

# assignment\_04-02\_MunjewarSheetal.R

sheetal

2023-01-08

```
# Assignment: ASSIGNMENT 4.2  
# Name: Munjewar, Sheetal  
# Date: 2023-01-08
```

```
# Check your current working directory using `getwd()`  
getwd()
```

```
## [1] "E:/Data_Science_DSC510/DSC520-Statistics/dsc520/assignments/assignment04"
```

```
# List the contents of the working directory with the `dir()` function  
dir()
```

```
## [1] "assignment_04-01_MunjewarSheetal.pdf"  
## [2] "assignment_04-01_MunjewarSheetal.R"  
## [3] "assignment_04-02_MunjewarSheetal.pdf"  
## [4] "assignment_04-02_MunjewarSheetal.R"  
## [5] "assignment_04-02_MunjewarSheetal.spin.R"  
## [6] "assignment_04-02_MunjewarSheetal.spin.Rmd"  
## [7] "assignment_04_LastnameFirstname.R"  
## [8] "assignment_04_LastnameFirstname.Rmd"  
## [9] "assignment_04_MunjewarSheetal.R"  
## [10] "bibliography.bib"  
## [11] "data-transformation.pdf"
```

```
# If the current directory does not contain the `data` directory, set the  
# working directory to project root folder (the folder should contain the `data` directory  
# Use `setwd()` if needed  
# setwd("E:\\Data_Science_DSC510\\DSC520-Statistics\\dsc520")  
setwd("E:\\Data_Science_DSC510\\DSC520-Statistics\\dsc520")  
score_df <- read.csv("data/scores.csv")  
housing_df <- read.csv("data/week-7-housing.csv")  
  
#score_df <- read.csv("data/scores.csv")  
#summary(housing_df)  
#head(housing_df)  
#str(housing_df)
```

```

# The R base manual tells you that it's called as follows: apply(X, MARGIN, FUN, ...)
#
# where: Reference help
#
# -> X is an array or a matrix if the dimension of the array is 2;
# -> MARGIN is a variable defining how the function is applied: when MARGIN=1, it applies over rows, wh
# -> FUN, which is the function that you want to apply to the data. It can be any R function, including

##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

```

```

library(ggplot2)

#score_df <- read.csv("data/scores.csv")
#head(score_df)
##?apply()

```

```

# score_df %>% distinct(Section)

```

```

# score_df %>% distinct()

```

```

#head(score_df, n=5)
#head(score_df, n=15)
#head(score_df$Section, n=15)
#head(score_df$Count, n=1)

#tail(score_df, n=5)
#tail(score_df, n=15)
#tail(score_df$Section, n=15)
#tail(score_df$Count, n=1)

```

```

apply(score_df,MARGIN=2,FUN=min)

```

```

##      Count      Score  Section
##      "10"      "200" "Regular"

```

```

apply(score_df,MARGIN=2,FUN=max)

```

```

##      Count      Score  Section
##      "30"      "395" "Sports"

```

```
apply(housing_df,MARGIN=2,FUN=min)
```

```
##           Sale.Date           Sale.Price           sale_reason
##           "1/1/2009"           " 698"           " 0"
##           sale_instrument       sale_warning       sitetype
##           " 0"           ""           "A1"
##           addr_full           zip5           ctyname
##           "10002 242ND WAY NE"           "98052"           ""
##           postalctyn           lon           lat
##           "REDMOND"           "-121.9499"           "47.45635"
##           building_grade square_feet_total_living       bedrooms
##           " 2"           " 240"           " 0"
##           bath_full_count       bath_half_count       bath_3qtr_count
##           " 0"           "0"           "0"
##           year_built           year_renovated       current_zoning
##           "1900"           " 0"           "A10"
##           sq_ft_lot           prop_type       present_use
##           " 785"           "R"           " 0"
```

```
apply(housing_df,MARGIN=2,FUN=max)
```

```
##           Sale.Date           Sale.Price           sale_reason
##           "9/9/2016"           "4400000"           "19"
##           sale_instrument       sale_warning       sitetype
##           "27"           "8 46"           "R4"
##           addr_full           zip5           ctyname
##           "9985 185TH CT NE"           "98074"           "SAMMAMISH"
##           postalctyn           lon           lat
##           "REDMOND"           "-122.1643"           "47.73255"
##           building_grade square_feet_total_living       bedrooms
##           "13"           "13540"           "11"
##           bath_full_count       bath_half_count       bath_3qtr_count
##           "23"           "8"           "8"
##           year_built           year_renovated       current_zoning
##           "2016"           "2016"           "URPS0"
##           sq_ft_lot           prop_type       present_use
##           "1631322"           "R"           "300"
```

```
#apply(housing_df,MARGIN=2,FUN=which.max)
```

```
#apply(housing_df,MARGIN=2,FUN=max)
```

```
#apply(housing_df,MARGIN=2,FUN=min)
```

```
#apply(housing_df,MARGIN=2,FUN=summary)
```

```
test <- list(head(housing_df$Sale.Price,n=10))
str(test)
```

```
## List of 1
```

```
## $ : int [1:10] 698000 649990 572500 420000 369900 184667 1050000 875000 660000 650000
```

```
#lapply(test,FUN=mean)
```

```
#lapply(housing_df,FUN=mean)  
lapply(test,FUN=max)
```

```
## [[1]]  
## [1] 1050000
```

```
lapply(test,FUN=summary)
```

```
## [[1]]  
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
## 184667  458125  649995  613006  688500 1050000
```

```
# head(housing_df$Sale.Price,n=100)
```

```
my.df = data.frame(nums=seq(0.1,0.6,by=0.1), chars=letters[1:6],bools=sample(c(TRUE,FALSE), 6, replace=TRUE))  
my.df
```

```
##      nums chars bools  
## 1  0.1      a  TRUE  
## 2  0.2      b FALSE  
## 3  0.3      c  TRUE  
## 4  0.4      d  TRUE  
## 5  0.5      e FALSE  
## 6  0.6      f  TRUE
```

```
my.list = list(nums=seq(0.1,0.6,by=0.1), chars=letters[1:12],bools=sample(c(TRUE,FALSE), 6, replace=TRUE))  
my.list
```

```
## $nums  
## [1] 0.1 0.2 0.3 0.4 0.5 0.6  
##  
## $chars  
## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l"  
##  
## $bools  
## [1] FALSE TRUE TRUE TRUE TRUE TRUE
```

```
str(my.list)
```

```
## List of 3  
## $ nums : num [1:6] 0.1 0.2 0.3 0.4 0.5 0.6  
## $ chars: chr [1:12] "a" "b" "c" "d" ...  
## $ bools: logi [1:6] FALSE TRUE TRUE TRUE TRUE TRUE
```

```
# aggregate sum of marks with subjects  
list(score_df$Count)
```

```
## [[1]]
## [1] 10 10 20 10 10 10 10 30 10 10 20 10 10 10 20 10 10 20 10 10 20 10 30 10
## [26] 20 10 30 20 10 20 20 10 10 20 10 20 10
```

```
aggregate(score_df$Score,list(score_df$Section),FUN=sum)
```

```
## Group.1 x
## 1 Regular 6225
## 2 Sports 5840
```

```
#aggregate(housing_df$zip5,list(housing_df$Sale.Price),FUN=sum)
```

```
str(housing_df)
```

```
## 'data.frame': 12865 obs. of 24 variables:
## $ Sale.Date : chr "1/3/2006" "1/3/2006" "1/3/2006" "1/3/2006" ...
## $ Sale.Price : int 698000 649990 572500 420000 369900 184667 1050000 875000 660000 650000 ...
## $ sale_reason : int 1 1 1 1 1 1 1 1 1 1 ...
## $ sale_instrument : int 3 3 3 3 3 15 3 3 3 3 ...
## $ sale_warning : chr "" "" "" "" ...
## $ sitetype : chr "R1" "R1" "R1" "R1" ...
## $ addr_full : chr "17021 NE 113TH CT" "11927 178TH PL NE" "13315 174TH AVE NE" "3303 ...
## $ zip5 : int 98052 98052 98052 98052 98052 98053 98053 98053 98053 98052 ...
## $ ctynome : chr "REDMOND" "REDMOND" "" "REDMOND" ...
## $ postalctyn : chr "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
## $ lon : num -122 -122 -122 -122 -122 ...
## $ lat : num 47.7 47.7 47.7 47.6 47.7 ...
## $ building_grade : int 9 9 8 8 7 7 10 10 9 8 ...
## $ square_feet_total_living : int 2810 2880 2770 1620 1440 4160 3960 3720 4160 2760 ...
## $ bedrooms : int 4 4 4 3 3 4 5 4 4 4 ...
## $ bath_full_count : int 2 2 1 1 1 2 3 2 2 1 ...
## $ bath_half_count : int 1 0 1 0 0 1 0 1 1 0 ...
## $ bath_3qtr_count : int 0 1 1 1 1 1 1 0 1 1 ...
## $ year_built : int 2003 2006 1987 1968 1980 2005 1993 1988 1978 1976 ...
## $ year_renovated : int 0 0 0 0 0 0 0 0 0 0 ...
## $ current_zoning : chr "R4" "R4" "R6" "R4" ...
## $ sq_ft_lot : int 6635 5570 8444 9600 7526 7280 97574 30649 42688 94889 ...
## $ prop_type : chr "R" "R" "R" "R" ...
## $ present_use : int 2 2 2 2 2 2 2 2 2 2 ...
```

```
aggregate(housing_df$Sale.Price,list(housing_df$zip5),FUN=max)
```

```
## Group.1 x
## 1 98052 4400000
## 2 98053 3850000
## 3 98059 645000
## 4 98074 2160200
```

```
aggregate(housing_df$Sale.Price,list(housing_df$zip5),FUN=min)
```

```
##      Group.1      x
## 1    98052    2031
## 2    98053     698
## 3    98059  645000
## 4    98074  434000
```

```
#aggregate(housing_df$Sale.Price, list(housing_df$zip5), FUN=sum)
```

```
hdf_2006 <- housing_df[housing_df["Sale.Price"] >= 500008, 1:3]
```

```
dates <- as.POSIXct(housing_df$Sale.Date, format = "%m/%d/%Y")
years <- format(dates, format="%Y")
str(dates)
```

```
## POSIXct[1:12865], format: "2006-01-03" "2006-01-03" "2006-01-03" "2006-01-03" "2006-01-03" ...
```

```
summary(dates)
```

```
##              Min.              1st Qu.              Median
## "2006-01-03 00:00:00.000" "2008-07-07 00:00:00.000" "2011-11-17 00:00:00.000"
##              Mean              3rd Qu.              Max.
## "2011-07-28 15:24:21.266" "2014-06-05 00:00:00.000" "2016-12-16 00:00:00.000"
```

```
class(dates)
```

```
## [1] "POSIXct" "POSIXt"
```

```
str(housing_df)
```

```
## 'data.frame':    12865 obs. of  24 variables:
## $ Sale.Date      : chr  "1/3/2006" "1/3/2006" "1/3/2006" "1/3/2006" ...
## $ Sale.Price     : int   698000 649990 572500 420000 369900 184667 1050000 875000 660000 650000 ...
## $ sale_reason    : int    1 1 1 1 1 1 1 1 1 1 ...
## $ sale_instrument : int    3 3 3 3 3 15 3 3 3 3 ...
## $ sale_warning   : chr    "" "" "" "" ...
## $ sitetype       : chr    "R1" "R1" "R1" "R1" ...
## $ addr_full      : chr    "17021 NE 113TH CT" "11927 178TH PL NE" "13315 174TH AVE NE" "3303 ...
## $ zip5           : int   98052 98052 98052 98052 98052 98053 98053 98053 98052 ...
## $ ctynome        : chr    "REDMOND" "REDMOND" "" "REDMOND" ...
## $ postalctyn     : chr    "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
## $ lon            : num   -122 -122 -122 -122 -122 ...
## $ lat            : num    47.7 47.7 47.7 47.6 47.7 ...
## $ building_grade : int    9 9 8 8 7 7 10 10 9 8 ...
## $ square_feet_total_living: int  2810 2880 2770 1620 1440 4160 3960 3720 4160 2760 ...
## $ bedrooms       : int    4 4 4 3 3 4 5 4 4 4 ...
## $ bath_full_count : int    2 2 1 1 1 2 3 2 2 1 ...
## $ bath_half_count : int    1 0 1 0 0 1 0 1 1 0 ...
## $ bath_3qtr_count : int    0 1 1 1 1 1 1 0 1 1 ...
## $ year_built      : int   2003 2006 1987 1968 1980 2005 1993 1988 1978 1976 ...
## $ year_renovated   : int    0 0 0 0 0 0 0 0 0 0 ...
## $ current_zoning  : chr    "R4" "R4" "R6" "R4" ...
```

```
## $ sq_ft_lot           : int  6635 5570 8444 9600 7526 7280 97574 30649 42688 94889 ...
## $ prop_type           : chr   "R" "R" "R" "R" ...
## $ present_use         : int   2 2 2 2 2 2 2 2 2 2 ...
```

```
head(years)
```

```
## [1] "2006" "2006" "2006" "2006" "2006" "2006"
```

```
dates <- NULL
years <- NULL
```

```
housing_df %>% select(Sale.Price,Sale.Date) %>% filter(Sale.Price > 4000000)
```

```
##      Sale.Price  Sale.Date
## 1      4400000    3/2/2010
## 2      4400000    3/2/2010
## 3      4380542  11/17/2011
## 4      4380542  11/17/2011
## 5      4380542  11/17/2011
## 6      4380542  11/17/2011
## 7      4380542  11/17/2011
## 8      4380542  11/17/2011
## 9      4380542  11/17/2011
## 10     4380542  11/17/2011
## 11     4380542  11/17/2011
## 12     4380542  11/17/2011
## 13     4380542  11/17/2011
## 14     4380542  11/17/2011
## 15     4380542  11/17/2011
## 16     4380542  11/17/2011
## 17     4140203  11/17/2011
## 18     4140203  11/17/2011
## 19     4140203  11/17/2011
## 20     4140203  11/17/2011
## 21     4140203  11/17/2011
## 22     4140203  11/17/2011
## 23     4140203  11/17/2011
## 24     4140203  11/17/2011
## 25     4140203  11/17/2011
## 26     4140203  11/17/2011
## 27     4140203  11/17/2011
## 28     4140203  11/17/2011
## 29     4140203  11/17/2011
## 30     4140203  11/17/2011
## 31     4140203  11/17/2011
## 32      4311000   4/27/2016
```

```
#housing_df <- read.csv("data/week-7-housing.csv")
#str(housing_df)
```

```
#housing_df <- read.csv("data/week-7-housing.csv")
```

```
housing_df %>%
  filter(Sale.Price > 1000000) %>%
  group_by(Sale.Date) %>%
  summarize(average_revenue=mean(Sale.Price),sdev_revenue=sd(Sale.Price))
```

```
## # A tibble: 617 x 3
##   Sale.Date average_revenue sdev_revenue
##   <chr>         <dbl>         <dbl>
## 1 1/11/2008      1050000          NA
## 2 1/12/2006      1392000          NA
## 3 1/12/2015      1049990          NA
## 4 1/13/2007      1600000          NA
## 5 1/14/2016      1250000          NA
## 6 1/17/2014      1280000      42426.
## 7 1/18/2011      1870000          NA
## 8 1/19/2016      1840000          NA
## 9 1/23/2006      1445000          NA
## 10 1/23/2015      1041990          NA
## # ... with 607 more rows
```

```
#housing_df <- read.csv("data/week-7-housing.csv")
```

```
#housing_df %>%
# mutate(year=format(as.Date(Sale.Date, format="%m/%d/%Y"), "%Y"))
#head(housing_df)
```

```
housing_df %>%
  filter(Sale.Price > 1000000) %>%
  mutate(year=format(as.Date(Sale.Date, format="%m/%d/%Y"), "%Y")) %>%
  group_by(year) %>%
  summarize(average_sale_price=mean(Sale.Price),sdev_sale_price=sd(Sale.Price)) %>%
  arrange(desc(year))
```

```
## # A tibble: 11 x 3
##   year average_sale_price sdev_sale_price
##   <chr>         <dbl>         <dbl>
## 1 2016      1376730.      530913.
## 2 2015      1244531.      259357.
## 3 2014      1254733.      263183.
## 4 2013      1353820.      443343.
## 5 2012      2236266.      905252.
## 6 2011      2841839.     1361869.
## 7 2010      1462957.      645088.
## 8 2009      1369636.      292656.
## 9 2008      2574278.      866200.
## 10 2007      1624951.      588392.
## 11 2006      1383097.      369128.
```



```
#housing_df %>%
# mutate(Sale.month=format(as.Date(Sale.Date, format="%m/%d/%Y"), "%m"))

housing_df %>%
  filter(Sale.Price > 1000000) %>%
  mutate(monthyear = format(as.Date(Sale.Date, format="%m/%d/%Y"), "%m/%Y")) %>%
  group_by(monthyear) %>%
  summarize(average_sale_price=mean(Sale.Price), sdev_sale_price=sd(Sale.Price)) %>%
  arrange(monthyear)
```

```
## # A tibble: 124 x 3
##   monthyear average_sale_price sdev_sale_price
##   <chr>          <dbl>          <dbl>
## 1 01/2006          1235162.          212808.
## 2 01/2007          1387500          300520.
## 3 01/2008          1050000             NA
## 4 01/2009          1400000             NA
## 5 01/2011          1483333.          335012.
## 6 01/2012          1462500          441942.
## 7 01/2014          1240000          75498.
## 8 01/2015          1045990          5657.
## 9 01/2016          1415398          351044.
## 10 02/2006          1392522.          307312.
## # ... with 114 more rows
```

```
# housing_df$year <- NULL
```

```
# split sales date column, derive month on it and attach back to the frame.
```

```
housing_df$year <- format(as.Date(housing_df$Sale.Date, format="%m/%d/%Y"), "%Y")
tail(housing_df)
```

```
##      Sale.Date Sale.Price sale_reason sale_instrument sale_warning sitetype
## 12860 12/15/2016   824000          1              3              R1
## 12861 12/15/2016   798930          1              3              R1
## 12862 12/15/2016   750000          1              3              R1
## 12863 12/15/2016   629000          1              3              R1
## 12864 12/16/2016   835000          1              3              R1
## 12865 12/16/2016   455500          1              3              R1
##      addr_full zip5 ctyname postalctyn lon lat
## 12860 11314 177TH PL NE 98052 REDMOND REDMOND -122.1034 47.70083
## 12861 22506 NE 102ND PL 98053 REDMOND REDMOND -122.0406 47.69066
## 12862 13315 175TH AVE NE 98052 REDMOND REDMOND -122.1069 47.71926
## 12863 17716 NE 29TH ST 98052 REDMOND REDMOND -122.1026 47.63646
## 12864 9917 182ND CT NE 98052 REDMOND REDMOND -122.1003 47.68957
## 12865 8826 166TH AVE NE 98052 REDMOND REDMOND -122.1188 47.68141
##      building_grade square_feet_total_living bedrooms bath_full_count
## 12860              8              1980          3              2
## 12861              7              2920          3              2
## 12862              8              2320          4              1
## 12863              8              2000          4              1
## 12864              9              2460          4              2
```

```
## 12865          7          1150          3          1
##      bath_half_count bath_3qtr_count year_built year_renovated current_zoning
## 12860          1          0      2013          0          R4
## 12861          1          0      2001          0          URPS0
## 12862          1          1      1980          0          R6
## 12863          0          1      1967          0          R4
## 12864          1          0      1990          0          R4
## 12865          0          0      1961          0          R5
##      sq_ft_lot prop_type present_use year
## 12860      4500          R          2 2016
## 12861      5487          R          2 2016
## 12862     35048          R          2 2016
## 12863      9600          R          2 2016
## 12864      7615          R          2 2016
## 12865      8640          R          2 2016
```

```
# More examples.
housing_df$year <- format(mean(housing_df$Sale.Price))
housing_df$Zip <- housing_df$zip5 == "98052"
head(housing_df$Zip)
```

```
## [1] TRUE TRUE TRUE TRUE TRUE FALSE
```

```
housing_df$million_above <- housing_df$Sale.Price >= 1000000
head(housing_df$million_above)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE
```

```
# split sales date column, derive month on it and attach back to the frame.
housing_df$year <- format(as.Date(housing_df$Sale.Date, format="%m/%d/%Y"), "%Y")
tail(housing_df)
```

```
##      Sale.Date Sale.Price sale_reason sale_instrument sale_warning sitetype
## 12860 12/15/2016    824000          1          3          R1
## 12861 12/15/2016    798930          1          3          R1
## 12862 12/15/2016    750000          1          3          R1
## 12863 12/15/2016    629000          1          3          R1
## 12864 12/16/2016    835000          1          3          R1
## 12865 12/16/2016    455500          1          3          R1
##      addr_full zip5 ctyname postalctyn lon lat
## 12860 11314 177TH PL NE 98052 REDMOND REDMOND -122.1034 47.70083
## 12861 22506 NE 102ND PL 98053 REDMOND REDMOND -122.0406 47.69066
## 12862 13315 175TH AVE NE 98052 REDMOND REDMOND -122.1069 47.71926
## 12863 17716 NE 29TH ST 98052 REDMOND REDMOND -122.1026 47.63646
## 12864 9917 182ND CT NE 98052 REDMOND REDMOND -122.1003 47.68957
## 12865 8826 166TH AVE NE 98052 REDMOND REDMOND -122.1188 47.68141
##      building_grade square_feet_total_living bedrooms bath_full_count
## 12860          8          1980          3          2
## 12861          7          2920          3          2
## 12862          8          2320          4          1
## 12863          8          2000          4          1
```

```
## 12864          9          2460          4          2
## 12865          7          1150          3          1
##      bath_half_count bath_3qtr_count year_built year_renovated current_zoning
## 12860          1          0        2013          0          R4
## 12861          1          0        2001          0          URPS0
## 12862          1          1        1980          0          R6
## 12863          0          1        1967          0          R4
## 12864          1          0        1990          0          R4
## 12865          0          0        1961          0          R5
##      sq_ft_lot prop_type present_use year  Zip million_above
## 12860      4500          R          2 2016  TRUE      FALSE
## 12861      5487          R          2 2016 FALSE      FALSE
## 12862     35048          R          2 2016  TRUE      FALSE
## 12863      9600          R          2 2016  TRUE      FALSE
## 12864      7615          R          2 2016  TRUE      FALSE
## 12865      8640          R          2 2016  TRUE      FALSE
```

```
# More examples.
housing_df$year <- format(mean(housing_df$Sale.Price))
housing_df$Zip <- housing_df$zip5 == "98052"
head(housing_df$Zip)
```

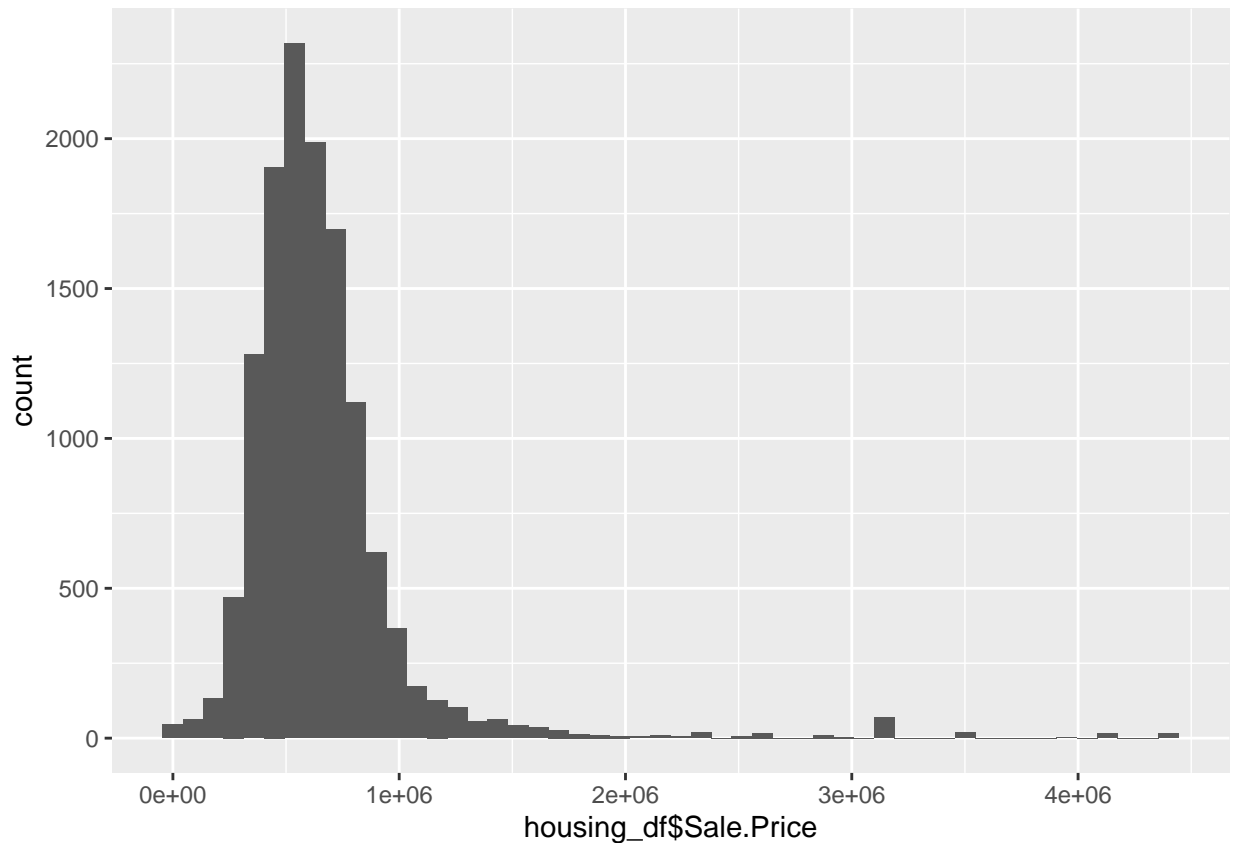
```
## [1] TRUE TRUE TRUE TRUE TRUE FALSE
```

```
housing_df$million_above <- housing_df$Sale.Price >= 1000000
head(housing_df$million_above)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE
```

```
ggplot(housing_df, aes(housing_df$Sale.Price)) + geom_histogram(bins = 50)
```

```
## Warning: Use of 'housing_df$Sale.Price' is discouraged.
## i Use 'Sale.Price' instead.
```



```
Mean_df <- housing_df %>%
  filter(Sale.Price > 1000000) %>%
  mutate(year = format(as.Date(Sale.Date, format="%m/%d/%Y"), "%Y")) %>%
  group_by(year) %>%
  summarize(average_sale_price=mean(Sale.Price), sdev_sale_price=sd(Sale.Price)) %>%
  arrange(year)
```

```
head(Mean_df)
```

```
## # A tibble: 6 x 3
##   year average_sale_price sdev_sale_price
##   <chr>          <dbl>          <dbl>
## 1 2006         1383097.         369128.
## 2 2007         1624951.         588392.
## 3 2008         2574278.         866200.
## 4 2009         1369636.         292656.
## 5 2010         1462957.         645088.
## 6 2011         2841839.         1361869.
```

```
str(Mean_df)
```

```
## tibble [11 x 3] (S3: tbl_df/tbl/data.frame)
## $ year      : chr [1:11] "2006" "2007" "2008" "2009" ...
## $ average_sale_price: num [1:11] 1383097 1624951 2574278 1369636 1462957 ...
## $ sdev_sale_price  : num [1:11] 369128 588392 866200 292656 645088 ...
```

```

#length(Mean_df$monthyear)

ggplot(Mean_df, aes(x = Mean_df$year, y = Mean_df$average_sale_price, label=Mean_df$year)) +
  geom_point(size = 2.1, color="Blue") +
  geom_line() +
  ggtitle("Mean Sales Transaction Per Year") +
  xlab("Year") +
  ylab("Sales Mean Prices") +
  ## geom_text() +
  geom_errorbar(aes(ymin=Mean_df$average_sale_price - Mean_df$sdev_sale_price),
                ymax=Mean_df$average_sale_price + Mean_df$sdev_sale_price,
                width=0.5 )

```

```

## Warning: Use of 'Mean_df$year' is discouraged.
## i Use 'year' instead.

```

```

## Warning: Use of 'Mean_df$average_sale_price' is discouraged.
## i Use 'average_sale_price' instead.

```

```

## Warning: Use of 'Mean_df$year' is discouraged.
## i Use 'year' instead.
## Use of 'Mean_df$year' is discouraged.
## i Use 'year' instead.

```

```

## Warning: Use of 'Mean_df$average_sale_price' is discouraged.
## i Use 'average_sale_price' instead.

```

```

## Warning: Use of 'Mean_df$year' is discouraged.
## i Use 'year' instead.

```

```

## Warning: Use of 'Mean_df$average_sale_price' is discouraged.
## i Use 'average_sale_price' instead.

```

```

## Warning: Use of 'Mean_df$sdev_sale_price' is discouraged.
## i Use 'sdev_sale_price' instead.

```

```

## Warning: Use of 'Mean_df$year' is discouraged.
## i Use 'year' instead.

```

```

## Warning: Use of 'Mean_df$average_sale_price' is discouraged.
## i Use 'average_sale_price' instead.

```

```

## Warning: Use of 'Mean_df$year' is discouraged.
## i Use 'year' instead.

```

```

## 'geom_line()': Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?

```

