Problem_Set_Week_2

Helge_Moes

22-01-2023

Table of Contents

Directions for the student 1
Questions 2
Answer: 6
6. For the selected table (tibble) in your *Data Project* (see below), spot one violation of the four rules for tidy data (see tutorial materials) and fix it with R code. 8
Answer: 9

Directions for the student

- Put all R code in code chunks and verbal answers outside code chunks.
- If you cannot make a piece of R code to work, set the code chunk option eval=FALSE.
- Ensure that the R Markdown document knits without problems into a PDF or Word document.
- Submit the R Markdown document on Canvas (under Assignments) before the deadline.

```
#Load all libraries in this code chunk.
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.2 ──
## ✔ ggplot2 3.4.0      ✔ purrr   1.0.1
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.5.0
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
```

```
##
##     date, intersect, setdiff, union
```

Questions

The work space **jazz_concerts.RData** contains information on all professional jazz concerts in The Netherlands in 1991. Load this work space.

Use the `tidyverse::` package (including `forcats::` and `lubridate::`) to answer the following questions.

1. Distinguish (at least) three types of cases (units) in the data. Explain why it is not optimal to have these types of cases (units) in one table (tibble). Create a separate table for each type of case (without duplicate rows) and ensure that the tables can be linked via a fourth table.

```
#Add your R code for answering this question here.

#load file and present initial table (linked table)
load("Jazz_concerts.RData")
concerts

## # A tibble: 4,540 × 18
##    musicnr lastname firstname national resid…¹ country concnr podium city
date
##      <int> <chr>    <chr>     <chr>    <chr>   <chr>    <int> <chr>  <chr>
<chr>
##  1       2 Baars    Ab        Ned      Amster… Ned         14 Parad… Tilb…
1/3/…
##  2       2 Baars    Ab        Ned      Amster… Ned         24 Bimhu… Amst…
1/5/…
##  3       2 Baars    Ab        Ned      Amster… Ned        462 Brouw… Leeu…
3/3/…
##  4       2 Baars    Ab        Ned      Amster… Ned        799 Bimhu… Amst…
4/13…
##  5       2 Baars    Ab        Ned      Amster… Ned        910 Bimhu… Amst…
4/27…
##  6       2 Baars    Ab        Ned      Amster… Ned       1298 Bimhu… Amst…
6/29…
##  7       2 Baars    Ab        Ned      Amster… Ned       1410 Parad… Amst…
8/17…
##  8       2 Baars    Ab        Ned      Amster… Ned       1443 Dordt… Dord…
8/31…
##  9       2 Baars    Ab        Ned      Amster… Ned       2776 <NA>   Zwol…
9/7/…
## 10       2 Baars    Ab        Ned      Amster… Ned       1663 Brouw… Leeu…
9/29…
## # … with 4,530 more rows, 8 more variables: time <chr>, instrument1 <chr>,
## #   instrument2 <chr>, instrument3 <chr>, instrument4 <chr>, bandnr <int>,
## #   bandname <chr>, band_info <chr>, and abbreviated variable name ¹
residence
```

```
#tibble type 1, containing the number of the musicians and their personal
information
tibble_1_musician <- concerts %>%
  select(musicnr, lastname, firstname, national, residence, country) %>%
  distinct()
tibble_1_musician
```

```
## # A tibble: 2,005 × 6
##    musicnr lastname   firstname    national residence country
##      <int> <chr>      <chr>        <chr>    <chr>     <chr>
##  1       2 Baars      Ab           Ned      Amsterdam Ned
##  2       3 Bakema     Abel         Ned      <NA>      <NA>
##  3       4 Jong       Ad de        Ned      <NA>      <NA>
##  4       5 Olivier    Adam         <NA>     <NA>      <NA>
##  5       6 Boon       Adri         Ned      <NA>      <NA>
##  6       7 Braat      Adrie        Ned      <NA>      <NA>
##  7       8 Takase     Aki          Jap      <NA>      <NA>
##  8       9 Purves     Alan 'Gunga' GB       <NA>      <NA>
##  9      10 Laurillard Alan         <NA>     <NA>      <NA>
## 10      11 Macon      Albert       <NA>     <NA>      <NA>
## # … with 1,995 more rows
```

```
#tibble type 2, containing the number of concerts their location and time
information
tibble_2_concert <- concerts %>%
  select(concnr, podium, city, date, time, instrument1, instrument2,
instrument3, instrument4) %>%
  distinct()
tibble_2_concert
```

```
## # A tibble: 3,602 × 9
##    concnr podium               city  date  time  instr…¹ instr…² instr…³
instr…⁴
##     <int> <chr>                <chr> <chr> <chr> <chr>   <chr>   <chr>
<chr>
##  1     14 Paradox              Tilb… 1/3/… 21:0… unknown unknown unknown
unknown
##  2     24 Bimhuis              Amst… 1/5/… 21:0… unknown unknown unknown
unknown
##  3    462 Brouwershoeck, De    Leeu… 3/3/… 15:0… reets   unknown unknown
unknown
##  4    799 Bimhuis              Amst… 4/13… 21:0… reets   unknown unknown
unknown
##  5    910 Bimhuis              Amst… 4/27… 21:0… reets   unknown unknown
unknown
##  6   1298 Bimhuis              Amst… 6/29… 21:0… saxoph… clarin… unknown
unknown
##  7   1410 Paradiso             Amst… 8/17… 22:0… unknown unknown unknown
unknown
##  8   1443 Dordtse Jazz Sociët… Dord… 8/31… 22:0… clarin… barito… unknown
```

```
unknown
##  9   2776 <NA>                 Zwol… 9/7/… 23:0… clarin… unknown unknown
unknown
## 10   1663 Brouwershoeck, De    Leeu… 9/29… 15:0… unknown unknown unknown
unknown
## # … with 3,592 more rows, and abbreviated variable names ¹instrument1,
## #   ²instrument2, ³instrument3, ⁴instrument4

#tibble type 3, containing the number of band and further information
tibble_3_band <- concerts %>%
  select(bandnr, bandname, band_info) %>%
  distinct()
tibble_3_band

## # A tibble: 485 × 3
##    bandnr bandname                                band_info
##     <int> <chr>                                   <chr>
##  1      0 <none>                                  <NA>
##  2    326 ICP Orkest                              <NA>
##  3      6 Ab Baars Trio                           <NA>
##  4    240 Filiaal                                 <NA>
##  5    337 J.C. Tans Orchestra                     <NA>
##  6    691 Theo Loevendie Kwintet                  <NA>
##  7    719 Trio Ab Baars/Misha Mengelberg/Sunny Mur <NA>
##  8    942 Dutch Swing College Band                <NA>
##  9    188 Duo Aki Takase/Paul van Kemenade        <NA>
## 10      3 A Damn Stir                             <NA>
## # … with 475 more rows
```

###Answer: The three types of cases that are to be distinguished are: musician (musicnr, lastname, firstname, national, residence, country), band (bandnur, bandname, band_info), concert (concnr, podium, city, instrument1, instrument2, instrument3, instrument4, date, time). It is not optimal to have these units in one tibble, because it leads to duplicate information, which makes it difficult to read or to use the information presented in the dataset. By splitting this tibble based on the separate cases, the structure of the data is clear and there is no repetition of data.

Grading	Max points	Awarded
Ex. 1	2	

2. For one of the tables (tibbles) that you have created in Question 1, select a field that is a primary key in one table and a foreign key in another table. Motivate your selection and demonstrate that the variable is a primary key (in one table) and a foreign key (in the other table).

```
#Add your R code for answering this question here.

# Added the variables in the count() function that I explored to be primary
keys
concerts %>%
  count(concnr, musicnr) %>%
```

```
  # represent different types of cases (unit).
  filter(n > 1)
```

```
## # A tibble: 0 × 3
## # … with 3 variables: concnr <int>, musicnr <int>, n <int>
```

```
# Added the variables in the count() function that I explored to be a foreign
key
tibble_2_concert %>%
  count(concnr) %>%
  # represent different types of cases (unit).
  filter(n > 1)
```

```
## # A tibble: 569 × 2
##    concnr     n
##     <int> <int>
##  1     10     6
##  2     11     4
##  3     22     4
##  4     39     4
##  5     40     8
##  6     60     5
##  7     66     3
##  8     67     4
##  9     75     4
## 10     78     7
## # … with 559 more rows
```

###Answer: A primary key is the column or set of columns that uniquely identifies each observation in your dataset. If no combination of values appears at least once, this indicates the presence of a primary key. In the case of of the original concerts file, this was the combination of musicnr and concnr. This resulted in 0 rows, confirming the notion that it is a primary key.

A foreign key identifies an observation in another table. To test this, tibble_2_concert has been chosen, since concnr is present in both concerts and tibble_2_concert tables. In the case of tibble_2_concert, this was concnr, since the data is a primary key in concerts and is present in tibble_2_concert.

Grading	Max points	Awarded
Ex. 2	1	

3. The musicians performing at a concert are supposed to belong to the band that performs at this concert. How many musicians play in more than one band in this data set? Use the tables (tibbles) that you created for Question 1. (If that was not successful, use the original data). Note: Use your skills from the first week.

```
#Add your R code for answering this question here.

# Group data by musician
musician_bands <- concerts %>%
```

```
  group_by(musicnr) %>%
  summarise(n_bands = n_distinct(bandnr))

# Count how many musicians belong to more than one band
n_multi_band_musicians <- sum(musician_bands$n_bands > 1)

n_multi_band_musicians
```

```
## [1] 421
```

Answer:

The linked table has been used, since it contains musicnr and bandnr. Moreover, this ensures that the output is correct. According to the the original dataset, 421 musicians were playing in more than 1 band.

Grading	Max points	Awarded
Ex. 3	2	

4. A musician may play up to four instruments during a concert (variables `instrument1` to `instrument4`). Change these variables into factors and change the category "unknown" to "not applicable" for the second to fourth instrument. A musician may play less than four instruments during a concert, so the second (third, fourth) instrument do not have to be known.

```
#Add your R code for answering this question here.

concerts_Ex4<- concerts
# first the variables of instruments are transformed into factors
concerts[,c("instrument1", "instrument2", "instrument3", "instrument4")] <-
lapply(concerts[,c("instrument1", "instrument2", "instrument3",
"instrument4")], as.factor)

# instrument 2 category "unknown" is changed to "not applicable"
concerts_Ex4[concerts$instrument2 == "unknown", "instrument2"] <- "not
applicable"

# instrument 3 category "unknown" is changed to "not applicable"
concerts_Ex4[concerts$instrument3 == "unknown", "instrument3"] <- "not
applicable"

# instrument 4 category "unknown" is changed to "not applicable"
concerts_Ex4[concerts$instrument4 == "unknown", "instrument4"] <- "not
applicable"

concerts_Ex4
```

```
## # A tibble: 4,540 × 18
##    musicnr lastname firstname national resid…¹ country concnr podium city
date
##      <int> <chr>    <chr>     <chr>    <chr>   <chr>    <int> <chr>  <chr>
```

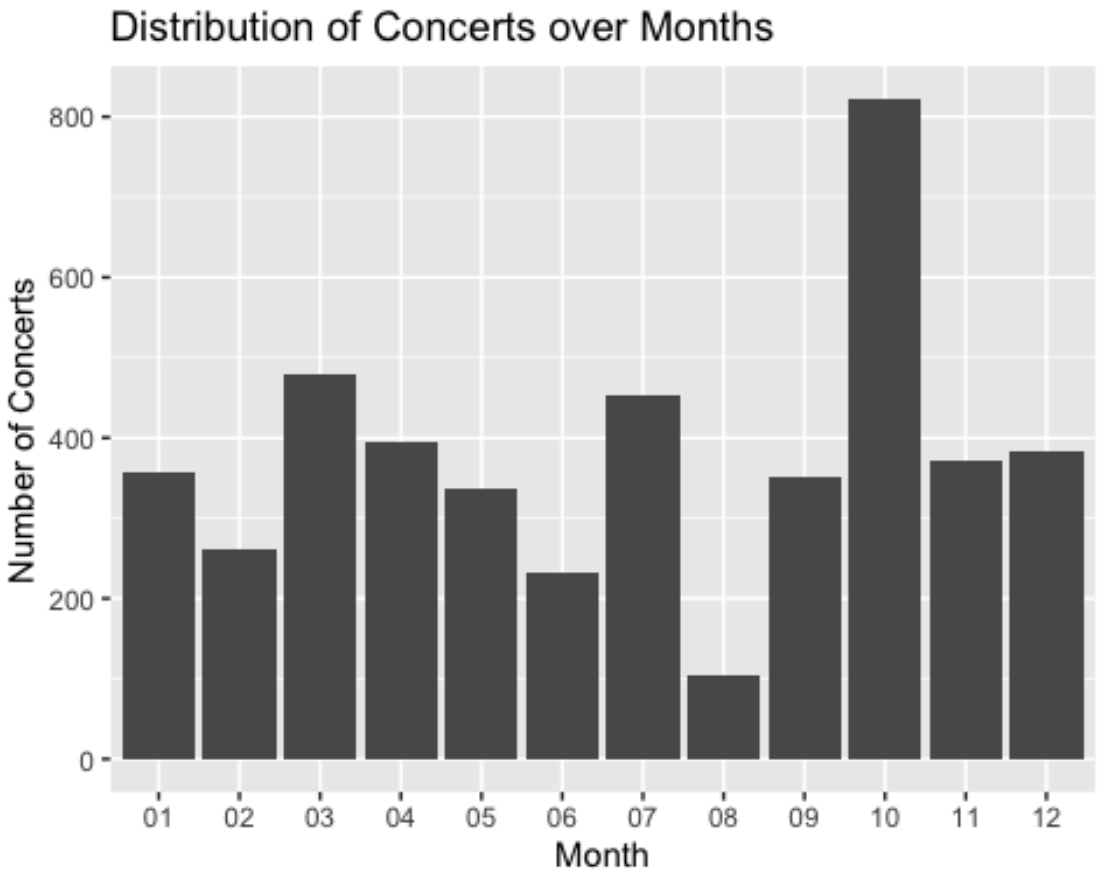
```
<chr>
##  1       2 Baars    Ab        Ned      Amster… Ned          14 Parad… Tilb…
1/3/…
##  2       2 Baars    Ab        Ned      Amster… Ned          24 Bimhu… Amst…
1/5/…
##  3       2 Baars    Ab        Ned      Amster… Ned         462 Brouw… Leeu…
3/3/…
##  4       2 Baars    Ab        Ned      Amster… Ned         799 Bimhu… Amst…
4/13…
##  5       2 Baars    Ab        Ned      Amster… Ned         910 Bimhu… Amst…
4/27…
##  6       2 Baars    Ab        Ned      Amster… Ned        1298 Bimhu… Amst…
6/29…
##  7       2 Baars    Ab        Ned      Amster… Ned        1410 Parad… Amst…
8/17…
##  8       2 Baars    Ab        Ned      Amster… Ned        1443 Dordt… Dord…
8/31…
##  9       2 Baars    Ab        Ned      Amster… Ned        2776 <NA>   Zwol…
9/7/…
## 10       2 Baars    Ab        Ned      Amster… Ned        1663 Brouw… Leeu…
9/29…
## # … with 4,530 more rows, 8 more variables: time <chr>, instrument1 <chr>,
## #   instrument2 <chr>, instrument3 <chr>, instrument4 <chr>, bandnr <int>,
## #   bandname <chr>, band_info <chr>, and abbreviated variable name ¹
residence
```

Grading	Max points	Awarded
Ex. 4	1	

5. Change the string variable `date` into a date variable and plot the distribution of concerts over months (grouped by month).

```
#Add your R code for answering this question here.


# Create new varialbe for month
concerts$month <- as.factor(format(mdy(concerts$date), "%m"))
# Plot distribution of concerts over months
ggplot(concerts, aes(x = month)) + geom_bar(stat = "count") + xlab("Month") +
ylab("Number of Concerts") + ggtitle("Distribution of Concerts over Months")
```



Grading	Max points	Awarded
Ex. 5	1	

6. For the selected table (tibble) in your *Data Project* (see below), spot one violation of the four rules for tidy data (see tutorial materials) and fix it with R code.

Project Data set	Use table (tibble)
nujij.nl	Article.csv
IT Call Center	Transactions.csv
Social Evolution	MusicGenrePreference.csv
Friends and Families	SurveyMonthly.2010_07.csv
Chancellor Debate	Debate 2009.csv
EU 2014 Election	Dataset MCA EPE 2014 NL FINAL.csv

```
#Add your R code for answering this question here.

MusicGenrePreference <- read_csv("MusicGenrePreference.csv")

## Rows: 264 Columns: 13
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
```

```
## chr  (11): indie / alternative rock, classic rock, heavy metal / hardcore,
p...
## dbl   (1): user.id
## 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.
```

```
#Tidying of the data
MGP_tidy <- MusicGenrePreference %>%
  pivot_longer(c("indie / alternative rock" : "other"), names_to =
"musictype", values_to = "interestlevel", values_drop_na = TRUE)

MGP_tidy
```

```
## # A tibble: 1,519 × 4
##    user.id date       musictype                   interestlevel
##      <dbl> <date>     <chr>                       <chr>
##  1      55 2008-09-19 classic rock                2 Moderate Interest
##  2      55 2008-09-19 heavy metal / hardcore      3 High Interest
##  3      55 2008-09-19 pop / top 40                2 Moderate Interest
##  4      55 2008-09-19 techno / lounge / electronic 3 High Interest
##  5      55 2008-09-19 hip-hop / r&b               2 Moderate Interest
##  6      55 2008-09-19 jazz                        1 Slight Interest
##  7      55 2008-09-19 classical                   1 Slight Interest
##  8      36 2008-09-19 indie / alternative rock    2 Moderate Interest
##  9      36 2008-09-19 pop / top 40                2 Moderate Interest
## 10      36 2008-09-19 techno / lounge / electronic 3 High Interest
## # … with 1,509 more rows
```

Answer:

The violation of the four rules for tidy data that can be observed in MusicGenrePreference.csv is that each observation must have its own row. In the original data set the different variables address the same information, stack repeated information and it does not allow values as column. Furthermore, 'values_drop_na = TRUE' is used in order to get rid of the variables that contain 'NA'. This prevents any further repetition or omitted variables to be excluded of the tibble.

Therefore, the the pivot_longer function is used in order to

Grading	Max points	Awarded
Ex. 6	2	
Flawless knitting	1	
Total	10	