Assignment Week 5

Karlie Schwartzwald

2022-07-08

First, I open the 2014 American Community Survey data:

1. Dplyr package

```
library(readxl)
setwd('C:/Users/karli/OneDrive/Documents/Data_Science/DSC520_Stats_for_DS')
week_6_housing <- read_excel("week-6-housing.xlsx")
colnames(week_6_housing)[2] <- "Sale_Price"
housing_complete <- na.omit(week_6_housing)
head(housing_complete)
```

```
## # A tibble: 6 x 24
##   `Sale Date`         Sale_Price sale_reason sale_instrument sale_warning
##   <dttm>                   <dbl>       <dbl>           <dbl> <chr>
## 1 2006-01-03 00:00:00     369900           1               3 15
## 2 2006-01-31 00:00:00     148000          14              15 18
## 3 2006-02-10 00:00:00     560000           1               3 45
## 4 2006-02-10 00:00:00     560000           1               3 45
## 5 2006-02-13 00:00:00    1520000          18               3 52
## 6 2006-02-14 00:00:00     275000           1               3 26
## # ... with 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>
```

```
housing_complete %>%
  group_by(bedrooms) %>%
  summarize(mean(square_feet_total_living))
```

Using the GroupBy function:

```
## # A tibble: 9 x 2
##   bedrooms `mean(square_feet_total_living)`
##      <dbl>                            <dbl>
```

```
## 1        0                          3525
## 2        1                           897.
## 3        2                          1383.
## 4        3                          1881.
## 5        4                          2830.
## 6        5                          3367.
## 7        6                          3607.
## 8        7                          2593.
## 9        9                          4460
```

```
housing_complete %>%
 group_by(building_grade) %>%
 summarize(mean(Sale_Price))
```

Using the Summarize function:

```
## # A tibble: 9 x 2
##   building_grade `mean(Sale_Price)`
##            <dbl>              <dbl>
## 1              5            1168150
## 2              6             407087.
## 3              7             363904.
## 4              8             760210.
## 5              9             874999.
## 6             10            1145904.
## 7             11             934732.
## 8             12            1943025.
## 9             13            1957080
```

```
housing_complete %>%
  select(Sale_Price, sq_ft_lot) %>%
  mutate(Sale_Price/sq_ft_lot)
```

Using the Mutate function:

```
## # A tibble: 1,108 x 3
##    Sale_Price sq_ft_lot `Sale_Price/sq_ft_lot`
##         <dbl>     <dbl>                  <dbl>
##  1     369900      7526                   49.1
##  2     148000      3430                   43.1
##  3     560000      7910                   70.8
##  4     560000      8566                   65.4
##  5    1520000     19173                   79.3
##  6     275000      2868                   95.9
##  7     375000     14725                   25.5
##  8     855000      6126                  140.
##  9     855000      9301                   91.9
## 10     855000      6675                  128.
## # ... with 1,098 more rows
```

```
housing_complete %>% filter_("year_renovated != 0")
```

Using the Filter function:

```
## Warning: ‘filter_()‘ was deprecated in dplyr 0.7.0.
## Please use ‘filter()‘ instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was generated.
```

```
## # A tibble: 28 x 24
##    ‘Sale Date‘         Sale_Price sale_reason sale_instrument sale_warning
##    <dttm>                   <dbl>       <dbl>           <dbl> <chr>
##  1 2006-02-13 00:00:00    1520000          18               3 52
##  2 2006-06-27 00:00:00     510000           1               3 26
##  3 2006-09-28 00:00:00     375000           1               3 26
##  4 2006-11-03 00:00:00     684000           1               3 26
##  5 2007-02-16 00:00:00     550000           8               3 12
##  6 2007-08-06 00:00:00    1405000           1               3 40 52
##  7 2007-08-13 00:00:00    1405000           1               4 41 52
##  8 2008-10-09 00:00:00    2000000           1               3 45
##  9 2009-04-10 00:00:00      29537          14              15 18 22 51
## 10 2010-06-16 00:00:00     615000           1               3 26
## # ... with 18 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>
```

```
housing_complete %>%
  select_('Sale_Price')
```

Using the Select function:

```
## Warning: ‘select_()‘ was deprecated in dplyr 0.7.0.
## Please use ‘select()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was generated.
```

```
## # A tibble: 1,108 x 1
##    Sale_Price
##         <dbl>
##  1     369900
##  2     148000
##  3     560000
##  4     560000
##  5    1520000
```

```
##  6     275000
##  7     375000
##  8     855000
##  9     855000
## 10     855000
## # ... with 1,098 more rows
```

```
housing_complete %>%
  group_by(bedrooms) %>%
  summarize(avgprice=mean(Sale_Price)) %>%
  arrange(avgprice)
```

Using the Arrange function:

```
## # A tibble: 9 x 2
##   bedrooms avgprice
##      <dbl>    <dbl>
## 1        7  373366.
## 2        2  417312.
## 3        3  422280.
## 4        9  581500
## 5        0  725000
## 6        4  825943.
## 7        6  878071.
## 8        5 1144987.
## 9        1 1371650
```

2. Purrr Package

```
# You could use zip_n, keep, discard, compact, etc.
library(purrr)
```

Using the keep function:

```
##
## Attaching package: 'purrr'

## The following object is masked from 'package:magrittr':
##
##     set_names
```

```
expensive_sales <- housing_complete$Sale_Price %>%
  keep(~{mean(.x)>600000})
head(expensive_sales)
```

```
## [1] 1520000  855000  855000  855000  750000  707000
```

```
# You could use zip_n, keep, discard, compact, etc.
complete_sale_price <- housing_complete$Sale_Price %>%
  compact()
head(complete_sale_price)
```

Using the function:

```
## [1]  369900  148000  560000  560000 1520000  275000
```

3. cbind and rbind

```
bathrooms <- cbind('Full Bath'=week_6_housing$bath_full_count, '3/4 Bath'=week_6_housing$bath_3qtr_coun
head(bathrooms)
```

cbind

```
##      Full Bath 3/4 Bath 1/2 Bath
## [1,]         2        0        1
## [2,]         2        1        0
## [3,]         1        1        1
## [4,]         1        1        0
## [5,]         1        1        0
## [6,]         2        1        1
```

```
cheap_sales <- week_6_housing %>%
  filter(Sale_Price<600000) %>%
  cbind()
head(cheap_sales)
```

rbind

```
##    Sale Date Sale_Price sale_reason sale_instrument sale_warning sitetype
## 1 2006-01-03     572500           1               3         <NA>       R1
## 2 2006-01-03     420000           1               3         <NA>       R1
## 3 2006-01-03     369900           1               3           15       R1
## 4 2006-01-03     184667           1              15        18 51       R1
## 5 2006-01-04     599950           1               3         <NA>       R1
## 6 2006-01-04     526787           1               3         <NA>       R1
##            addr_full  zip5 ctyname postalctyn       lon      lat building_grade
## 1 13315 174TH AVE NE 98052    <NA>    REDMOND -122.1085 47.71986              8
## 2  3303 178TH AVE NE 98052 REDMOND    REDMOND -122.1037 47.63914              8
## 3  16126 NE 108TH CT 98052 REDMOND    REDMOND -122.1242 47.69748              7
## 4   8101 229TH DR NE 98053    <NA>    REDMOND -122.0341 47.67545              7
## 5   14924 NE 74TH CT 98052 REDMOND    REDMOND -122.1411 47.67142              9
## 6   7858 148TH CT NE 98052 REDMOND    REDMOND -122.1425 47.67407              8
```

```
##   square_feet_total_living bedrooms bath_full_count bath_half_count
## 1                     2770        4               1               1
## 2                     1620        3               1               0
## 3                     1440        3               1               0
## 4                     4160        4               2               1
## 5                     2180        3               2               1
## 6                     2480        3               2               1
##   bath_3qtr_count year_built year_renovated current_zoning sq_ft_lot prop_type
## 1               1       1987              0             R6      8444         R
## 2               1       1968              0             R4      9600         R
## 3               1       1980              0             R6      7526         R
## 4               1       2005              0           URPSO      7280         R
## 5               0       1988              0             R5      7949         R
## 6               0       2005              0             R5      2647         R
##   present_use
## 1           2
## 2           2
## 3           2
## 4           2
## 5           2
## 6           2
```

4. Split a string, then concatenate the results back together

```
library(stringr)

# create string
string1 <- "I need to split a string~ I will split based on the tilde~ Hopefully there will be enough ti
print(string1)
```

```
## [1] "I need to split a string~ I will split based on the tilde~ Hopefully there will be enough tilde
```

```
# Split String
split_string1 <- str_split(string=string1,pattern='~')
print(split_string1)
```

```
## [[1]]
## [1] "I need to split a string"
## [2] " I will split based on the tilde"
## [3] " Hopefully there will be enough tilde to split on"
## [4] " Testing now."
```

```
paste(unlist(split_string1), sep = " ", collapse = "~")
```

```
## [1] "I need to split a string~ I will split based on the tilde~ Hopefully there will be enough tilde
```