R- Assignment4

Payton Kim

9/29/2019

1. Compute the following using %>% operator. Notice that • x %>% f = f(x), • x %>% f %>% g = g(f(x)) and • x %>% f(y) = f(x,y)

a. sin(2019)
b. sin(cos(2019))
c. sin(cos(tan(log(2019))))
d. log2(2019)

```
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
```

```
#a.
a <- 2019 %>% sin()
a
```

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

```
#b
b <- a %>% cos()
b
```

```
## [1] 0.6490506
```

```
#c
c <- b %>% tan() %>% log()
c
```

```
## [1] -0.2761391
```

```
#d
log2(2019)
```

```
## [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).

```r
library(readxl)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v readr   1.3.1
## v tibble  2.1.3     v purrr   0.3.2
## v tidyr   1.0.0     v stringr 1.4.0
## v ggplot2 3.2.1     v forcats 0.4.0

## -- Conflicts ---------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
c <- read_excel("C:/Users/student/Downloads/c2015.xlsx")
class(c)
```

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

```r
c$SEX <- ifelse(c$SEX == "Unknown","Female",c$SEX)
c$AGE <- ifelse(c$AGE == "Less than 1","0", c$AGE)
c$AGE <- as.numeric(c$AGE)
```

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

```r
mean <- mean(c$AGE,na.rm = TRUE)
c$AGE <- ifelse(is.na(c$AGE),mean, c$AGE)
library(stringr)
c$TRAV_SP <- str_replace(c$TRAV_SP, " MPH","")
c$TRAV_SP <- str_replace(c$TRAV_SP, "No Rep","")
c$TRAV_SP <- str_replace(c$TRAV_SP, "Unknown","")
c$TRAV_SP <- as.numeric(c$TRAV_SP)
```

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

```r
c = c[!(is.na(c$TRAV_SP)),]
```

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

```r
c %>%
  filter(SEX == "Female", DAY_WEEK == c("Saturday", "Sunday")) %>%
  summarize(avgage = mean(AGE, na.rm = 1), avgspeed = mean(TRAV_SP, na.rm = 1))
```

```
## # A tibble: 1 x 2
##   avgage avgspeed
##    <dbl>    <dbl>
## 1   35.9     50.2
```

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

```r
nc <- select_if(c, is.numeric)
names(nc)
```

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

```r
head(nc)
```

```
## # A tibble: 6 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   10001      1      1    127     1     2     40    68  2015      55
## 2   10002      1      1     83     1    22     13    49  2015      70
## 3   10003      1      1     11     1     1     25    31  2015      80
## 4   10003      1      2     11     1     1     25    20  2015      80
## 5   10004      1      1     45     4     0     57    40  2015      75
## 6   10005      1      1     45     7     7      9    24  2015      15
## # ... with 2 more variables: LATITUDE <dbl>, LONGITUD <dbl>
```

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

```r
summarise_all(select_if(c, is.numeric), list(mean = ~mean(.,na.rm = 1)))
```

```
## # A tibble: 1 x 12
##   ST_CASE_mean VEH_NO_mean PER_NO_mean COUNTY_mean DAY_mean HOUR_mean
##          <dbl>       <dbl>       <dbl>       <dbl>    <dbl>     <dbl>
## 1      250204.        1.49        1.66        74.2     15.5      13.8
## # ... with 6 more variables: MINUTE_mean <dbl>, AGE_mean <dbl>,
## #   YEAR_mean <dbl>, TRAV_SP_mean <dbl>, LATITUDE_mean <dbl>,
## #   LONGITUD_mean <dbl>
```

6. 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)

```r
c %>%
summarise_if(is.numeric, ~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 250204.   1.49   1.66   74.2  15.5  13.8   28.8  38.7  2015    49.9
## # ... with 2 more variables: LATITUDE <dbl>, LONGITUD <dbl>
```

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

```r
c %>%
  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 220376.      1      1     67    15    15     30    36  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

```
c %>%
  summarize_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 170029.   1.26   1.68   72.5  8.79  7.70   17.4  20.3     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(.))

```
c %>%
  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     43     0     0       0
## # ... with 2 more variables: LATITUDE <int>, LONGITUD <int>
```

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

```
c %>%
  summarise_if(is.numeric, ~mean(.,na.rm = 1)) %>%
  log()
```

```
## Warning in FUN(X[[i]], ...): 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    12.4  0.397  0.507   4.31  2.74  2.63   3.36  3.66  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.

```
c %>%
  summarise_if(is.numeric, ~mean(.,na.rm = 1)) %>%
  abs() %>%
  log()
```

```
## # 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    12.4  0.397  0.507   4.31  2.74  2.63   3.36  3.66  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

```
c %>%
  summarize_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       0        0        0        0     0
## # ... 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

```
summarize_all(select_if(c, 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       0        0        0        0     0
## # ... 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())

```
c %>%
  summarize_at(vars(STATE), ~length(table(.)))
```

```
## # A tibble: 1 x 1
##   STATE
##   <int>
## 1    51
```

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

```
c %>%
  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     3       3       8       26        4       10     8
## # ... 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.

```
summarize_all(select_if(c, 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     3       3       8       26        4       10     8
## # ... 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

```
summarize_all(select_if(c, ~length(table(.))>30), ~length(table(.)))
```

```
## # A tibble: 1 x 12
##   STATE ST_CASE PER_NO COUNTY   DAY MINUTE   AGE MOD_YEAR TRAV_SP LATITUDE
##   <int>   <int>  <int>  <int> <int>  <int> <int>    <int>   <int>    <int>
## 1    51   12313     46    175    31     60   102       64     130    12243
## # ... with 2 more variables: LONGITUD <int>, HARM_EV <int>
```

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

```
c %>%
  select_if(is.character) %>%
  select_if(~length(table(.))>30) %>% names
```

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

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

```
c %>%
  select_if(is.numeric) %>%
  select_if(~max(., na.rm = TRUE)>30) %>% names
```

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

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

```
c %>%
  select_if(is.numeric) %>%
summarize_if(~max(., na.rm = TRUE)>30, ~mean(., na.rm = 1))
```

```
## # A tibble: 1 x 11
##   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 250204.   1.49   1.66   74.2  15.5  13.8   28.8  38.7  2015    49.9
## # ... with 1 more variable: LATITUDE <dbl>
```

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

```
c %>%
  select_if(is.numeric) %>%
  select_if(~max(., na.rm = TRUE)>30) %>%
  summarize_all( ~mean(., na.rm = 1))
```

```
## # A tibble: 1 x 11
##   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 250204.   1.49   1.66   74.2  15.5  13.8   28.8  38.7  2015    49.9
## # ... with 1 more variable: LATITUDE <dbl>
```

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

```
d1 <- c %>%
  select_if(is.numeric) %>%
  select_if(~sd(., na.rm = TRUE)>10)
head(d1)
```

```
## # A tibble: 6 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
```

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 %>%
  mutate_all(~(.) - mean(.,na.rm = TRUE)) %>%
  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 1.91e-11 6.38e-15 -4.86e-16 -1.45e-15 3.25e-15 -1.66e-15
```

24. Standarizing 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 %>%
  mutate_all(~(.) - mean(.,na.rm = TRUE)) %>%
  mutate_all(~(.)/sd(., na.rm = 1)) %>%
  summarize_all(~mean(.,na.rm = 1))
```

```
## # A tibble: 1 x 6
##     ST_CASE   COUNTY    MINUTE       AGE  TRAV_SP  LONGITUD
##       <dbl>    <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 -3.27e-17 6.03e-17 -3.19e-17 -7.27e-17 1.57e-16 -7.66e-17
```

```
d1 %>%
  mutate_all(~(.) - mean(.,na.rm = TRUE)) %>%
  mutate_all(~(.)/sd(., na.rm = 1)) %>%
  summarize_all(~sd(.,na.rm = 1))
```

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