DSC520-Week5_Assignment00

Reenie Christudass

2022-07-10

```
library(readxl)
df <- read_excel("C:/Users/chris/dsc520/data/week-7-housing.xlsx")
print(df)
```

```
## # A tibble: 12,865 x 24
##    `Sale Date`         `Sale Price` sale_reason sale_instrument sale_warning
##    <dttm>                     <dbl>       <dbl>           <dbl> <chr>
##  1 2006-01-03 00:00:00       698000           1               3 <NA>
##  2 2006-01-03 00:00:00       649990           1               3 <NA>
##  3 2006-01-03 00:00:00       572500           1               3 <NA>
##  4 2006-01-03 00:00:00       420000           1               3 <NA>
##  5 2006-01-03 00:00:00       369900           1               3 15
##  6 2006-01-03 00:00:00       184667           1              15 18 51
##  7 2006-01-04 00:00:00      1050000           1               3 <NA>
##  8 2006-01-04 00:00:00       875000           1               3 <NA>
##  9 2006-01-04 00:00:00       660000           1               3 <NA>
## 10 2006-01-04 00:00:00       650000           1               3 <NA>
## # ... with 12,855 more rows, and 19 more variables: sitetype <chr>,
## #   addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>, lon <dbl>,
## #   lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>,
## #   bath_3qtr_count <dbl>, year_built <dbl>, year_renovated <dbl>,
## #   current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>
```

```
## Summary of each column
head(apply(df, 2, summary))
```

```
##        Sale Date   Sale Price  sale_reason sale_instrument sale_warning
## Length "12865"     "12865"     "12865"     "12865"         "12865"
## Class  "character" "character" "character" "character"     "character"
## Mode   "character" "character" "character" "character"     "character"
##        sitetype    addr_full   zip5        ctyname     postalctyn  lon
## Length "12865"     "12865"     "12865"     "12865"     "12865"     "12865"
## Class  "character" "character" "character" "character" "character" "character"
## Mode   "character" "character" "character" "character" "character" "character"
##        lat         building_grade square_feet_total_living bedrooms
## Length "12865"     "12865"        "12865"                  "12865"
## Class  "character" "character"    "character"              "character"
## Mode   "character" "character"    "character"              "character"
##        bath_full_count bath_half_count bath_3qtr_count year_built
## Length "12865"         "12865"         "12865"         "12865"
```

```
## Class  "character"     "character"     "character"     "character"
## Mode   "character"     "character"     "character"     "character"
##        year_renovated current_zoning sq_ft_lot   prop_type   present_use
## Length "12865"        "12865"        "12865"     "12865"     "12865"
## Class  "character"    "character"    "character" "character" "character"
## Mode   "character"    "character"    "character" "character" "character"
```

```
install.packages("dplyr", repos="http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/chris/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'dplyr' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'dplyr'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying C:
## \Users\chris\AppData\Local\R\win-library\4.2\00LOCK\dplyr\libs\x64\dplyr.dll
## to C:\Users\chris\AppData\Local\R\win-library\4.2\dplyr\libs\x64\dplyr.dll:
## Permission denied
```

```
## Warning: restored 'dplyr'
```

```
##
## The downloaded binary packages are in
##  C:\Users\chris\AppData\Local\Temp\Rtmpo3swn2\downloaded_packages
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.1
```

```
##
## 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
```

```
install.packages("magrittr", repos="http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/chris/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'magrittr' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'magrittr'

## Warning in file.copy(savedcopy, lib, recursive = TRUE):
## problem copying C:\Users\chris\AppData\Local\R\win-
## library\4.2\00LOCK\magrittr\libs\x64\magrittr.dll to C:
## \Users\chris\AppData\Local\R\win-library\4.2\magrittr\libs\x64\magrittr.dll:
## Permission denied

## Warning: restored 'magrittr'

##
## The downloaded binary packages are in
##  C:\Users\chris\AppData\Local\Temp\Rtmpo3swn2\downloaded_packages
```

```
library(magrittr)
```

```
## Warning: package 'magrittr' was built under R version 4.2.1
```

```
install.packages("tidyverse", repos="http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/chris/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\chris\AppData\Local\Temp\Rtmpo3swn2\downloaded_packages
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.1

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v readr   2.1.2

## Warning: package 'tidyr' was built under R version 4.2.1

## Warning: package 'readr' was built under R version 4.2.1

## Warning: package 'forcats' was built under R version 4.2.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
```

```
##Use the apply function on a variable in your dataset
sapply(df, class)
```

```
## $`Sale Date`
## [1] "POSIXct" "POSIXt"
##
## $`Sale Price`
## [1] "numeric"
##
## $sale_reason
## [1] "numeric"
##
## $sale_instrument
## [1] "numeric"
##
## $sale_warning
## [1] "character"
##
## $sitetype
## [1] "character"
##
## $addr_full
## [1] "character"
##
## $zip5
## [1] "numeric"
##
## $ctyname
## [1] "character"
##
## $postalctyn
## [1] "character"
##
## $lon
## [1] "numeric"
##
## $lat
## [1] "numeric"
##
## $building_grade
## [1] "numeric"
##
## $square_feet_total_living
## [1] "numeric"
##
## $bedrooms
## [1] "numeric"
##
## $bath_full_count
## [1] "numeric"
##
## $bath_half_count
## [1] "numeric"
```

```
##
## $bath_3qtr_count
## [1] "numeric"
##
## $year_built
## [1] "numeric"
##
## $year_renovated
## [1] "numeric"
##
## $current_zoning
## [1] "character"
##
## $sq_ft_lot
## [1] "numeric"
##
## $prop_type
## [1] "character"
##
## $present_use
## [1] "numeric"
```

```
names(df) <- sub(" ", "_", names(df))
print(df)
```

```
## # A tibble: 12,865 x 24
##    Sale_Date           Sale_Price sale_reason sale_instrument sale_warning
##    <dttm>                   <dbl>       <dbl>           <dbl> <chr>
##  1 2006-01-03 00:00:00     698000           1               3 <NA>
##  2 2006-01-03 00:00:00     649990           1               3 <NA>
##  3 2006-01-03 00:00:00     572500           1               3 <NA>
##  4 2006-01-03 00:00:00     420000           1               3 <NA>
##  5 2006-01-03 00:00:00     369900           1               3 15
##  6 2006-01-03 00:00:00     184667           1              15 18 51
##  7 2006-01-04 00:00:00    1050000           1               3 <NA>
##  8 2006-01-04 00:00:00     875000           1               3 <NA>
##  9 2006-01-04 00:00:00     660000           1               3 <NA>
## 10 2006-01-04 00:00:00     650000           1               3 <NA>
## # ... with 12,855 more rows, and 19 more variables: sitetype <chr>,
## #   addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>, lon <dbl>,
## #   lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>,
## #   bath_3qtr_count <dbl>, year_built <dbl>, year_renovated <dbl>,
## #   current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>
```

```
## Group by Property type and Year built
df_grp_Type_year = df %>% group_by(prop_type,year_built)  %>%
                    summarise(Sale_Price = sum(Sale_Price),
                              .groups = 'drop')

df_grp_Type_year
```

```
## # A tibble: 109 x 3
```

```
##    prop_type year_built Sale_Price
##    <chr>          <dbl>      <dbl>
##  1 R               1900    2366998
##  2 R               1903     430000
##  3 R               1905     620000
##  4 R               1906     550000
##  5 R               1909       1070
##  6 R               1910     150000
##  7 R               1912    1859000
##  8 R               1913     915000
##  9 R               1914     835000
## 10 R               1915     456300
## # ... with 99 more rows
```

```
## Mutate (Newly created variables)
## Select statemenet
df %>%
 select(year_built, Sale_Price,square_feet_total_living) %>%  mutate(Sale_Price_by_sq_feet = Sale_Price
```

```
## # A tibble: 12,865 x 4
##    year_built Sale_Price square_feet_total_living Sale_Price_by_sq_feet
##         <dbl>      <dbl>                    <dbl>                 <dbl>
##  1       2003     698000                     2810                 248.
##  2       2006     649990                     2880                 226.
##  3       1987     572500                     2770                 207.
##  4       1968     420000                     1620                 259.
##  5       1980     369900                     1440                 257.
##  6       2005     184667                     4160                  44.4
##  7       1993    1050000                     3960                 265.
##  8       1988     875000                     3720                 235.
##  9       1978     660000                     4160                 159.
## 10       1976     650000                     2760                 236.
## # ... with 12,855 more rows
```

```
## Filter Statement - Filter the dataset only for year 2003
df %>% filter(year_built == "2003")
```

```
## # A tibble: 357 x 24
##    Sale_Date           Sale_Price sale_reason sale_instrument sale_warning
##    <dttm>                   <dbl>       <dbl>           <dbl> <chr>
##  1 2006-01-03 00:00:00     698000           1               3 <NA>
##  2 2006-01-10 00:00:00     482000           1               3 <NA>
##  3 2006-01-31 00:00:00     148000          14              15 18
##  4 2006-02-01 00:00:00     393000           1               3 <NA>
##  5 2006-02-17 00:00:00     390000           1               3 <NA>
##  6 2006-02-23 00:00:00     543000           1               3 40
##  7 2006-02-24 00:00:00     543000           1               3 41
##  8 2006-03-01 00:00:00     585000           1               3 <NA>
##  9 2006-03-02 00:00:00     475000           1               3 <NA>
## 10 2006-03-06 00:00:00     650000           1               3 <NA>
## # ... with 347 more rows, and 19 more variables: sitetype <chr>,
## #   addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>, lon <dbl>,
## #   lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
```

```
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>,
## #   bath_3qtr_count <dbl>, year_built <dbl>, year_renovated <dbl>,
## #   current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>
```

```
## Arrange by Year_bult
arrange(df, year_built)
```

```
## # A tibble: 12,865 x 24
##    Sale_Date           Sale_Price sale_reason sale_instrument sale_warning
##    <dttm>                   <dbl>       <dbl>           <dbl> <chr>
##  1 2006-03-13 00:00:00     455000           1               3 <NA>
##  2 2006-10-04 00:00:00     675000           1               3 <NA>
##  3 2007-02-16 00:00:00     550000           8               3 12
##  4 2009-12-04 00:00:00     400000          18               4 <NA>
##  5 2010-07-06 00:00:00        698           1              26 24
##  6 2013-05-23 00:00:00     286300           4              18 15 31
##  7 2007-03-16 00:00:00     430000           1               3 <NA>
##  8 2006-10-18 00:00:00     620000           1               3 <NA>
##  9 2012-02-28 00:00:00     550000           1               3 16 45
## 10 2010-05-14 00:00:00       1070           1              26 24
## # ... with 12,855 more rows, and 19 more variables: sitetype <chr>,
## #   addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>, lon <dbl>,
## #   lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>,
## #   bath_3qtr_count <dbl>, year_built <dbl>, year_renovated <dbl>,
## #   current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>
```

```
##Using the purrr package - perform 2 functions on your dataset.
##You could use zip_n, keep, discard, compact, etc.
df %>% map(is.numeric)
```

```
## $Sale_Date
## [1] FALSE
##
## $Sale_Price
## [1] TRUE
##
## $sale_reason
## [1] TRUE
##
## $sale_instrument
## [1] TRUE
##
## $sale_warning
## [1] FALSE
##
## $sitetype
## [1] FALSE
##
## $addr_full
## [1] FALSE
##
## $zip5
```

```
## [1] TRUE
##
## $ctyname
## [1] FALSE
##
## $postalctyn
## [1] FALSE
##
## $lon
## [1] TRUE
##
## $lat
## [1] TRUE
##
## $building_grade
## [1] TRUE
##
## $square_feet_total_living
## [1] TRUE
##
## $bedrooms
## [1] TRUE
##
## $bath_full_count
## [1] TRUE
##
## $bath_half_count
## [1] TRUE
##
## $bath_3qtr_count
## [1] TRUE
##
## $year_built
## [1] TRUE
##
## $year_renovated
## [1] TRUE
##
## $current_zoning
## [1] FALSE
##
## $sq_ft_lot
## [1] TRUE
##
## $prop_type
## [1] FALSE
##
## $present_use
## [1] TRUE
```

```
## split some data
## Create new variables
df <- df %>% separate(Sale_Date, c('Year', 'Month','Date'))
df
```

```
## # A tibble: 12,865 x 26
##    Year  Month Date  Sale_Price sale_reason sale_instrument sale_warning
##    <chr> <chr> <chr>      <dbl>       <dbl>           <dbl> <chr>
##  1 2006  01    03        698000           1               3 <NA>
##  2 2006  01    03        649990           1               3 <NA>
##  3 2006  01    03        572500           1               3 <NA>
##  4 2006  01    03        420000           1               3 <NA>
##  5 2006  01    03        369900           1               3 15
##  6 2006  01    03        184667           1              15 18 51
##  7 2006  01    04       1050000           1               3 <NA>
##  8 2006  01    04        875000           1               3 <NA>
##  9 2006  01    04        660000           1               3 <NA>
## 10 2006  01    04        650000           1               3 <NA>
## # ... with 12,855 more rows, and 19 more variables: sitetype <chr>,
## #   addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>, lon <dbl>,
## #   lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>,
## #   bath_3qtr_count <dbl>, year_built <dbl>, year_renovated <dbl>,
## #   current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>
```

```
## then concatenate the results back together
df
```

```
## # A tibble: 12,865 x 26
##    Year  Month Date  Sale_Price sale_reason sale_instrument sale_warning
##    <chr> <chr> <chr>      <dbl>       <dbl>           <dbl> <chr>
##  1 2006  01    03        698000           1               3 <NA>
##  2 2006  01    03        649990           1               3 <NA>
##  3 2006  01    03        572500           1               3 <NA>
##  4 2006  01    03        420000           1               3 <NA>
##  5 2006  01    03        369900           1               3 15
##  6 2006  01    03        184667           1              15 18 51
##  7 2006  01    04       1050000           1               3 <NA>
##  8 2006  01    04        875000           1               3 <NA>
##  9 2006  01    04        660000           1               3 <NA>
## 10 2006  01    04        650000           1               3 <NA>
## # ... with 12,855 more rows, and 19 more variables: sitetype <chr>,
## #   addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>, lon <dbl>,
## #   lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>,
## #   bath_3qtr_count <dbl>, year_built <dbl>, year_renovated <dbl>,
## #   current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>
```

```
df$Sale_Date = paste(df$Year,"+",df$Month ,"+",df$Date)
df
```

```
## # A tibble: 12,865 x 27
##    Year  Month Date  Sale_Price sale_reason sale_instrument sale_warning
##    <chr> <chr> <chr>      <dbl>       <dbl>           <dbl> <chr>
##  1 2006  01    03        698000           1               3 <NA>
##  2 2006  01    03        649990           1               3 <NA>
##  3 2006  01    03        572500           1               3 <NA>
##  4 2006  01    03        420000           1               3 <NA>
```

```
##  5 2006  01    03        369900           1               3 15
##  6 2006  01    03        184667           1              15 18 51
##  7 2006  01    04       1050000           1               3 <NA>
##  8 2006  01    04        875000           1               3 <NA>
##  9 2006  01    04        660000           1               3 <NA>
## 10 2006  01    04        650000           1               3 <NA>
## # ... with 12,855 more rows, and 20 more variables: sitetype <chr>,
## #   addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>, lon <dbl>,
## #   lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>,
## #   bath_3qtr_count <dbl>, year_built <dbl>, year_renovated <dbl>,
## #   current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>,
## #   Sale_Date <chr>
```

```
df <- read.csv("C:/Users/chris/dsc520/data/scores.csv")
df
```

```
##    Count Score Section
## 1     10   200  Sports
## 2     10   205  Sports
## 3     20   235  Sports
## 4     10   240  Sports
## 5     10   250  Sports
## 6     10   265 Regular
## 7     10   275 Regular
## 8     30   285  Sports
## 9     10   295 Regular
## 10    10   300 Regular
## 11    20   300  Sports
## 12    10   305  Sports
## 13    10   305 Regular
## 14    10   310 Regular
## 15    10   310  Sports
## 16    20   320 Regular
## 17    10   305 Regular
## 18    10   315  Sports
## 19    20   320 Regular
## 20    10   325 Regular
## 21    10   325  Sports
## 22    20   330 Regular
## 23    10   330  Sports
## 24    30   335  Sports
## 25    10   335 Regular
## 26    20   340 Regular
## 27    10   340  Sports
## 28    30   350 Regular
## 29    20   360 Regular
## 30    10   360  Sports
## 31    20   365 Regular
## 32    20   365  Sports
## 33    10   370  Sports
## 34    10   370 Regular
## 35    20   375 Regular
## 36    10   375  Sports
```

```
## 37    20   380 Regular
## 38    10   395  Sports
```

```
print(df)
```

```
##    Count Score Section
## 1     10   200  Sports
## 2     10   205  Sports
## 3     20   235  Sports
## 4     10   240  Sports
## 5     10   250  Sports
## 6     10   265 Regular
## 7     10   275 Regular
## 8     30   285  Sports
## 9     10   295 Regular
## 10    10   300 Regular
## 11    20   300  Sports
## 12    10   305  Sports
## 13    10   305 Regular
## 14    10   310 Regular
## 15    10   310  Sports
## 16    20   320 Regular
## 17    10   305 Regular
## 18    10   315  Sports
## 19    20   320 Regular
## 20    10   325 Regular
## 21    10   325  Sports
## 22    20   330 Regular
## 23    10   330  Sports
## 24    30   335  Sports
## 25    10   335 Regular
## 26    20   340 Regular
## 27    10   340  Sports
## 28    30   350 Regular
## 29    20   360 Regular
## 30    10   360  Sports
## 31    20   365 Regular
## 32    20   365  Sports
## 33    10   370  Sports
## 34    10   370 Regular
## 35    20   375 Regular
## 36    10   375  Sports
## 37    20   380 Regular
## 38    10   395  Sports
```

```
## Use the cbind and rbind function on your dataset
df = rbind(df, data.frame("Count"="999", "Score"="5555", Section="Regular"))
df
```

```
##    Count Score Section
## 1     10   200  Sports
## 2     10   205  Sports
## 3     20   235  Sports
```

```
## 4     10   240  Sports
## 5     10   250  Sports
## 6     10   265 Regular
## 7     10   275 Regular
## 8     30   285  Sports
## 9     10   295 Regular
## 10    10   300 Regular
## 11    20   300  Sports
## 12    10   305  Sports
## 13    10   305 Regular
## 14    10   310 Regular
## 15    10   310  Sports
## 16    20   320 Regular
## 17    10   305 Regular
## 18    10   315  Sports
## 19    20   320 Regular
## 20    10   325 Regular
## 21    10   325  Sports
## 22    20   330 Regular
## 23    10   330  Sports
## 24    30   335  Sports
## 25    10   335 Regular
## 26    20   340 Regular
## 27    10   340  Sports
## 28    30   350 Regular
## 29    20   360 Regular
## 30    10   360  Sports
## 31    20   365 Regular
## 32    20   365  Sports
## 33    10   370  Sports
## 34    10   370 Regular
## 35    20   375 Regular
## 36    10   375  Sports
## 37    20   380 Regular
## 38    10   395  Sports
## 39   999  5555 Regular
```

```
##Column bind
df <- data.frame(c1 = c(200, 205, 295, 300))
df
```

```
##    c1
## 1 200
## 2 205
## 3 295
## 4 300
```

```
df2 <- data.frame(c4 = c(200, 205, 295, 300),
 c5 = c("Football", "Softball", "Cricket", "Tennis"))
newDf <- cbind(df, df2)
newDf
```

```
##    c1  c4       c5
```

```
## 1 200 200 Football
## 2 205 205 Softball
## 3 295 295  Cricket
## 4 300 300   Tennis
```

```
## KEEP, DISCORD, COMPACT
##df
##ls1 <- (list(df$zip5))
##ls1 %>% keep(names(.) == "9")
newDf
```

```
##    c1  c4       c5
## 1 200 200 Football
## 2 205 205 Softball
## 3 295 295  Cricket
## 4 300 300   Tennis
```

```
l <- list(
    list(col1 = 'to keep', col2 = 1),
    list(col1 = 'to discard', col2 = 2)
)
purrr::keep(l, ~ .x[['col1']] == 'to keep')
```

```
## [[1]]
## [[1]]$col1
## [1] "to keep"
##
## [[1]]$col2
## [1] 1
```

```
purrr::discard(l, ~ .x[['col1']] == 'to discard')
```

```
## [[1]]
## [[1]]$col1
## [1] "to keep"
##
## [[1]]$col2
## [1] 1
```