Home Work Assignment 4

Pradeep Sahoo

7/1/2018

My Github repository for my assignments can be found at this URL: [My Github] (https://github.com/Pradeep-Sahoo/R-Assignments/tree/master/R_Project1)

```
library(mdsr)
library(tidyverse)
library(nycflights13)
library(tibble)
```

Section 5.6.7: #2, #4 and #6 only. Extra Credit: Do #5

2.

```
not_cancelled <- flights %>%
  filter(!is.na(dep_delay), !is.na(arr_delay))

not_cancelled %>% group_by(dest) %>% summarize(count_dest = n())
```

```
## # A tibble: 104 x 2
##    dest  count_dest
##    <chr>      <int>
##  1 ABQ          254
##  2 ACK          264
##  3 ALB          418
##  4 ANC            8
##  5 ATL        16837
##  6 AUS         2411
##  7 AVL          261
##  8 BDL          412
##  9 BGR          358
## 10 BHM          269
## # ... with 94 more rows
```

Come up with another approach that will give you the same output as count function counts the data set by a the specified variable. The same result can be achieved if we group by the data set by the same variable. Then using length() function counts the observation in each group

```
#equivalent of not_cancelled %>%   count(dest)
not_cancelled %>%
  group_by(dest) %>%
  summarise(n = length(dest))
```

```
## # A tibble: 104 x 2
##    dest      n
##    <chr> <int>
##  1 ABQ     254
##  2 ACK     264
```

```
##  3 ALB     418
##  4 ANC       8
##  5 ATL   16837
##  6 AUS    2411
##  7 AVL     261
##  8 BDL     412
##  9 BGR     358
## 10 BHM     269
## # ... with 94 more rows
```

```
#not_cancelled %>% count(tailnum, wt = distance)
not_cancelled %>%
 group_by(tailnum)%>%summarise(dist <- sum(distance))
```

```
## # A tibble: 4,037 x 2
##    tailnum `dist <- sum(distance)`
##    <chr>                     <dbl>
##  1 D942DN                     3418
##  2 N0EGMQ                   239143
##  3 N10156                   109664
##  4 N102UW                    25722
##  5 N103US                    24619
##  6 N104UW                    24616
##  7 N10575                   139903
##  8 N105UW                    23618
##  9 N107US                    21677
## 10 N108UW                    32070
## # ... with 4,027 more rows
```

3.

Our definition of cancelled flights (is.na(dep_delay) | is.na(arr_delay)) is slightly suboptimal. Why? Which is the most important column?

I think arr_delay is important. Because if the flight may go land in a different airport other than the Destination Airport. In that case the arr_delay would be NULL. So to know the about the Cancelled Flight both Dep_delay and Arr_delay has to be NULL

4.

Look at the number of cancelled flights per day. Is there a pattern? Is the proportion of cancelled flights related to the average delay?

Need to check with Robert.

```
#canceled_delayed <-
 # flights%>%
#group_by(flights, year, month, day)
```

Section 10.5: #1, #2, #3 and #6 only

1

Class function can be used to tell whether an object is a Tibble. For Ex:

```
class(mtcars)
```

```
## [1] "data.frame"
```

2

In the below example in a Data Frame we need not specify the whole name of the column to subset. We have just specified the x from xyz and it gives the value by $. This is sometime errorneous. If we have 2 column with similar names. In that case the result may be erroneous.

Also subsetting a single column with [,"column name"] returns a tibble from a tibble. where as a dataframe returns a vector. Following example suggests that.

```
df <- data.frame(abc = 1, xyz = "a")
df$x
```

```
## [1] a
## Levels: a
```

```
df1 <- data.frame(abc = 1, ab = 2, cde = 3)
df1$a
```

```
## NULL
```

```
df[, "xyz"]
```

```
## [1] a
## Levels: a
```

```
df[, c("abc", "xyz")]
```

```
##   abc xyz
## 1   1   a
```

```
tb <- as_tibble(df)
tb$x
```

```
## Warning: Unknown or uninitialised column: 'x'.
```

```
## NULL
```

```
tb[,"xyz"]
```

```
## # A tibble: 1 x 1
##   xyz
##   <fct>
## 1 a
```

```
#using $ in a Tibble returns the vector
tb$abc
```

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

```
tb$xyz
```

```
## [1] a
## Levels: a
```

```
tb[,c("abc","xyz")]
```

```
## # A tibble: 1 x 2
##     abc xyz
##   <dbl> <fct>
## 1     1 a
```

3

If you have the name of a variable stored in an object, e.g. var <- “mpg”, how can you extract the reference variable from a tibble?

We can extract the reference of the Variable by using [[]] Here Name of the variable is “mpg” which is stored in obj called var. We need to find the values of mpg.

```
var <- "mpg"
mtcars[[var]]
```

```
##  [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2
## [15] 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4
## [29] 15.8 19.7 15.0 21.4
```

6

What option controls how many additional column names are printed at the footer of a tibble? Need to check.

Section 12.3.3: #2, #3 and #4 only

2

```
table4a
```

```
## # A tibble: 3 x 3
##   country     `1999` `2000`
## * <chr>        <int>  <int>
## 1 Afghanistan    745   2666
## 2 Brazil       37737  80488
## 3 China       212258 213766
```

```
#table4a %>%
  #gather(1999, 2000, key = "year", value = "cases")
```

This is becaue the Variable 1999 and 2000 are starting with numbers, so we need to enclose them in ‘(tick marks).

```
table4a %>%
  gather(`1999`, `2000`, key = "year", value = "cases")
```

```
## # A tibble: 6 x 3
##   country     year   cases
##   <chr>       <chr>  <int>
## 1 Afghanistan 1999     745
## 2 Brazil      1999   37737
## 3 China       1999  212258
## 4 Afghanistan 2000    2666
## 5 Brazil      2000   80488
## 6 China       2000  213766
```

3

Why does spreading this tibble fail? How could you add a new column to fix the problem?

```
people <- tribble(
  ~name,             ~key,    ~value,
  #-----------------/--------/------
  "Phillip Woods",   "age",       45,
  "Phillip Woods",   "height",   186,
  "Phillip Woods",   "age",       50,
  "Jessica Cordero", "age",       37,
  "Jessica Cordero", "height",   156
)
people
```

```
## # A tibble: 5 x 3
##   name            key    value
##   <chr>           <chr>  <dbl>
## 1 Phillip Woods   age       45
## 2 Phillip Woods   height   186
## 3 Phillip Woods   age       50
## 4 Jessica Cordero age       37
## 5 Jessica Cordero height   156
```

This suggest that duplicate values are present. Same combination of Name and Key are appearing multiple times. Like Phillip Woods, age is coming muliple times. So the error, need to add one more variable like Year to break the duplicacy.

```
people <- tribble(
  ~name,             ~Year,     ~key,    ~value,
  #-----------------/--------/--------/------
  "Phillip Woods",   2010, "age",       45,
  "Phillip Woods",   2010, "height",   186,
  "Phillip Woods",   2015, "age",       50,
  "Jessica Cordero", 2010, "age",       37,
  "Jessica Cordero", 2010, "height",   156
)
people
```

```
## # A tibble: 5 x 4
##   name             Year key    value
```

```
##   <chr>           <dbl> <chr>  <dbl>
## 1 Phillip Woods    2010 age       45
## 2 Phillip Woods    2010 height   186
## 3 Phillip Woods    2015 age       50
## 4 Jessica Cordero  2010 age       37
## 5 Jessica Cordero  2010 height   156
```

```
spread(people,key = key, value = value)
```

```
## # A tibble: 3 x 4
##   name             Year   age height
##   <chr>           <dbl> <dbl>  <dbl>
## 1 Jessica Cordero  2010    37    156
## 2 Phillip Woods    2010    45    186
## 3 Phillip Woods    2015    50     NA
```

```
preg <- tribble(
  ~pregnant, ~male, ~female,
  "yes",     NA,    10,
  "no",      20,    12
)
preg %>% gather(male, female, key = gender, value = who_knows)
```

```
## # A tibble: 4 x 3
##   pregnant gender who_knows
##   <chr>    <chr>      <dbl>
## 1 yes      male          NA
## 2 no       male          20
## 3 yes      female        10
## 4 no       female        12
```

Section 12.4.3: #1 and #2 only

2

```
file_path = '/Users/pradeepsahoo/Downloads/baby_names.txt'
my_data <- read.delim(file_path,header = TRUE,sep  = '|')
glimpse(my_data)
```

```
## Observations: 30,000
## Variables: 5
## $ year <int> 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 188...
## $ sex  <fct> F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, ...
## $ name <fct> Mary, Anna, Emma, Elizabeth, Minnie, Margaret, Ida, Alice...
## $ n    <int> 7065, 2604, 2003, 1939, 1746, 1578, 1472, 1414, 1320, 128...
## $ prop <dbl> 0.072384329, 0.026679234, 0.020521700, 0.019865989, 0.017...
```

```
write.csv(my_data,file = '/Users/pradeepsahoo/Downloads/baby_names.csv')
csv_data <- read.csv( '/Users/pradeepsahoo/Downloads/baby_names.csv')
glimpse(csv_data)
```

```
## Observations: 30,000
## Variables: 6
```

```
## $ X    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17...
## $ year <int> 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 188...
## $ sex  <fct> F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, ...
## $ name <fct> Mary, Anna, Emma, Elizabeth, Minnie, Margaret, Ida, Alice...
## $ n    <int> 7065, 2604, 2003, 1939, 1746, 1578, 1472, 1414, 1320, 128...
## $ prop <dbl> 0.072384329, 0.026679234, 0.020521700, 0.019865989, 0.017...
```