Assignment3

Huiyi Huang

9/30/2019

#1 redo 13-26

```
setwd("C:/Users/student/Documents/Fall2019/")
getwd
```

```
## function ()
## .Internal(getwd())
## <bytecode: 0x0000000015138d78>
## <environment: namespace:base>
```

```
#install.packages("dplyr")

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
```

```
Titanic<-read.csv(file= 'titanic.csv')
```

#1 redo 13-26

```
Titanic %>%
  select(Age,Sex)%>%
  #13
  filter(Sex=="female",Age)%>%
  summarise(mean_female_age = mean(Age))
```

```
##   mean_female_age
## 1        27.91571
```

```
#14 Calculate the median fare of the passengers in Class 1
Titanic %>%
  select(Fare,Pclass)%>%
  filter(Pclass == 1) %>%
  summarise(median_fare_class1 = median(Fare))
```

```
##   median_fare_class1
## 1            60.2875
```

```
median(Titanic$Fare[Titanic$Pclass=='1'])
```

```
## [1] 60.2875
```

```
#15. Calculate the median fare of the female passengers that are not in Class 1
Titanic %>%
  select(Fare,Pclass,Sex)%>%
  filter(Sex=='female', Pclass!= 1)%>%
  summarise(median(Fare))
```

```
##   median(Fare)
## 1     14.45625
```

```
#16. Calculate the median age of survived passengers who are female and Class 1 or Class 2,

Titanic %>%
  select(Age,Survived,Sex,Pclass)%>%
  filter(Survived==1, Sex=="female",Pclass !=3,Age != 'NA' ) %>%
  summarise(median(Age))
```

```
##   median(Age)
## 1          31
```

```
#17. Calculate the mean fare of female teenagers survived passengers
Titanic %>%
  select(Age,Survived,Sex,Fare)%>%
  filter(Survived==1, Sex=="female",Age < 18)  %>%
  summarise(mean(Fare))
```

```
##   mean(Fare)
## 1   33.17226
```

```
#18. Calculate the mean fare of female teenagers survived passengers for each class
Titanic %>%
  group_by(Pclass) %>%
  summarise(m=mean(Fare))
```

```
## # A tibble: 3 x 2
##   Pclass     m
##    <int> <dbl>
## 1      1  84.2
## 2      2  20.7
## 3      3  13.7
```

```
#19. Calculate the ratio of Survived and not Survived for passengers who are who pays more than the
#average fare
Titanic %>%
  group_by(Survived)%>%
  summarize(mean(Fare))
```

```
## # A tibble: 2 x 2
##   Survived `mean(Fare)`
##      <int>        <dbl>
## 1        0         22.1
## 2        1         48.4
```

```
  48.39541/22.11789
```

```
## [1] 2.188066
```

```
#20. Add column that standardizes the fare (subtract the mean and divide by standard deviation) and nam
#it sfare
T =Titanic %>%
  mutate(S_fare= (Fare-mean(Fare))/sd(Fare))
```

```
#21. Add categorical variable named cfare that takes value cheap for passengers paying less the average
#fare and takes value expensive for passengers paying more than the average fare.
T =Titanic %>%
  mutate(cfare= ifelse(Fare>mean(Fare),'expensive','cheap'))
```

```
#22. Add categorical variable named cage that takes value 0 for age 0-10, 1 for age 10-20, 2 for age 20
#and so on
T =Titanic %>%
  mutate(cage = case_when(Age<10~'0',
                          (Age>=10)&(Age<20)~'1',
                          (Age>=20)&(Age<30)~'2',
                          (Age>=30)&(Age<40)~'3',
                          (Age>=40)&(Age<50)~'4',
                          (Age>=50)&(Age<60)~'5',
                          (Age>=60)&(Age<70)~'6',
                          (Age>=70)&(Age<80)~'7',
                          (Age>=80)~'8'))
```

```
#23. Show the frequency of Ports of Embarkation. It appears that there are two missing values in the
#Embarked variable. Assign the most frequent port to the missing ports. Hint: Use the levels
#function to modify the categories of categorical variables.
Titanic %>%
  count(Embarked)%>%
  mutate(Embarked=replace(Embarked,Embarked=='','S'))
```

```
## # A tibble: 4 x 2
##   Embarked     n
##   <fct>    <int>
## 1 S            2
## 2 C          168
## 3 Q           77
## 4 S          644
```

#2 assignment2 question 4

```
#install.packages('readxl') # install the library
library(readxl) # load the library
```

```
#2. Using Dplyr and in Assignment 2, redo 4 using sample_n function, redo 5 using glimpse, redo 11, 12
#and 13. For 11, 12 and 13, you may want to use the combo group_by and summarise
read_excel("C:/Users/student/Documents/Fall2019/c2015.xlsx")
```

```
## # A tibble: 80,587 x 28
##    STATE ST_CASE VEH_NO PER_NO COUNTY   DAY MONTH  HOUR MINUTE AGE   SEX  
##    <chr>   <dbl>  <dbl>  <dbl>  <dbl> <dbl> <chr> <dbl>  <dbl> <chr> <chr>
##  1 Alab~   10001      1      1    127     1 Janu~     2     40 68    Male 
##  2 Alab~   10002      1      1     83     1 Janu~    22     13 49    Male 
##  3 Alab~   10003      1      1     11     1 Janu~     1     25 31    Male 
##  4 Alab~   10003      1      2     11     1 Janu~     1     25 20    Fema~
##  5 Alab~   10004      1      1     45     4 Janu~     0     57 40    Male 
##  6 Alab~   10005      1      1     45     7 Janu~     7      9 24    Male 
##  7 Alab~   10005      2      1     45     7 Janu~     7      9 60    Male 
##  8 Alab~   10006      1      1    111     8 Janu~     9     59 64    Male 
##  9 Alab~   10006      1      2    111     8 Janu~     9     59 17    Male 
## 10 Alab~   10007      1      1     89     8 Janu~    18     33 80    Male 
## # ... with 80,577 more rows, and 17 more variables: PER_TYP <chr>,
## #   INJ_SEV <chr>, SEAT_POS <chr>, DRINKING <chr>, YEAR <dbl>,
## #   MAN_COLL <chr>, OWNER <chr>, MOD_YEAR <chr>, TRAV_SP <chr>,
## #   DEFORMED <chr>, DAY_WEEK <chr>, ROUTE <chr>, LATITUDE <dbl>,
## #   LONGITUD <dbl>, HARM_EV <chr>, LGT_COND <chr>, WEATHER <chr>
```

```
c2015=read_excel("C:/Users/student/Documents/Fall2019/c2015.xlsx")

set.seed(2019)
y<- sample_n(c2015,1000)
```

```
#5. Use summary function to have a quick look at the data. You will notice there is one variable is act
#5a constant. Remove that variable from the data.

glimpse(y)
```

```
## Observations: 1,000
## Variables: 28
## $ STATE    <chr> "New Jersey", "Arizona", "Tennessee", "Minnesota", "M...
## $ ST_CASE  <dbl> 340336, 40327, 470789, 270119, 290576, 62865, 330095,...
## $ VEH_NO   <dbl> 1, 1, 1, 2, 1, 1, 0, 0, 2, 5, 1, 2, 1, 0, 1, 1, 2, 1,...
## $ PER_NO   <dbl> 1, 1, 1, 4, 1, 1, 1, 1, 4, 1, 1, 1, 5, 1, 1, 2, 1, 1,...
## $ COUNTY   <dbl> 27, 13, 163, 59, 201, 19, 15, 127, 13, 115, 29, 141, ...
## $ DAY      <dbl> 19, 7, 2, 16, 2, 6, 3, 30, 17, 30, 19, 12, 9, 30, 9, ...
## $ MONTH    <chr> "September", "May", "December", "May", "October", "Ju...
## $ HOUR     <dbl> 3, 22, 8, 21, 15, 15, 14, 20, 7, 14, 14, 17, 18, 6, 4...
## $ MINUTE   <dbl> 17, 15, 26, 59, 38, 20, 32, 20, 41, 36, 15, 50, 55, 4...
## $ AGE      <chr> "Unknown", "47", "23", "15", "55", "56", "26", "63", ...
## $ SEX      <chr> "Unknown", "Female", "Male", "Female", "Male", "Male"...
## $ PER_TYP  <chr> "Driver of a Motor Vehicle In-Transport", "Driver of ...
## $ INJ_SEV  <chr> "Unknown", "No Apparent Injury (O)", "Unknown", "Susp...
## $ SEAT_POS <chr> "Front Seat, Left Side", "Front Seat, Left Side", "Fr...
## $ DRINKING <chr> "Not Reported", "No (Alcohol Not Involved)", "Unknown...
## $ YEAR     <dbl> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015,...
## $ MAN_COLL <chr> "Not a Collision with Motor Vehicle In-Transport", "N...
## $ OWNER    <chr> "Unknown", "Driver (in this crash) Not Registered Own...
## $ MOD_YEAR <chr> "Unknown", "2003", "1994", "2011", "2000", "2013", NA...
## $ TRAV_SP  <chr> "Unknown", "048 MPH", "Not Rep", "055 MPH", "055 MPH"...
## $ DEFORMED <chr> "Unknown", "Functional Damage", "Minor Damage", "Disa...
## $ DAY_WEEK <chr> "Saturday", "Thursday", "Wednesday", "Saturday", "Fri...
## $ ROUTE    <chr> "State Highway", "Local Street", "County Road", "Stat...
## $ LATITUDE <dbl> 40.95270, 33.41048, 36.57834, 45.42841, 37.13481, 36....
## $ LONGITUD <dbl> -74.59644, -112.06459, -82.27889, -93.36788, -89.5946...
## $ HARM_EV  <chr> "Pedestrian", "Pedestrian", "Pedalcyclist", "Motor Ve...
## $ LGT_COND <chr> "Dark - Not Lighted", "Dark - Lighted", "Dark - Not L...
## $ WEATHER  <chr> "Clear", "Clear", "Clear", "Rain", "Cloud", "Clear", ...
```

```
#11. Compare the average speed of those who had "No Apprent Injury" and the rest. What do you
#observe?
#12. Use the SEAT_POS variable to filter the data so that there is only drivers in the dataset. Compare
#average speed of man drivers and woman drivers. Comment on the results.
#13. Compare the average speed of drivers who drink and those who do not. Comment on the results.
#Hint: This calculation can be done manually or by using the aggregate function or by function in
#base R. For example:
library(stringr)
#11
y%>%
  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))) %>%
  group_by(INJ_SEV)%>%
  summarise(mean(TRAV_SP1))
```

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

```
## # A tibble: 7 x 2
##   INJ_SEV                     `mean(TRAV_SP1)`
##   <chr>                                  <dbl>
## 1 Fatal Injury (K)                        52.1
## 2 Injured, Severity Unknown               45.5
## 3 No Apparent Injury (O)                  48.7
## 4 Possible Injury (C)                     48.1
## 5 Suspected Minor Injury(B)               51.4
## 6 Suspected Serious Injury(A)             52.6
## 7 Unknown                                 46.8
```

```
#Obviously, no injury type of drivers who have slower speed comparing to other drivers who have injurie
```

```
#12
y%>%
  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))) %>%
  filter(SEAT_POS=="Front Seat, Left Side")%>%
  group_by(SEX) %>%
  summarise(mean(TRAV_SP1))
```

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

```
## # A tibble: 3 x 2
##   SEX     `mean(TRAV_SP1)`
##   <chr>              <dbl>
## 1 Female              49.0
## 2 Male                51.1
## 3 Unknown             45.5
```

```
#Male drivers tend to drive faster than female drivers
```

```
#13
y%>%
  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))) %>%
  group_by(DRINKING)%>%
  summarise(mean(TRAV_SP1))
```

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

```
## # A tibble: 4 x 2
##   DRINKING                  `mean(TRAV_SP1)`
##   <chr>                                <dbl>
## 1 No (Alcohol Not Involved)             48.8
## 2 Not Reported                          51.3
## 3 Unknown (Police Reported)             51.6
## 4 Yes (Alcohol Involved)                57.2
```

```
#Drinking alcohol drivers drive faster than not drinking drivers
```

```
#3. Calculate the travel speed (TRAV_SP variable) by day. Compare the travel speed of the first 5 days
#the last 5 days of months.
y%>%
  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))) %>%
  filter(DAY==1:5)%>%
  group_by(DAY)%>%
  summarise(mean = mean(TRAV_SP1))%>%
  summarise(fist_5_day_mean=mean(mean))
```

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

```
## # A tibble: 1 x 1
##   fist_5_day_mean
##             <dbl>
## 1            45.0
```

```
#continued question 3
y%>%
  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))) %>%
  filter(DAY==27:31)%>%
  group_by(DAY)%>%
  summarise(mean2 = mean(TRAV_SP1)) %>%
  summarise(last_5_day_mean=mean(mean2))
```

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

```
## # A tibble: 1 x 1
##   last_5_day_mean
##             <dbl>
## 1            49.8
```

```
# From calculated the mean of first 5 days and last 5 days mean speed, here comes conclusion: people wh
```

```
#4. Calculate the travel speed (TRAV_SP variable) by day of the week. Compare the travel speed of the
#weekdays and weekends.
y%>%
  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))) %>%
  group_by(DAY_WEEK=="Saturday" | DAY_WEEK=="Sunday")%>%
  summarise(mean(TRAV_SP1))
```

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

```
## # A tibble: 2 x 2
##   `DAY_WEEK == "Saturday" | DAY_WEEK == "Sunday"` `mean(TRAV_SP1)`
##   <lgl>                                                      <dbl>
## 1 FALSE                                                       50.1
## 2 TRUE                                                        52.0
```

```
#The comparasion calculated the mean of speed when day of week is weekend or not. The conclusion is tha
```

```
#5. Find the top 5 states with greatest travel speed.
y%>%
  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))) %>%
  group_by(STATE)%>%
  summarise(mean=mean(TRAV_SP1))%>%
  top_n(5,mean)
```

```
## Warning: NAs introduced by coercion
## # A tibble: 5 x 2
##   STATE         mean
##   <chr>        <dbl>
## 1 Colorado      56.9
## 2 Nevada        62.1
## 3 North Dakota  62.2
## 4 South Dakota  69.5
## 5 Wyoming       61.3
```

```
#6. Rank the travel speed by MONTH

y%>%
  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))) %>%
  group_by(MONTH)%>%
  summarise(mean=mean(TRAV_SP1))%>%
  arrange(desc(mean))
```

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

```
## # A tibble: 12 x 2
##    MONTH      mean
##    <chr>     <dbl>
##  1 December   52.8
##  2 April      52.4
##  3 September  52.3
##  4 June       51.5
##  5 November   51.4
##  6 October    51.4
##  7 August     50.1
##  8 May        49.9
##  9 February   49.8
## 10 July       49.1
## 11 March      48.9
## 12 January    48.7
```

```
#7. Find the average speed of teenagers in December.
y%>%
  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))) %>%
  filter(AGE<18 & MONTH=="December")%>%
  summarise(mean(TRAV_SP1))
```

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

```
## # A tibble: 1 x 1
##   `mean(TRAV_SP1)`
##              <dbl>
## 1             58.1
```

```
#8. Find the month that female drivers drive fastest on average.
y%>%
  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))) %>%
  filter(SEX=="Female")%>%
  group_by(MONTH)%>%
  summarise(mean=mean(TRAV_SP1))%>%
  top_n(1,mean)
```

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

```
## # A tibble: 1 x 2
##   MONTH     mean
##   <chr>    <dbl>
## 1 December  53.5
```

```
#9. Find the month that male driver drive slowest on average.
y%>%
  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))) %>%
  filter(SEX=="Male")%>%
  group_by(MONTH)%>%
  summarise(mean=mean(TRAV_SP1))%>%
  arrange(mean)
```

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

```
## # A tibble: 12 x 2
##    MONTH      mean
##    <chr>     <dbl>
##  1 February   48.3
##  2 May        48.3
##  3 July       49.0
##  4 March      49.0
##  5 January    49.1
##  6 June       51.9
##  7 August     52.0
##  8 September  52.1
##  9 December   52.4
## 10 November   52.5
## 11 October    52.7
## 12 April      53.6
```

```
#It's Febrary.
```

```
#10. Create a new column containing information about the season of the accidents. Compare the percenta
y1 = y%>%
  mutate(SEASON=case_when(
    MONTH == "March"|MONTH =="April"|MONTH =="May"~"Spring",
    MONTH == "June"|MONTH =="July"|MONTH =="August"~ "Summer",
    MONTH == "September"|MONTH =="October"|MONTH =="November"~"Autumn",
    MONTH == "December"|MONTH =="January"|MONTH =="February"~"Winter"))
```

```
#continue question10
y1 %>%
  group_by(SEASON) %>% summarize(prop=prop.table(table(INJ_SEV))[1])
```

```
## # A tibble: 4 x 2
##   SEASON  prop
##   <chr>  <dbl>
## 1 Autumn 0.440
## 2 Spring 0.418
## 3 Summer 0.459
## 4 Winter 0.409
```

```
#11. Compare the percentage of fatal injuries for different type of deformations (DEFORMED variable)
y%>%



  group_by(INJ_SEV,DEFORMED) %>%
  summarise(n =n()) %>%
  mutate(prop=n/sum(n))
```

```
## # A tibble: 33 x 4
## # Groups:   INJ_SEV [7]
##    INJ_SEV                   DEFORMED               n    prop
##    <chr>                     <chr>              <int>   <dbl>
##  1 Fatal Injury (K)          Disabling Damage     315 0.726
##  2 Fatal Injury (K)          Functional Damage      9 0.0207
##  3 Fatal Injury (K)          Minor Damage           7 0.0161
##  4 Fatal Injury (K)          No Damage              2 0.00461
##  5 Fatal Injury (K)          Not Reported           9 0.0207
##  6 Fatal Injury (K)          Unknown                7 0.0161
##  7 Fatal Injury (K)          <NA>                  85 0.196
##  8 Injured, Severity Unknown Disabling Damage       3 1
##  9 No Apparent Injury (O)    Disabling Damage      94 0.355
## 10 No Apparent Injury (O)    Functional Damage     63 0.238
## # ... with 23 more rows
```