Class Activity 10

Your name here

January 21 2024

Problem 1

a) Use `read_csv()` to import the `desserts` data set from https://raw.githubusercontent.com/deepbas/statdatasets/main/desserts.csv. Use `glimpse` to see if the data import is alright.

```
url <- "https://raw.githubusercontent.com/deepbas/statdatasets/main/desserts.csv"
desserts <- read_csv(url)
glimpse(desserts)
Rows: 549
Columns: 16
$ series                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ episode               <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, ~
$ baker                 <chr> "Annetha", "David", "Edd", "Jasminder", "Jonatha~
$ technical             <chr> "2nd", "3rd", "1st", "N/A", "9th", "N/A", "8th",~
$ result                <chr> "IN", "IN", "IN", "IN", "IN", "IN", "IN", "IN", ~
$ uk_airdate            <chr> "17 August 2010", "17 August 2010", "17 August 2~
$ us_season             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ us_airdate            <date> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
$ showstopper_chocolate <chr> "chocolate", "chocolate", "no chocolate", "no ch~
$ showstopper_dessert   <chr> "other", "other", "other", "other", "other", "ca~
$ showstopper_fruit     <chr> "no fruit", "no fruit", "no fruit", "no fruit", ~
$ showstopper_nut       <chr> "no nut", "no nut", "no nut", "no nut", "almond"~
$ signature_chocolate   <chr> "no chocolate", "chocolate", "no chocolate", "no~
$ signature_dessert     <chr> "cake", "cake", "cake", "cake", "cake", "cake", ~
$ signature_fruit       <chr> "no fruit", "fruit", "fruit", "fruit", "fruit", ~
$ signature_nut         <chr> "no nut", "no nut", "no nut", "no nut", "no nut"~
```

Does everything look good? Import the dataset with correct data types, if needed. Fix the problems, if any.

```
# your r-code

desserts <- read_csv(url,
                     col_types = list(
                       technical = col_number(),
                       uk_airdate = col_date()
                     ))
```

```
problems(desserts)
# A tibble: 556 x 5
     row   col expected        actual         file
   <int> <int> <chr>           <chr>          <chr>
 1     2     6 date in ISO8601 17 August 2010 ""
 2     3     6 date in ISO8601 17 August 2010 ""
 3     4     6 date in ISO8601 17 August 2010 ""
 4     5     4 a number        N/A            ""
 5     5     6 date in ISO8601 17 August 2010 ""
 6     6     6 date in ISO8601 17 August 2010 ""
 7     7     4 a number        N/A            ""
 8     7     6 date in ISO8601 17 August 2010 ""
 9     8     6 date in ISO8601 17 August 2010 ""
10     9     4 a number        N/A            ""
# i 546 more rows

desserts <- read_csv(
  "https://raw.githubusercontent.com/deepbas/statdatasets/main/desserts.csv",
    col_types = list(
    technical = col_number(),
    uk_airdate = col_date(format = "%d %B %Y")
  )
)

problems(desserts)
# A tibble: 7 x 5
    row   col expected actual file
  <int> <int> <chr>    <chr>  <chr>
1     5     4 a number N/A    ""
2     7     4 a number N/A    ""
3     9     4 a number N/A    ""
4    11     4 a number N/A    ""
5    35     4 a number N/A    ""
6    36     4 a number N/A    ""
7    37     4 a number N/A    ""

desserts <- read_csv(
  "https://raw.githubusercontent.com/deepbas/statdatasets/main/desserts.csv",
  col_types = list(
    technical = col_number(),
    uk_airdate = col_date(format = "%d %B %Y")
  ),
  na = c("", "NA", "N/A")
)

problems(desserts)
# A tibble: 0 x 5
# i 5 variables: row <int>, col <int>, expected <chr>, actual <chr>, file <chr>
glimpse(desserts)
Rows: 549
Columns: 16
$ series                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ episode               <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, ~
```

```
$ baker                 <chr> "Annetha", "David", "Edd", "Jasminder", "Jonatha~
$ technical             <dbl> 2, 3, 1, NA, 9, NA, 8, NA, 10, NA, 8, 6, 2, 1, 3~
$ result                <chr> "IN", "IN", "IN", "IN", "IN", "IN", "IN", "IN", ~
$ uk_airdate            <date> 2010-08-17, 2010-08-17, 2010-08-17, 2010-08-17,~
$ us_season             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ us_airdate            <date> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
$ showstopper_chocolate <chr> "chocolate", "chocolate", "no chocolate", "no ch~
$ showstopper_dessert   <chr> "other", "other", "other", "other", "other", "ca~
$ showstopper_fruit     <chr> "no fruit", "no fruit", "no fruit", "no fruit", ~
$ showstopper_nut       <chr> "no nut", "no nut", "no nut", "no nut", "almond"~
$ signature_chocolate   <chr> "no chocolate", "chocolate", "no chocolate", "no~
$ signature_dessert     <chr> "cake", "cake", "cake", "cake", "cake", "cake", ~
$ signature_fruit       <chr> "no fruit", "fruit", "fruit", "fruit", "fruit", ~
$ signature_nut         <chr> "no nut", "no nut", "no nut", "no nut", "no nut"~
```

Problem 2

Use the appropriate `read_<type>()` function to import the following data sets:

- `https://deepbas.io/data/simple-1.dat`
- `https://deepbas.io/data/mild-1.csv`
- `https://deepbas.io/data/tricky-1.csv`
- `https://deepbas.io/data/tricky-2.csv`

If you hit any errors/problems, be sure to explore them and identify the issue, even if you can't "fix" it.

a)

```
simple1 <- readr::read_csv("https://deepbas.io/data/simple-1.dat")
problems(simple1)
# A tibble: 0 x 5
# i 5 variables: row <int>, col <int>, expected <chr>, actual <chr>, file <chr>
```

b)

```
mild1 <- readr::read_delim("https://deepbas.io/data/mild-1.csv", delim = "|")
problems(mild1)
# A tibble: 0 x 5
# i 5 variables: row <int>, col <int>, expected <chr>, actual <chr>, file <chr>
```

c)

```
tricky1 <-  read_csv("https://deepbas.io/data/tricky-1.csv")
problems(tricky1)
# A tibble: 2 x 5
    row   col expected  actual    file
  <int> <int> <chr>     <chr>     <chr>
1     4     4 5 columns 4 columns ""
2     7     4 5 columns 4 columns ""
```

The issue is that we have missing values that aren't specifically included in the rows 4 and 7 of the **original** file (so rows 3 and 6 once we load the data). We can fix this with post processing.

```
tricky1[3,] <- c(tricky1[3,1:2], NA, tricky1[3,3:4])
tricky1[6,] <- c(tricky1[6,1:2], NA, tricky1[6,3:4])
tricky1
```

```
# A tibble: 10 x 5
   first     last             address                 city         postcode
   <chr>     <chr>            <chr>                   <chr>        <chr>
 1 Leah      Downs            688-5741 Ut St.         Owensboro    V9Z 9K2
 2 Boris     Kirby            257-5422 Vel Avenue     Rialto       C6I 9S0
 3 Naida     Franco           <NA>                    Atwater      T8K 7U8
 4 Xena      Tucker           7218 A St.              Grand Forks  M6O 1X4
 5 Rylee     Wise             155-6070 Purus. St.     Bradford     65359
 6 Gallagher 2415 Ligula. St. <NA>                    Carbondale   55211
 7 Griffin   Benjamin         3261 Ac St.             Guayama      94450
 8 Rinah     Bradley          787-9626 Eget Avenue    Norton       17673
 9 Tobias    Walter           4717 Mauris. Street     Attleboro    73678
10 Boris     Farley           893-8193 Quisque Avenue San Clemente 74492
```

d)

```
tricky2 <- read_csv("https://deepbas.io/data/tricky-2.csv")
problems(tricky2)
# A tibble: 0 x 5
# i 5 variables: row <int>, col <int>, expected <chr>, actual <chr>, file <chr>
```

This looks like a missing value problem again! Let's look at the rows with missing values:

```
# parse in sections

tricky2_part1 <- read_csv("https://deepbas.io/data/tricky-2.csv",
                          n_max = 7)

# fix the city column

tricky2_part1 <-  tricky2_part1 %>% separate(city, c("city", "state"), sep = ",")


# remove the last row

tricky2_part1 <-  tricky2_part1 %>% select(-c(7))
cnames <- colnames(tricky2_part1)
```

```
tricky2_part2 <- read_csv("https://deepbas.io/data/tricky-2.csv",
                          skip = 8,
                          col_names = cnames)
```

```
tricky2_part2 <- read_csv(
  "https://deepbas.io/data/tricky-2.csv",
  skip = 8,
  col_names = c("iata", "airport", "city", "state",  "latitude", "longitude")
)
```

```
# join the two parts
data_combined <- full_join(tricky2_part1, tricky2_part2)
```

Acknowledgemant

Parts of the activities are adapted from similar activity written by Adam Loy.