Assignment 4

Huiyi Huang

10/9/2019

```r
library(readxl)
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
```

```r
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
library(stringr)
```

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

(2019)

```r
#1
2019 %>% sin()
```

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

```
2019 %>% cos() %>% sin()
```

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

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

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

```
2019 %>% log(2)
```

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

```
#2 Fixing the SEX, AGE and TRAV_SP following the steps in Assignment 2 (This time, do it on the entire
c2015=read_excel("C:/Users/student/Documents/Fall2019/c2015.xlsx")
y =c2015 %>% #repllace NA in SEX into "Female"
  mutate(SEX = replace(SEX,SEX == "Unknown","Female")) %>%
  mutate(SEX = replace(SEX,SEX == "Not Rep","Female"))
```

```
#Fix variable age
y1 = y %>%
  mutate(AGE = replace(AGE, AGE == "Less than 1" , "0")) %>%
  mutate(AGE = as.numeric(AGE))%>%
  mutate(AGE = replace(AGE,is.na(AGE),mean(AGE,na.rm=TRUE)))
```

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

```
#Fix variable travel speed
y1=y1 %>%
  mutate(TRAV_SP1=str_replace(TRAV_SP," MPH","")) %>%
  mutate(TRAV_SP1 = as.numeric(TRAV_SP1))%>%
  mutate(TRAV_SP1 = replace(TRAV_SP1,is.na(TRAV_SP1),mean(TRAV_SP1,na.rm=TRUE)))
```

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

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

y1 %>%
  group_by(DAY_WEEK) %>%
  filter(SEX=="Female") %>%
  summarise(m_a=mean(AGE)) %>%
  summarise(m_a_wend=(36.33485+36.48690)/2)
```

```
## # A tibble: 1 x 1
##   m_a_wend
##      <dbl>
## 1     36.4
```

```
y1 %>%
  filter(SEX=="Female") %>%
  group_by(DAY_WEEK) %>%
  summarise(m_s=mean(TRAV_SP1)) %>%
  summarise(m_s_wend=(49.43715  +50.57528   )/2)
```

```
## # A tibble: 1 x 1
##   m_s_wend
##      <dbl>
## 1     50.0
```

```
#4. Use select_if and is.numeric functions to create a dataset with only numeric variables. Print out t
y1 %>%
  select_if(is.numeric) %>%
  glimpse
```

```
## Observations: 80,587
## Variables: 12
## $ ST_CASE  <dbl> 10001, 10002, 10003, 10003, 10004, 10005, 10005, 1000...
## $ VEH_NO   <dbl> 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 0, 1, 1, 1, 2, 1, 2,...
## $ PER_NO   <dbl> 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1,...
## $ COUNTY   <dbl> 127, 83, 11, 11, 45, 45, 45, 111, 111, 89, 89, 73, 73...
## $ DAY      <dbl> 1, 1, 1, 1, 4, 7, 7, 8, 8, 8, 8, 3, 3, 13, 5, 5, 7, 7...
## $ HOUR     <dbl> 2, 22, 1, 1, 0, 7, 7, 9, 9, 18, 18, 21, 21, 8, 18, 18...
## $ MINUTE   <dbl> 40, 13, 25, 25, 57, 9, 9, 59, 59, 33, 33, 30, 30, 0, ...
## $ AGE      <dbl> 68.00000, 49.00000, 31.00000, 20.00000, 40.00000, 24....
## $ YEAR     <dbl> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015,...
## $ LATITUDE <dbl> 33.87865, 34.91044, 32.14201, 32.14201, 31.43981, 31....
## $ LONGITUD <dbl> -87.32533, -86.90871, -85.75846, -85.75846, -85.51030...
## $ TRAV_SP1 <dbl> 55.00000, 70.00000, 80.00000, 80.00000, 75.00000, 15....
```

```
#5. Calculate the mean of all numeric variables using select_if and summarise_all
y1 %>%
  select_if(is.numeric) %>%
  summarise_all(mean,na.rm=TRUE)
```

```
## # A tibble: 1 x 12
##   ST_CASE VEH_NO PER_NO COUNTY   DAY  HOUR MINUTE   AGE  YEAR LATITUDE
##     <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     36.5
## # ... with 2 more variables: LONGITUD <dbl>, TRAV_SP1 <dbl>
```

```
#6. We can shortcut 3 and 4 by using summarise_if: Use summarise_if to Calculate the mean of all numeri
y1 %>%
  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 LATITUDE
##     <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     36.5
## # ... with 2 more variables: LONGITUD <dbl>, TRAV_SP1 <dbl>
```

```
#7. Use summarise_if to calculate the median of all numeric variables.
y1 %>%
  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 LATITUDE
##     <dbl>  <dbl>  <dbl>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>    <dbl>
## 1  270282      1      1     71    15    15     29    37  2015     36.2
## # ... with 2 more variables: LONGITUD <dbl>, TRAV_SP1 <dbl>
```

```
#8. Use summarise_if to calculate the standard deviation of all numeric variables. (sd function for sta
y1 %>%
  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 LATITUDE
##     <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     5.25
## # ... with 2 more variables: LONGITUD <dbl>, TRAV_SP1 <dbl>
```

```
#9. Use summarise_if to calculate the number of missing values for each numeric variables. Hint: Use ~s
y1 %>%
  summarize_if(is.numeric, ~sum(is.na(.)))
```

```
## # A tibble: 1 x 12
##   ST_CASE VEH_NO PER_NO COUNTY   DAY  HOUR MINUTE   AGE  YEAR LATITUDE
##     <int>  <int>  <int>  <int> <int> <int>  <int> <int> <int>    <int>
## 1       0      0      0      0     0     0    377     0     0      479
## # ... with 2 more variables: LONGITUD <int>, TRAV_SP1 <int>
```

```
#10. Calculate the log of the average for each numeric variable.
y1 %>%
  summarize_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 LATITUDE
##     <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     3.60
## # ... with 2 more variables: LONGITUD <dbl>, TRAV_SP1 <dbl>
```

```
#11. You will notice that there is one NA is produced in 10. Fix this by calculating the log of the abs
y1 %>%
  summarize_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 LATITUDE
##     <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     3.60
## # ... with 2 more variables: LONGITUD <dbl>, TRAV_SP1 <dbl>
```

```r
#12. Calculate the number of missing values for each categorical variables using summarise_if
y1 %>%
  summarize_if(is.character, ~sum(is.na(.)))
```

```
## # A tibble: 1 x 17
##   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     7197  7197
## # ... with 8 more variables: MOD_YEAR <int>, TRAV_SP <int>,
## #   DEFORMED <int>, DAY_WEEK <int>, ROUTE <int>, HARM_EV <int>,
## #   LGT_COND <int>, WEATHER <int>
```

```r
#13. Calculate the number of missing values for each categorical variables using summarise_all
y1 %>%
  select_if(is.character) %>%
  summarize_all(~sum(is.na(.)))
```

```
## # A tibble: 1 x 17
##   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     7197  7197
## # ... with 8 more variables: MOD_YEAR <int>, TRAV_SP <int>,
## #   DEFORMED <int>, DAY_WEEK <int>, ROUTE <int>, HARM_EV <int>,
## #   LGT_COND <int>, WEATHER <int>
```

```r
#14. Calculate the number of states in the dataset. **Hint: You can use length(table())
y1 %>% select(STATE) %>% table %>% length
```

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

```r
#dont need to do this in pipe
```

```r
#15. Calculate the number of uniques values for each categorical variables using summarise_if.
y1 %>%
  summarise_if(is.character, ~sum(is.na(.)))
```

```
## # A tibble: 1 x 17
##   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     7197  7197
## # ... with 8 more variables: MOD_YEAR <int>, TRAV_SP <int>,
## #   DEFORMED <int>, DAY_WEEK <int>, ROUTE <int>, HARM_EV <int>,
## #   LGT_COND <int>, WEATHER <int>
```

```r
#16. Calculate the number of uniques values for each categorical variables using summarise_all.
y1 %>%
  summarise_all(~sum(is.na(.)),is.character)
```

```
## # A tibble: 1 x 29
##   STATE ST_CASE VEH_NO PER_NO COUNTY   DAY MONTH  HOUR MINUTE   AGE   SEX
##   <int>   <int>  <int>  <int>  <int> <int> <int> <int>  <int> <int> <int>
## 1     0       0      0      0      0     0     0     0    377     0     0
## # ... with 18 more variables: PER_TYP <int>, INJ_SEV <int>,
## #   SEAT_POS <int>, DRINKING <int>, YEAR <int>, MAN_COLL <int>,
## #   OWNER <int>, MOD_YEAR <int>, TRAV_SP <int>, DEFORMED <int>,
## #   DAY_WEEK <int>, ROUTE <int>, LATITUDE <int>, LONGITUD <int>,
## #   HARM_EV <int>, LGT_COND <int>, WEATHER <int>, TRAV_SP1 <int>
```

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

y1%>% select_if(~length(table(.))>30) %>% names
```

```
##  [1] "STATE"    "ST_CASE"  "VEH_NO"   "PER_NO"   "COUNTY"   "DAY"
##  [7] "MINUTE"   "AGE"      "MOD_YEAR" "TRAV_SP"  "LATITUDE" "LONGITUD"
## [13] "HARM_EV"  "TRAV_SP1"
```

```
#18. Print out the names of all categorical variables that more than 30 distinct values
y1 %>%
  summarise_if(is.character, ~length(table(.))>30) %>% names
```

```
##  [1] "STATE"    "MONTH"    "SEX"      "PER_TYP"  "INJ_SEV"  "SEAT_POS"
##  [7] "DRINKING" "MAN_COLL" "OWNER"    "MOD_YEAR" "TRAV_SP"  "DEFORMED"
## [13] "DAY_WEEK" "ROUTE"    "HARM_EV"  "LGT_COND" "WEATHER"
```

```
#19. Print out the names of all numeric variables that has the maximum values greater than 30
y1 %>%
  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"     "LATITUDE" "TRAV_SP1"
```

```
#20. Calculate the mean of all numeric variables that has the maximum values greater than 30 using ‘sum
y1 %>%
  summarise_if(is.numeric,~mean(.,na.rm=TRUE),~max(.)>30)
```

```
## # A tibble: 1 x 12
##   ST_CASE VEH_NO PER_NO COUNTY   DAY  HOUR MINUTE   AGE  YEAR LATITUDE
##     <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     36.5
## # ... with 2 more variables: LONGITUD <dbl>, TRAV_SP1 <dbl>
```

```
#21. Calculate the mean of all numeric variables that has the maximum values greater than 30 using ‘sum
y1 %>%
  select_if(is.numeric) %>%
  select_if(~max(., na.rm=TRUE)>30) %>%

  summarise_all(~mean(.,na.rm=TRUE))
```

```
## # A tibble: 1 x 11
##   ST_CASE VEH_NO PER_NO COUNTY   DAY  HOUR MINUTE   AGE  YEAR LATITUDE
##     <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     36.5
## # ... with 1 more variable: TRAV_SP1 <dbl>
```

```r
#22. Create a dataset containing variables with standard deviation greater than 10. Call this data d1
d1=y1%>%
  select_if(is.numeric) %>%
  select_if(~sd(.,na.rm=TRUE)>10)
```

```r
#23. Centralizing a variable is subtract it by its mean. Centralize the variables of d1 using mutate_al
d1 %>%
  mutate_all(~(.)-mean(.)) %>%
  summarise_all(mean)
```

```
## # A tibble: 1 x 6
##    ST_CASE   COUNTY MINUTE      AGE LONGITUD TRAV_SP1
##      <dbl>    <dbl>  <dbl>    <dbl>    <dbl>    <dbl>
## 1 4.73e-11 1.32e-14     NA 1.58e-15       NA 1.17e-15
```

```r
#24. Standarizing a variable is to subtract it to its mean and then divide by its standard deviation. S

d1 %>%
  mutate_all(~((.)-mean(.))/sd(.)) %>%
  summarise_all(mean)
```

```
## # A tibble: 1 x 6
##     ST_CASE   COUNTY MINUTE      AGE LONGITUD TRAV_SP1
##       <dbl>    <dbl>  <dbl>    <dbl>    <dbl>    <dbl>
## 1 -9.97e-17 1.15e-16     NA 8.49e-17       NA 7.75e-17
```