

Assignment 4. Data Wrangling with Dplyr

Louis Dion

September 23, 2019

This assignment assumes that you have taken the **Introduction to the Tidyverse** and **Data Manipulation with dplyr in R** course at Datacamp. You can use base R functions and dplyr functions in the assignment.

Submission Instruction. You will need to submit on **Blackboard**, in the **Assignment** section, the follows:

- A knitted pdf
- A link to the markdown document in your Github
- A link to the pdf document in your Github

Questions

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##   filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

1. Compute the follows using `%>%` operator. **Notice that**

- `x %>% f = f(x)`,
- `x %>% f %>% g = g(f(x))` and
- `x %>% f(y) = f(x,y)`

a. `sin(2019)`

```
2019%>%sin
```

```
## [1] 0.8644605
```

b. `sin(cos(2019))`

```
2019%>%cos%>%sin
```

```
## [1] -0.4817939
```

```
c. sin(cos(tan(log(2019))))
```

```
2019%>%log%>%tan%>%cos%>%sin
```

```
## [1] -0.5939393
```

```
d. log2(2019)
```

```
2019%>%log2
```

```
## [1] 10.97943
```

2. Fixing the SEX, AGE and TRAV_SP following the steps in Assignment 2 (This time, do it on the entire dataset instead of the sample dataset). #Bring in dataset

```
library(readxl)
c2015<-read_excel('C:/Users/student/Documents/MATH421/data/c2015.xlsx')
```

```
#Change Unknown and Not Rep to NA
```

```
c2015<-c2015%>%replace(.,c2015=='Unknown'|c2015=='Not Rep',NA)
```

```
#Fix SEX variable
```

```
c2015<-c2015%>%mutate(SEX=replace(SEX,is.na(SEX),'Female'))
```

```
#Fix AGE variable
```

```
c2015<-c2015%>%mutate(AGE=replace(AGE,AGE=='Less than 1','0'),AGE=as.numeric(AGE),AGE=replace(AGE,is.na
```

```
#Fix TRAV_SP variable
```

```
library(stringr)
c2015<-c2015%>%mutate(TRAV_SP=apply(strsplit(TRAV_SP,split=" ",fixed = TRUE),function(x) (x[1])), TRAV
```

```
## Warning: NAs introduced by coercion
```

3. Calculate the average age and average speed of female in the accident happened in the weekend.

```
femweekend<-c2015%>%filter(SEX=='Female',DAY_WEEK=='Saturday'|DAY_WEEK=='Sunday')%>%summarize(age=mean(
femweekend
```

```
## # A tibble: 1 x 2
##   age speed
##   <dbl> <dbl>
## 1  36.4  50.2
```

Notice: These questions are to practice `select_if` and `summarise_if`, `summarise_all...` functions in `dplyr`. Check out the uses of these functions [here](#) and [here](#).

4. Use `select_if` and `is.numeric` functions to create a dataset with only numeric variables. Print out the names of all numeric variables

```
number<-c2015%>%select_if(is.numeric)
number%>%names
```

```
## [1] "ST_CASE" "VEH_NO" "PER_NO" "COUNTY" "DAY" "HOURL"
## [7] "MINUTE" "AGE" "YEAR" "TRAV_SP" "LATITUDE" "LONGITUD"
```

5. Calculate the mean of all numeric variables using `select_if` and `summarise_all`

```
number%>%summarize_all(.,mean,na.rm=1)
```

```
## # A tibble: 1 x 12
##   ST_CASE VEH_NO PER_NO COUNTY DAY HOUR MINUTE AGE YEAR TRAV_SP
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 275607. 1.39 1.63 91.7 15.5 14.0 28.4 39.1 2015 49.9
## # ... with 2 more variables: LATITUDE <dbl>, LONGITUD <dbl>
```

6. We can shortcut 3 and 4 by using `summarise_if`: Use `summarise_if` to Calculate the mean of all numeric variables. (You may need to use `na.rm = TRUE` to ignore the NAs)

```
c2015%>%summarise_if(is.numeric, mean, na.rm = TRUE)
```

```
## # A tibble: 1 x 12
##   ST_CASE VEH_NO PER_NO COUNTY DAY HOUR MINUTE AGE YEAR TRAV_SP
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 275607. 1.39 1.63 91.7 15.5 14.0 28.4 39.1 2015 49.9
## # ... with 2 more variables: LATITUDE <dbl>, LONGITUD <dbl>
```

7. Use `summarise_if` to calculate the median of all numeric variables.

```
c2015%>%summarise_if(is.numeric, median, na.rm = TRUE)
```

```
## # A tibble: 1 x 12
##   ST_CASE VEH_NO PER_NO COUNTY DAY HOUR MINUTE AGE YEAR TRAV_SP
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 270282 1 1 71 15 15 29 37 2015 53
## # ... with 2 more variables: LATITUDE <dbl>, LONGITUD <dbl>
```

8. Use `summarise_if` to calculate the standard deviation of all numeric variables. (`sd` function for standard deviation)

```
c2015%>%summarise_if(is.numeric, sd, na.rm = TRUE)
```

```
## # A tibble: 1 x 12
##   ST_CASE VEH_NO PER_NO COUNTY DAY HOUR MINUTE AGE YEAR TRAV_SP
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 163031. 1.45 1.84 95.0 8.78 9.06 17.3 20.1 0 20.9
## # ... with 2 more variables: LATITUDE <dbl>, LONGITUD <dbl>
```

9. Use `summarise_if` to calculate the number of missing values for each numeric variables. *Hint:* Use `~sum(is.na(.))`

```
c2015%>%summarise_if(is.numeric, ~sum(is.na(.)))
```

```
## # A tibble: 1 x 12
##   ST_CASE VEH_NO PER_NO COUNTY DAY HOUR MINUTE AGE YEAR TRAV_SP
##   <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1      0      0      0      0      0      0      377      0      0 54549
## # ... with 2 more variables: LATITUDE <int>, LONGITUD <int>
```

10. Calculate the log of the average for each numeric variable.

```
c2015%>%summarise_if(is.numeric, ~log(mean(., na.rm=TRUE)))
```

```
## Warning in log(mean(., na.rm = TRUE)): NaNs produced
```

```
## # A tibble: 1 x 12
##   ST_CASE VEH_NO PER_NO COUNTY DAY HOUR MINUTE AGE YEAR TRAV_SP
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 12.5 0.329 0.488 4.52 2.74 2.64 3.35 3.67 7.61 3.91
## # ... with 2 more variables: LATITUDE <dbl>, LONGITUD <dbl>
```

11. You will notice that there is one NA is produced in 10. Fix this by calculating the log of the absolute value average for each numeric variable.

```
c2015%>%summarise_if(is.numeric, ~log(abs(mean(., na.rm=TRUE))))
```

```
## # A tibble: 1 x 12
##   ST_CASE VEH_NO PER_NO COUNTY DAY HOUR MINUTE AGE YEAR TRAV_SP
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 12.5 0.329 0.488 4.52 2.74 2.64 3.35 3.67 7.61 3.91
## # ... with 2 more variables: LATITUDE <dbl>, LONGITUD <dbl>
```

12. Calculate the number of missing values for each categorical variables using `summarise_if`

```
c2015%>%summarise_if(is.character, ~sum(is.na(.)))
```

```
## # A tibble: 1 x 16
##   STATE MONTH   SEX PER_TYP INJ_SEV SEAT_POS DRINKING MAN_COLL OWNER
##   <int> <int> <int>   <int>   <int>   <int>   <int>   <int> <int>
## 1     0     0     0     0     770     716     0    7373  8826
## # ... with 7 more variables: MOD_YEAR <int>, DEFORMED <int>,
## #   DAY_WEEK <int>, ROUTE <int>, HARM_EV <int>, LGT_COND <int>,
## #   WEATHER <int>
```

13. Calculate the number of missing values for each categorical variables using `summarise_all`

```
c2015%>%select_if(is.character)%>%summarize_all(~sum(is.na(.)))
```

```
## # A tibble: 1 x 16
##   STATE MONTH   SEX PER_TYP INJ_SEV SEAT_POS DRINKING MAN_COLL OWNER
##   <int> <int> <int>   <int>   <int>   <int>   <int>   <int> <int>
## 1     0     0     0     0     770     716     0    7373  8826
## # ... with 7 more variables: MOD_YEAR <int>, DEFORMED <int>,
## #   DAY_WEEK <int>, ROUTE <int>, HARM_EV <int>, LGT_COND <int>,
## #   WEATHER <int>
```

14. Calculate the number of states in the dataset. ****Hint:** You can use `length(table())`

```
length(table(c2015$STATE))
```

```
## [1] 51
```

```
#There are 51 "States" in the dataset, the 50 actual states and presumably Puerto Rico.
```

15. Calculate the number of uniques values for each categorical variables using `summarise_if`.

```
c2015%>%summarize_if(is.character,~length(table(.)))
```

```
## # A tibble: 1 x 16
##   STATE MONTH   SEX PER_TYP INJ_SEV SEAT_POS DRINKING MAN_COLL OWNER
##   <int> <int> <int>   <int>   <int>   <int>   <int>   <int> <int>
## 1    51    12     2    11     7     28     4     10     7
## # ... with 7 more variables: MOD_YEAR <int>, DEFORMED <int>,
## #   DAY_WEEK <int>, ROUTE <int>, HARM_EV <int>, LGT_COND <int>,
## #   WEATHER <int>
```

16. Calculate the number of uniques values for each categorical variables using `summarise_all`.

```
c2015%>%select_if(is.character)%>%summarize_all(~length(table(.)))
```

```
## # A tibble: 1 x 16
##   STATE MONTH   SEX PER_TYP INJ_SEV SEAT_POS DRINKING MAN_COLL OWNER
##   <int> <int> <int>   <int>   <int>   <int>   <int>   <int> <int>
## 1    51    12     2    11     7     28     4     10     7
## # ... with 7 more variables: MOD_YEAR <int>, DEFORMED <int>,
## #   DAY_WEEK <int>, ROUTE <int>, HARM_EV <int>, LGT_COND <int>,
## #   WEATHER <int>
```

17. Print out the names of all variables that have more than 30 distinct values

```
thirty<-c2015%>%select_if(~length(table(.))>30)
thirty%>%names
```

```
## [1] "STATE"      "ST_CASE"    "VEH_NO"     "PER_NO"     "COUNTY"    "DAY"
## [7] "MINUTE"     "AGE"        "MOD_YEAR"   "TRAV_SP"    "LATITUDE"   "LONGITUD"
## [13] "HARM_EV"
```

18. Print out the names of all categorical variables that more than 30 distinct values

```
thirty%>%select_if(is.character)%>%names
```

```
## [1] "STATE"      "MOD_YEAR"   "HARM_EV"
```

19. Print out the names of all numeric variables that has the maximum values greater than 30

```
#using number dataset created above
number%>%summarize_all(~max(.,na.rm=TRUE))
```

```
## # A tibble: 1 x 12
##   ST_CASE VEH_NO PER_NO COUNTY DAY HOUR MINUTE AGE YEAR TRAV_SP
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  560130    58    51   999   31   99    59   114  2015    150
## # ... with 2 more variables: LATITUDE <dbl>, LONGITUD <dbl>
```

#saw hour has value of 99, going to replace that with NA before printing names since there is no hour 9

```
number<-number%>%mutate(HOUR=replace(HOUR,HOUR==99,NA))
number%>%select_if(~max(.,na.rm=TRUE)>30)%>%names
```

```
## [1] "ST_CASE"    "VEH_NO"     "PER_NO"     "COUNTY"    "DAY"        "MINUTE"
## [7] "AGE"        "YEAR"       "TRAV_SP"    "LATITUDE"
```

20. Calculate the mean of all numeric variables that has the maximum values greater than 30 using 'summarise_if'

```
number%>%summarize_if(~max(.,na.rm=TRUE)>30,mean,na.rm=TRUE)
```

```
## # A tibble: 1 x 10
##   ST_CASE VEH_NO PER_NO COUNTY DAY MINUTE AGE YEAR TRAV_SP LATITUDE
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 275607.  1.39  1.63  91.7 15.5  28.4 39.1 2015  49.9  36.5
```

21. Calculate the mean of all numeric variables that has the maximum values greater than 30 using 'summarise_all'

```
number%>%select_if(~max(.,na.rm=TRUE)>30)%>%summarize_all(~mean(.,na.rm=TRUE))
```

```
## # A tibble: 1 x 10
##   ST_CASE VEH_NO PER_NO COUNTY DAY MINUTE AGE YEAR TRAV_SP LATITUDE
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 275607. 1.39 1.63 91.7 15.5 28.4 39.1 2015 49.9 36.5
```

22. Create a dataset containing variables with standard deviation greater than 10. Call this data d1

```
d1<-number%>%select_if(~sd(.,na.rm=TRUE)>10.0)
d1
```

```
## # A tibble: 80,587 x 6
##   ST_CASE COUNTY MINUTE AGE TRAV_SP LONGITUD
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 10001 127 40 68 55 -87.3
## 2 10002 83 13 49 70 -86.9
## 3 10003 11 25 31 80 -85.8
## 4 10003 11 25 20 80 -85.8
## 5 10004 45 57 40 75 -85.5
## 6 10005 45 9 24 15 -85.5
## 7 10005 45 9 60 65 -85.5
## 8 10006 111 59 64 45 -85.4
## 9 10006 111 59 17 45 -85.4
## 10 10007 89 33 80 NA -86.5
## # ... with 80,577 more rows
```

23. Centralizing a variable is subtract it by its mean. Centralize the variables of d1 using mutate_all. Check the means of all centralized variables to confirm that they are all zeros.

```
d1<-d1%>%mutate_all(.,~(.-mean(.,na.rm=TRUE)))
d1%>%summarize_all(~mean(.,na.rm=TRUE))
```

```
## # A tibble: 1 x 6
##   ST_CASE COUNTY MINUTE AGE TRAV_SP LONGITUD
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 4.73e-11 1.32e-14 -1.25e-15 1.58e-15 3.25e-15 -6.92e-15
```

```
#all means in the dataset are at 0.
```

24. Standardizing a variable is to subtract it to its mean and then divide by its standard deviation. Standardize the variables of d1 using mutate_all. Check the means and standard deviation of all centralized variables to confirm that they are all zeros (for the means) and ones (for standard deviation).

```
d1<-d1%>%mutate_all(.,~(./sd(.,na.rm=TRUE)))
d1%>%summarize_all(~mean(.,na.rm=TRUE))
```

```
## # A tibble: 1 x 6
##   ST_CASE COUNTY MINUTE AGE TRAV_SP LONGITUD
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 -9.97e-17 1.15e-16 -6.85e-17 8.49e-17 1.57e-16 -3.50e-16
```

```
#all means are 0  
d1%>%summarize_all(~sd(.,na.rm=TRUE))
```

```
## # A tibble: 1 x 6  
##   ST_CASE COUNTY MINUTE   AGE TRAV_SP LONGITUD  
##   <dbl>   <dbl>   <dbl> <dbl>   <dbl>   <dbl>  
## 1     1.000  1.000     1. 1.000   1.000     1.
```

```
#all standard deviations are 1
```