14625, 14626, 14627,~
```

```
## $ daily          <dbl> 952.0, 796.0, 1211.5, 1213.5, 1644.0, 1490.5, 888.5, 99~
## $ line           <chr> "orange", "orange", "orange", "orange", "orange", "oran~
## $ type           <chr> "Boardings", "Boardings", "Boardings", "Boardings", "Bo~
## $ number         <dbl> 877, 777, 1203, 1194, 1645, 1457, 839, 999, 1023, 1375,~
```

4. Create a list data object called `classes_data` that combines the `vect_data` from the first part of the lab and `circ`. Use `glimpse()` to look at the data.

```
classes_data <- c(vect_data, charmcity)
glimpse(classes_data)
```

```
## List of 11
##  $ int_vect      : int [1:30] 1 2 3 4 5 6 7 8 9 10 ...
##  $ rand_vect     : int [1:30] 28 16 26 22 5 12 15 9 5 6 ...
##  $ TF_vect       : logi [1:30] TRUE TRUE FALSE TRUE TRUE FALSE ...
##  $ TF_vect2      : chr [1:30] "TRUE" "TRUE" "FALSE" "TRUE" ...
##  $ day           : chr [1:13752] "Monday" "Tuesday" "Wednesday" "Thursday" ...
##  $ date_formatted: Date[1:13752], format: "2010-01-11" "2010-01-12" ...
##  $ date          : num [1:13752] 14620 14621 14622 14623 14624 ...
##  $ daily         : num [1:13752] 952 796 1212 1214 1644 ...
##  $ line          : chr [1:13752] "orange" "orange" "orange" "orange" ...
##  $ type          : chr [1:13752] "Boardings" "Boardings" "Boardings" "Boardings" ...
##  $ number        : num [1:13752] 877 777 1203 1194 1645 ...
```

5. Use `range()` function on `date_formatted` variable to display the range of dates in the data set. How does this compare to that of `date`? Why? (Hint: use the pull function first to pull the values.)

```
range(charmcity$date_formatted)
```

```
## [1] "2010-01-11" "2013-03-01"
```

```
range(charmcity$date)
```

```
## [1] 14620 15765
```

```
#the date_formatted is in date class(year/month/day) and the date is in numerical class.
#because we changed the date_formatted variable to Date class
```

6. Use the `group_by` function on `day` and `line` variables with the `summarize()` function, to display the `number` of orange boardings observations with each day (hint: `use sum()`). Which day had the most boardings? Is this true for the other routes (purple boardings, green boardings, banner boardings)?

```
charmcity %>%
  group_by(day, line) %>%
  summarize(type)
```

```
## `summarise()` has grouped output by 'day', 'line'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 13,752 x 3
## # Groups:   day, line [28]
##    day    line   type
##    <chr>  <chr>  <chr>
##  1 Friday banner Boardings
##  2 Friday banner Boardings
##  3 Friday banner Boardings
##  4 Friday banner Boardings
##  5 Friday banner Boardings
##  6 Friday banner Boardings
##  7 Friday banner Boardings
##  8 Friday banner Boardings
##  9 Friday banner Boardings
## 10 Friday banner Boardings
## # ... with 13,742 more rows
## # i Use `print(n = ...)` to see more rows
```

```
sum((charmcity$line == "orange") & (charmcity$type == "Boardings"))
```

```
## [1] 1146
```

```
sum((charmcity$line == "purple") & (charmcity$type == "Boardings"))
```

```
## [1] 1146
```

```
sum((charmcity$line == "green") & (charmcity$type == "Boardings"))
```

```
## [1] 1146
```

```
sum((charmcity$line == "banner") & (charmcity$type == "Boardings"))
```

```
## [1] 1146
```

```
# yes, this is true for every other route
```