Assignment 4

Kyle Walker

9/24/2019

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)

```
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
```

```
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).

```
library(readxl)
library(stringr)
c2015 <- read_excel("C:/Users/student/Documents/Senior Year/MATH 421/Assignment 2/c2015.xlsx")
#Fixing Sex
c2015$SEX[is.na(c2015$SEX)]<- "Female"
#Fixing Age
c2015$AGE[c2015$AGE == 'Less than 1'] <- "0"
c2015$AGE <- as.numeric(c2015$AGE)
```

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

```
c2015$AGE[is.na(c2015$AGE)] <- mean(c2015$AGE)
#Fixing Trav_Sp
c2015$TRAV_SP <- str_replace(c2015$TRAV_SP, " MPH", "")
c2015$TRAV_SP <- str_replace(c2015$TRAV_SP, "Not Rep", "")
c2015$TRAV_SP <- str_replace(c2015$TRAV_SP, "Unknown", "")
c2015$TRAV_SP <- as.numeric(c2015$TRAV_SP)
```

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

```
c2015 = c2015[!(is.na(c2015$TRAV_SP)),]
```

####3. Calculate the average age and average speed of female in the accident happened in the weekend.
Notice: These questions are to practice select_if and summarise_if, summarise_all. . . functions in dplyr

```
c2015 %>%
  filter(SEX == "Female", DAY_WEEK %in% c("Friday", "Saturday", "Sunday")) %>%
    summarize_at(vars(AGE, TRAV_SP), mean, na.rm=T)
```

```
## # A tibble: 1 x 2
##     AGE TRAV_SP
##   <dbl>   <dbl>
## 1  36.5    49.4
```

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

```
c2015 %>%
  select_if(is.numeric) %>%
    names
```

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

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

```
c2015 %>%
  select_if(is.numeric) %>%
    summarize_all(mean, na.rm = T)
```

```
## # 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>
```

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 %>%
  summarize_if(is.numeric, mean, na.rm=T)
```

```
## # 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.

```
c2015 %>%
  summarize_if(is.numeric, median, na.rm=T)
```

```
## # 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    35  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 %>%
  summarize_if(is.numeric, sd, na.rm=T)
```

```
## # 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(.))

```
c2015 %>%
  summarize_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   226     0       0
## # ... with 2 more variables: LATITUDE <int>, LONGITUD <int>
```

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

```
c2015 %>%
  summarize_if(is.numeric, mean, na.rm=T) %>%
    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.

```
c2015 %>%
  summarize_if(is.numeric, mean, na.rm=T) %>%
    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

```
c2015 %>%
  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

```
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       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())

```
c2015 %>%
  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.

```
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     4       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.

```
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     4       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

```
c2015 %>%
  summarize_all(~length(table(.))>30) %>%
    names
```

```
##  [1] "STATE"    "ST_CASE"  "VEH_NO"   "PER_NO"   "COUNTY"   "DAY"
##  [7] "MONTH"    "HOUR"     "MINUTE"   "AGE"      "SEX"      "PER_TYP"
## [13] "INJ_SEV"  "SEAT_POS" "DRINKING" "YEAR"     "MAN_COLL" "OWNER"
## [19] "MOD_YEAR" "TRAV_SP"  "DEFORMED" "DAY_WEEK" "ROUTE"    "LATITUDE"
## [25] "LONGITUD" "HARM_EV"  "LGT_COND" "WEATHER"
```

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

```
c2015 %>%
  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

```
c2015 %>%
  select_if(is.numeric) %>%
    select_if(~max(.,na.rm=T) > 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'

```
c2015 %>%
    select_if(~max(., na.rm=T) > 30) %>%
      summarize_if(is.numeric, mean, na.rm = T)
```

```
## # 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'

```
 c2015 %>%
  select_if(is.numeric) %>%
    select_if(~max(.,na.rm=T) > 30) %>%
      summarize_all(~mean(.,na.rm=T))
```

```
## # 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 <- c2015 %>%
    select_if(is.numeric) %>%
      select_if(~sd(.,na.rm=T) >10)
d1
```

```
## # A tibble: 26,038 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   10010     33     45    80      30    -87.6
## # ... with 26,028 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 %>%
  select_if(is.numeric) %>%
    mutate_all(~(.) - mean(.,na.rm=T)) %>%
      summarize_all(~mean(.,na.rm=T))
```

```
## # 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 -3.51e-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).

```
#TRY TO DO IN ONE SUMMARIZE WITH A LIST
d1 %>%
  select_if(is.numeric) %>%
    mutate_all(~(. - mean(.,na.rm=T))/ sd(.,na.rm=T)) %>%
      summarize_all(~mean(.,na.rm=T))
```

```
## # A tibble: 1 x 6
##     ST_CASE   COUNTY    MINUTE       AGE  TRAV_SP  LONGITUD
##       <dbl>    <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 -3.27e-17 5.49e-17 -3.19e-17 -1.79e-16 1.57e-16 -7.66e-17
```

```
d1 %>%
  select_if(is.numeric) %>%
    mutate_all(~(. - mean(.,na.rm=T))/ sd(.,na.rm=T)) %>%
      summarize_all(~sd(.,na.rm=T))
```

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