Assignment 3

Kyle Walker

9/19/2019

1.Read the titanic data set as a tibble. Redo questions 13 to 23 in the Assignment 1 using dplyr.

Notice: you may want to use logical operators such as:

Operators Discription

!= not equal to

!x Not x

x | y x OR y

x & y x AND y

```
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
```

```
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()
```

```
library(readr)
titanic <- read_csv(file = "C:/Users/student/Documents/Senior Year/MATH 421/titanic.csv")
```

```
## Parsed with column specification:
## cols(
##   PassengerId = col_double(),
##   Survived = col_double(),
##   Pclass = col_double(),
##   Name = col_character(),
##   Sex = col_character(),
##   Age = col_double(),
##   SibSp = col_double(),
##   Parch = col_double(),
##   Ticket = col_character(),
##   Fare = col_double(),
##   Cabin = col_character(),
##   Embarked = col_character()
## )
```

13 Calculate the mean age of female passengers:

```
titanic %>%
  filter(Sex == "female") %>%
      summarize(mean(Age, na.rm = T))
```

```
## # A tibble: 1 x 1
##   `mean(Age, na.rm = T)`
##                    <dbl>
## 1                   27.9
```

14. Calculate the median fare of the passengers in Class 1

```
titanic %>%
  filter(Pclass==1) %>%
    summarize(median(Fare, na.rm=T))
```

```
## # A tibble: 1 x 1
##   `median(Fare, na.rm = T)`
##                       <dbl>
## 1                      60.3
```

15. Calculate the median fare of the female passengers that are not in Class 1

```
titanic %>%
  filter(Pclass!=1, Sex=='female') %>%
    summarize(median(Fare,na.rm =T))
```

```
## # A tibble: 1 x 1
##   `median(Fare, na.rm = T)`
##                       <dbl>
## 1                      14.5
```

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

```
titanic %>%
  filter(Pclass==1 | Pclass== 2, Sex=='female') %>%
    summarize(median(Fare, na.rm=T))
```

```
## # A tibble: 1 x 1
##   `median(Fare, na.rm = T)`
##                        <dbl>
## 1                       49.5
```

17. Calculate the mean fare of female teenagers survived passengers

```
titanic %>%
  filter(Sex=='female', Age >= 13, Age < 20, Survived == 1) %>%
    summarize(mean(Fare, na.rm=T))
```

```
## # A tibble: 1 x 1
##   `mean(Fare, na.rm = T)`
##                      <dbl>
## 1                     49.2
```

18. Calculate the mean fare of female teenagers survived passengers for each class

```
titanic %>%
  filter(Sex=='female', Survived==1) %>%
    group_by(Pclass) %>%
      summarize(mean(Fare, na.rm=T))
```

```
## # A tibble: 3 x 2
##   Pclass `mean(Fare, na.rm = T)`
##    <dbl>                   <dbl>
## 1      1                   106.
## 2      2                    22.3
## 3      3                    12.5
```

19. Calculate the ratio of Survived and not Survived for passengers who are who pays more than the average fare

```
titanic %>%
  filter(Fare>mean(Fare), Survived==0 | Survived == 1) %>%
    select(Survived) %>%
      summarize(sum(Survived) / length(Survived))
```

```
## # A tibble: 1 x 1
##   `sum(Survived)/length(Survived)`
##                               <dbl>
## 1                             0.597
```

20. Add column that standardizes the fare (subtract the mean and divide by standard deviation) and name it sfare

```
titanic %>%
  mutate(sfare = ((Fare - mean(Fare)) / sd(Fare) ))
```

```
## # A tibble: 891 x 13
##    PassengerId Survived Pclass Name  Sex     Age SibSp Parch Ticket  Fare
##          <dbl>    <dbl>  <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr>  <dbl>
##  1           1        0      3 Brau~ male     22     1     0 A/5 2~  7.25
##  2           2        1      1 Cumi~ fema~    38     1     0 PC 17~ 71.3
##  3           3        1      3 Heik~ fema~    26     0     0 STON/~  7.92
##  4           4        1      1 Futr~ fema~    35     1     0 113803 53.1
##  5           5        0      3 Alle~ male     35     0     0 373450  8.05
##  6           6        0      3 Mora~ male     NA     0     0 330877  8.46
##  7           7        0      1 McCa~ male     54     0     0 17463  51.9
##  8           8        0      3 Pals~ male      2     3     1 349909 21.1
##  9           9        1      3 John~ fema~    27     0     2 347742 11.1
## 10          10        1      2 Nass~ fema~    14     1     0 237736 30.1
## # ... with 881 more rows, and 3 more variables: Cabin <chr>,
## #   Embarked <chr>, sfare <dbl>
```

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.

```
titanic %>%
  mutate(cfare = cut(Fare, breaks= c(0, mean(Fare), Inf), labels = c("cheap", "expensive") ))
```

```
## # A tibble: 891 x 13
##    PassengerId Survived Pclass Name  Sex     Age SibSp Parch Ticket  Fare
##          <dbl>    <dbl>  <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr>  <dbl>
##  1           1        0      3 Brau~ male     22     1     0 A/5 2~  7.25
##  2           2        1      1 Cumi~ fema~    38     1     0 PC 17~ 71.3
##  3           3        1      3 Heik~ fema~    26     0     0 STON/~  7.92
##  4           4        1      1 Futr~ fema~    35     1     0 113803 53.1
##  5           5        0      3 Alle~ male     35     0     0 373450  8.05
##  6           6        0      3 Mora~ male     NA     0     0 330877  8.46
##  7           7        0      1 McCa~ male     54     0     0 17463  51.9
##  8           8        0      3 Pals~ male      2     3     1 349909 21.1
##  9           9        1      3 John~ fema~    27     0     2 347742 11.1
## 10          10        1      2 Nass~ fema~    14     1     0 237736 30.1
## # ... with 881 more rows, and 3 more variables: Cabin <chr>,
## #   Embarked <chr>, cfare <fct>
```

22. Add categorical variable named cage that takes value 0 for age 0-10, 1 for age 10-20, 2 for age 20-30, and so on

```
titanic %>%
  mutate(cage = cut(Age, breaks = c(0,10,20,30,40,50,60,70,80,90,Inf), labels = c(0,1,2,3,4,5,6,7,8,9)))
```

```
## # A tibble: 891 x 13
```

```
##    PassengerId Survived Pclass Name  Sex     Age SibSp Parch Ticket  Fare
##          <dbl>    <dbl>  <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr>  <dbl>
##  1           1        0      3 Brau~ male     22     1     0 A/5 2~  7.25
##  2           2        1      1 Cumi~ fema~    38     1     0 PC 17~ 71.3
##  3           3        1      3 Heik~ fema~    26     0     0 STON/~  7.92
##  4           4        1      1 Futr~ fema~    35     1     0 113803 53.1
##  5           5        0      3 Alle~ male     35     0     0 373450  8.05
##  6           6        0      3 Mora~ male     NA     0     0 330877  8.46
##  7           7        0      1 McCa~ male     54     0     0 17463  51.9
##  8           8        0      3 Pals~ male      2     3     1 349909 21.1
##  9           9        1      3 John~ fema~    27     0     2 347742 11.1
## 10          10        1      2 Nass~ fema~    14     1     0 237736 30.1
## # ... with 881 more rows, and 3 more variables: Cabin <chr>,
## #   Embarked <chr>, cage <fct>
```

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.

```r
titanic %>%
  group_by(Embarked) %>%
    count(Embarked)
```

```
## # A tibble: 4 x 2
## # Groups:   Embarked [4]
##   Embarked     n
##   <chr>    <int>
## 1 C          168
## 2 Q           77
## 3 S          644
## 4 <NA>         2
```

```r
titanic %>%
    mutate(Embarked = replace_na(Embarked, "S")) %>%
      count(Embarked)
```

```
## # A tibble: 3 x 2
##   Embarked     n
##   <chr>    <int>
## 1 C          168
## 2 Q           77
## 3 S          646
```

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

```r
library(readxl)
c2015 <- read_excel("C:/Users/student/Documents/Senior Year/MATH 421/Assignment 2/c2015.xlsx")
```

4. Use dim function to check the dimension of the data. Since this data is quite big, a common practice is to randomly subset the data to analyze. Use sample function to create a new dataset that has a random 1000 observations from the original data. Use set.seed(2019) before using the sample function to set the seed for the randomness so that everyone in class is working with the same random subset of the data.

```
set.seed(2019)
c2015 %>%
  sample_n(1000)
```

```
## # A tibble: 1,000 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 New ~  340336      1      1     27    19 Sept~     3     17 Unkn~ Unkn~
##  2 Ariz~   40327      1      1     13     7 May      22     15 47    Fema~
##  3 Tenn~  470789      1      1    163     2 Dece~     8     26 23    Male
##  4 Minn~  270119      2      4     59    16 May      21     59 15    Fema~
##  5 Miss~  290576      1      1    201     2 Octo~    15     38 55    Male
##  6 Cali~   62865      1      1     19     6 June     15     20 56    Male
##  7 New ~  330095      0      1     15     3 Dece~    14     32 26    Male
##  8 Iowa   190173      0      1    127    30 Augu~    20     20 63    Male
##  9 Cali~   62263      2      4     13    17 Dece~     7     41 6     Male
## 10 Alab~   10286      5      1    115    30 May      14     36 32    Male
## # ... with 990 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>
```

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

```
c2015 %>%
glimpse()
```

```
## Observations: 80,587
## Variables: 28
## $ STATE    <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama"...
## $ 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...
## $ MONTH    <chr> "January", "January", "January", "January", "January"...
## $ 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      <chr> "68", "49", "31", "20", "40", "24", "60", "64", "17",...
## $ SEX      <chr> "Male", "Male", "Male", "Female", "Male", "Male", "Ma...
## $ PER_TYP  <chr> "Driver of a Motor Vehicle In-Transport", "Driver of ...
## $ INJ_SEV  <chr> "Fatal Injury (K)", "Fatal Injury (K)", "Fatal Injury...
## $ SEAT_POS <chr> "Front Seat, Left Side", "Front Seat, Left Side", "Fr...
## $ DRINKING <chr> "Unknown (Police Reported)", "No (Alcohol Not Involve...
```

```
## $ 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> "Driver (in this crash) was  Registered Owner", "Driv...
## $ MOD_YEAR <chr> "2003", "2006", "2008", "2008", "2005", "2006", "2015...
## $ TRAV_SP  <chr> "055 MPH", "070 MPH", "080 MPH", "080 MPH", "075 MPH"...
## $ DEFORMED <chr> "Disabling Damage", "Disabling Damage", "Disabling Da...
## $ DAY_WEEK <chr> "Thursday", "Thursday", "Thursday", "Thursday", "Sund...
## $ ROUTE    <chr> "State Highway", "Interstate", "U.S. Highway", "U.S. ...
## $ 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...
## $ HARM_EV  <chr> "Embankment", "Ditch", "Tree (Standing Only)", "Tree ...
## $ LGT_COND <chr> "Dark - Not Lighted", "Dark - Not Lighted", "Dark - N...
## $ WEATHER  <chr> "Clear", "Cloud", "Clear", "Clear", "Cloud", "Clear",...
```

```r
select(c2015, -YEAR)
```

```
## # A tibble: 80,587 x 27
##    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 16 more variables: PER_TYP <chr>,
## #   INJ_SEV <chr>, SEAT_POS <chr>, DRINKING <chr>, 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>
```

11. Compare the average speed of those who had "No Apprent Injury" and the rest. What do you observe?

```r
library(stringr)
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
```

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

```r
c2015 %>%
  group_by(INJ_SEV) %>%
    summarize(mean(TRAV_SP, na.rm=T))
```

```
## # A tibble: 8 x 2
##   INJ_SEV                        `mean(TRAV_SP, na.rm = T)`
##   <chr>                                               <dbl>
## 1 Died Prior to Crash*                                 68.5
## 2 Fatal Injury (K)                                     54.5
## 3 Injured, Severity Unknown                            41.2
## 4 No Apparent Injury (O)                               41.6
## 5 Possible Injury (C)                                  48.0
## 6 Suspected Minor Injury(B)                            51.7
## 7 Suspected Serious Injury(A)                          54.6
## 8 Unknown                                              47.9
```

12. Use the SEAT_POS variable to filter the data so that there is only drivers in the dataset. Compare the average speed of man drivers and woman drivers. Comment on the results.

```
c2015 %>%
  filter(SEAT_POS == "Front Seat, Left Side") %>%
    group_by(SEX) %>%
      summarize(mean(TRAV_SP, na.rm = T))
```

```
## # A tibble: 4 x 2
##   SEX     `mean(TRAV_SP, na.rm = T)`
##   <chr>                        <dbl>
## 1 Female                        46.2
## 2 Male                          50.5
## 3 Not Rep                       50.3
## 4 Unknown                       46.5
```

```
#The males drive faster on average than females
```

13. Compare the average speed of drivers who drink and those who do not. Comment on the results.

```
c2015 %>%
  filter(DRINKING == "Yes (Alcohol Involved)" | DRINKING == "No (Alcohol Not Involved)") %>%
    group_by(DRINKING) %>%
      summarize(mean(TRAV_SP))
```

```
## # A tibble: 2 x 2
##   DRINKING                  `mean(TRAV_SP)`
##   <chr>                               <dbl>
## 1 No (Alcohol Not Involved)            46.5
## 2 Yes (Alcohol Involved)               60.1
```

```
#Drivers who are drinking drive faster
```

3. Calculate the travel speed (TRAV_SP variable) by day. Compare the travel speed of the first 5 days and the last 5 days of months.

```r
c2015 %>%
  group_by(DAY) %>%
    summarize(mean(TRAV_SP))
```

```
## # A tibble: 31 x 2
##      DAY `mean(TRAV_SP)`
##    <dbl>           <dbl>
##  1     1            49.1
##  2     2            51.4
##  3     3            50.0
##  4     4            49.3
##  5     5            50.5
##  6     6            49.9
##  7     7            49.0
##  8     8            50.6
##  9     9            46.6
## 10    10            50.5
## # ... with 21 more rows
```

```r
c2015 %>%
  filter(DAY <= 5 | DAY >= 26) %>%
    group_by(DAY <= 5, DAY >= 26) %>%
      summarize(mean(TRAV_SP, na.rm=T))
```

```
## # A tibble: 2 x 3
## # Groups:   DAY <= 5 [2]
##   `DAY <= 5` `DAY >= 26` `mean(TRAV_SP, na.rm = T)`
##   <lgl>      <lgl>                            <dbl>
## 1 FALSE      TRUE                              50.7
## 2 TRUE       FALSE                             50.0
```

```r
#There is no significant difference
```

4. Calculate the travel speed (TRAV_SP variable) by day of the week. Compare the travel speed of the weekdays and weekends.

```r
c2015 %>%
  group_by(DAY_WEEK) %>%
    summarize(mean(TRAV_SP,na.rm=T))
```

```
## # A tibble: 7 x 2
##   DAY_WEEK  `mean(TRAV_SP, na.rm = T)`
##   <chr>                          <dbl>
## 1 Friday                          49.4
## 2 Monday                          48.1
## 3 Saturday                        51.1
## 4 Sunday                          53.5
## 5 Thursday                        48.8
## 6 Tuesday                         48.2
## 7 Wednesday                       48.2
```

```r
#The weekends have a higher average travel speed than the weekdays
```

5. Find the top 5 states with greatest travel speed.

```r
c2015 %>%
  group_by(STATE) %>%
    summarize(Max_SPD = max(TRAV_SP)) %>%
      top_n(5, Max_SPD)
```

```
## # A tibble: 5 x 2
##   STATE          Max_SPD
##   <chr>            <dbl>
## 1 Alabama            140
## 2 Arizona            150
## 3 North Carolina     140
## 4 Texas              140
## 5 Virginia           150
```

6. Rank the travel speed by MONTH.

```r
c2015 %>%
  group_by(MONTH) %>%
    summarize(avg_SPD = mean(TRAV_SP)) %>%
      arrange(desc(avg_SPD))
```

```
## # A tibble: 12 x 2
##    MONTH     avg_SPD
##    <chr>       <dbl>
##  1 December     50.7
##  2 July         50.5
##  3 August       50.1
##  4 November     50.1
##  5 May          50.1
##  6 April        50.0
##  7 October      50.0
##  8 June         49.7
##  9 January      49.6
## 10 March        49.5
## 11 September    49.2
## 12 February     48.7
```

7. Find the average speed of teenagers in December.

```r
c2015 %>%
  filter(AGE <= 20, AGE >= 13, MONTH == "December") %>%
    summarize(mean(TRAV_SP, na.rm= T))
```

```
## # A tibble: 1 x 1
##   `mean(TRAV_SP, na.rm = T)`
##                        <dbl>
## 1                       54.9
```

8. Find the month that female drivers drive fastest on average.

```
c2015 %>%
  filter(SEX=="Female") %>%
    group_by(MONTH) %>%
      summarize(avg_SPD = mean(TRAV_SP)) %>%
        top_n(1, avg_SPD)
```

```
## # A tibble: 1 x 2
##   MONTH avg_SPD
##   <chr>   <dbl>
## 1 July     49.1
```

9. Find the month that male driver drive slowest on average.

```
c2015 %>%
  filter(SEX=="Male") %>%
    group_by(MONTH) %>%
      summarize(avg_SPD = mean(TRAV_SP)) %>%
        top_n(-1, avg_SPD)
```

```
## # A tibble: 1 x 2
##   MONTH    avg_SPD
##   <chr>      <dbl>
## 1 February    49.5
```

10. Create a new column containing information about the season of the accidents. Compare the percentage of Fatal Injury by seasons.

```
c2015 %>%
  mutate(Season = recode(MONTH,
                          "December" = "Winter",
                          "January" = "Winter",
                          "February" = "Winter",
                          "March"= "Spring",
                          "April"= "Spring",
                          "May" = "Spring",
                          "June"= "Summer",
                          "July"= "Summer",
                          "August" = "Summer",
                          "September"= "Fall",
                          "October"= "Fall",
                          "November" = "Fall")
        ) %>%
      filter(INJ_SEV == "Fatal Injury (K)") %>%
        group_by(Season) %>%
          summarize(Fatalities = n()) %>%
              mutate(Total = Fatalities / sum(Fatalities))
```

```
## # A tibble: 4 x 3
##   Season Fatalities Total
```

```
##   <chr>       <int> <dbl>
## 1 Fall         2519 0.256
## 2 Spring       2519 0.256
## 3 Summer       2756 0.280
## 4 Winter       2053 0.208
```

```
#Summer has the highest frequency of fatalities
```

11. Compare the percentage of fatal injuries for different type of deformations (DEFORMED variable)

```
c2015 %>%
  group_by(DEFORMED) %>%
    summarize(Fatalities = n()) %>%
      mutate(Total = Fatalities / sum(Fatalities))
```

```
## # A tibble: 6 x 3
##   DEFORMED          Fatalities   Total
##   <chr>                  <int>   <dbl>
## 1 Disabling Damage       20595 0.791
## 2 Functional Damage       2837 0.109
## 3 Minor Damage            1675 0.0643
## 4 No Damage                442 0.0170
## 5 Not Reported             344 0.0132
## 6 Unknown                  145 0.00557
```

```
#The majority of Deformed damage is Disabling Damage
```