Assignment5.2_BrownLincoln.R

x

2021-07-11

```
#Using either the same dataset(s) you used in the previous weeks' exercise or a brand-new dataset of yo
#perform the following transformations
#(Remember, anything you learn about the Housing dataset in these two weeks can be used for a later exe

#Using the dplyr package, use the 6 different operations to analyze/transform the data -
#GroupBy, Summarize, Mutate, Filter, Select, and Arrange -
#    X         X         X        X        X           X
#Remember this isn't just modifying data, you are learning about your data also -
#so play around and start to understand your dataset in more detail
#Using the purrr package - perform 2 functions on your dataset.  You could use zip_n, keep, discard, co
#
#Use the cbind and rbind function on your dataset
#Split a string, then concatenate the results back together

library(readxl)
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(stringr)


housing <- read_excel("/media/x/disk/School/DSC520/datasets/week-6-housing.xlsx")
dim(housing)
```

```
## [1] 12865    24
```

```
names <- colnames(housing)
names
```

```
##  [1] "Sale Date"                "Sale Price"
##  [3] "sale_reason"              "sale_instrument"
##  [5] "sale_warning"             "sitetype"
##  [7] "addr_full"                "zip5"
##  [9] "ctyname"                  "postalctyn"
## [11] "lon"                      "lat"
```

```
## [13] "building_grade"           "square_feet_total_living"
## [15] "bedrooms"                 "bath_full_count"
## [17] "bath_half_count"          "bath_3qtr_count"
## [19] "year_built"               "year_renovated"
## [21] "current_zoning"           "sq_ft_lot"
## [23] "prop_type"                "present_use"
```

```
housing
```

```
## # 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>
```

```
#Remove unnecessary columns
#Removing all columns that contain data that is useless without the codebook.
# I am removing the ctyname column because the zip5 column is better populated
remove_cols <- c(3,4,5,6,9,10,13,21,23)
refined <- housing |> select(-remove_cols)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(remove_cols)` instead of `remove_cols` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
avg_zip <- aggregate(`Sale Price` ~ zip5, housing, mean)

dim(refined)
```

```
## [1] 12865    15
```

```
cost_size <- housing |> select(c(2,14,22))

#Plyr package
library(plyr)
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
housing_size <- ddply(refined, .(square_feet_total_living), transform,
                      house.size = cut(square_feet_total_living, breaks = c(-Inf, 1000, 2000, 3000, Inf
                                       labels = c("Tiny", "Small", "Medium", "Large"))
)


#Dplyr package
library(dplyr)
sizes = unique(housing_size$house.size)
#Calculate the cost per sqft by the number of living square feet
cost_liv_sqft <- housing_size |> mutate(Sale.Price, `Cost.sqft` = Sale.Price / square_feet_total_living

#Calculate the cost per sqft by the total square feet of the lot
cost_lot_sqft <- housing_size |> mutate(Sale.Price, `Cost.sqft` = Sale.Price / sq_ft_lot)

#Group By
liv_size <- cost_liv_sqft |> group_by(house.size)
lot_size <- cost_lot_sqft |> group_by(house.size)

# Arrange by house size group and by Sale Price

arranged_liv <- liv_size |> arrange(desc(Sale.Price), desc(house.size))
arranged_lot <- lot_size |> arrange(desc(Sale.Price), desc(house.size))

# Display the mean house price for each zip code
means_zip <- housing |> group_by(zip5) |> summarize_at(vars(`Sale Price`),list(~ mean(.)))

# Display the count of house prices for each zip code
zip_count <- housing |> group_by(zip5) |> summarize_at(vars(`Sale Price`),list(length))

unique_cols <- map(liv_size, unique)
unique_cols |> str()
```

```
## List of 17
##  $ Sale.Date               : POSIXct[1:2933], format: "2016-12-10" "2007-08-22" ...
##  $ Sale.Price              : num [1:4019] 687500 1640000 1085000 650000 725000 ...
##  $ addr_full               : chr [1:9737] "5629 236TH AVE NE" "2350 W LAKE SAMMAMISH PKWY NE" "3122 W
##  $ zip5                    : num [1:4] 98053 98052 98074 98059
##  $ lon                     : num [1:9736] -122 -122 -122 -122 -122 ...
##  $ lat                     : num [1:9733] 47.7 47.6 47.6 47.6 47.6 ...
##  $ square_feet_total_living: num [1:654] 240 310 340 410 430 480 530 540 550 570 ...
##  $ bedrooms                : num [1:12] 0 1 2 3 4 5 6 7 9 8 ...
##  $ bath_full_count         : num [1:8] 0 1 2 3 4 5 23 6
##  $ bath_half_count         : num [1:7] 0 1 2 4 3 6 8
```

```
##  $ bath_3qtr_count         : num [1:7] 0 1 2 3 4 7 8
##  $ year_built              : num [1:109] 1953 1964 1954 1945 1983 ...
##  $ year_renovated          : num [1:40] 0 1992 2000 1984 2006 ...
##  $ sq_ft_lot               : num [1:6038] 120661 19556 29933 80346 65340 ...
##  $ present_use             : num [1:8] 2 6 29 300 8 9 3 0
##  $ house.size              : Factor w/ 4 levels "Tiny","Small",..: 1 2 3 4
##  $ Cost.sqft               : num [1:10734] 2865 5290 3191 1585 1686 ...
```

```
#Houses over $1 million
high_cost <- cost_liv_sqft$Sale.Price |> keep(~ (.x) > 1500000)

#Houses under 350,000
low_cost <- cost_liv_sqft$Sale.Price |> discard(~ (.x) > 150000)

#Split and rejoin a string
class(cost_liv_sqft$Sale.Date)
```

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

```
dates <- cost_liv_sqft$Sale.Date
years <- dates |> str_sub(start=1, end=4)
months <- dates |> str_sub(start=6, end=7)

cost_liv_sqft$mmyy <- paste(months, years, sep='-')

max(housing$bedrooms)
```

```
## [1] 11
```

```
min(housing$bedrooms)
```

```
## [1] 0
```

```
filter(housing, housing$bedrooms==0)
```

```
## # A tibble: 19 x 24
##    `Sale Date`         `Sale Price` sale_reason sale_instrument sale_warning
##    <dttm>                     <dbl>       <dbl>           <dbl> <chr>
##  1 2006-02-15 00:00:00      1390000           1               3 <NA>
##  2 2006-02-27 00:00:00       229000          18               3 13
##  3 2006-07-19 00:00:00       804000          14               3 22
##  4 2006-08-29 00:00:00       900000           1               3 <NA>
##  5 2006-12-20 00:00:00      1085000           1               3 <NA>
##  6 2007-07-16 00:00:00       475000           8               3 12 45
##  7 2007-08-22 00:00:00      1640000           1               3 <NA>
##  8 2009-10-07 00:00:00       745000           1               3 <NA>
##  9 2010-08-20 00:00:00      1055000           1               3 45
## 10 2010-08-31 00:00:00       915000           1               3 <NA>
## 11 2011-05-05 00:00:00       330535           1               3 54
## 12 2012-06-12 00:00:00       150000           1              15 18 51
## 13 2013-08-21 00:00:00      1300000           1               3 10 56
## 14 2014-06-24 00:00:00      1295648           1               3 <NA>
## 15 2015-06-15 00:00:00       743000           1               3 <NA>
## 16 2016-03-31 00:00:00       953830           1               3 <NA>
## 17 2016-06-23 00:00:00       925000           1               3 <NA>
## 18 2016-07-20 00:00:00       413617           1               3 <NA>
## 19 2016-12-10 00:00:00       687500           1               3 <NA>
```

```
## # ... 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>
```

```
# Create separate dataframes for each size of house
tiny <- filter(cost_liv_sqft, cost_liv_sqft$house.size==sizes[1])
small <- filter(cost_liv_sqft, cost_liv_sqft$house.size==sizes[2])
medium <- filter(cost_liv_sqft, cost_liv_sqft$house.size==sizes[3])
large <- filter(cost_liv_sqft, cost_liv_sqft$house.size==sizes[4])

# Recombine the sizes into one data frame
combine <- rbind(tiny, small, medium, large)
dim(combine)
```

```
## [1] 12865    18
```

```
# Restore missing columns removed at beginning, including new col added
restore <- housing |> select(remove_cols) |> cbind(combine)
dim(restore)
```

```
## [1] 12865    27
```

```
str(restore)
```

```
## 'data.frame':    12865 obs. of  27 variables:
##  $ sale_reason             : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ sale_instrument         : num  3 3 3 3 3 15 3 3 3 3 ...
##  $ sale_warning            : chr  NA NA NA NA ...
##  $ sitetype                : chr  "R1" "R1" "R1" "R1" ...
##  $ ctyname                 : chr  "REDMOND" "REDMOND" NA "REDMOND" ...
##  $ postalctyn              : chr  "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ building_grade          : num  9 9 8 8 7 7 10 10 9 8 ...
##  $ current_zoning          : chr  "R4" "R4" "R6" "R4" ...
##  $ prop_type               : chr  "R" "R" "R" "R" ...
##  $ Sale.Date               : POSIXct, format: "2016-12-10" "2007-08-22" ...
##  $ Sale.Price              : num  687500 1640000 1085000 650000 725000 ...
##  $ addr_full               : chr  "5629 236TH AVE NE" "2350 W LAKE SAMMAMISH PKWY NE" "3122 W AMES LA
##  $ zip5                    : num  98053 98052 98053 98053 98053 ...
##  $ lon                     : num  -122 -122 -122 -122 -122 ...
##  $ lat                     : num  47.7 47.6 47.6 47.6 47.6 ...
##  $ square_feet_total_living: num  240 310 340 410 430 480 480 530 540 550 ...
##  $ bedrooms                : num  0 0 0 1 1 1 1 1 1 1 ...
##  $ bath_full_count         : num  0 0 0 0 1 0 0 1 0 1 ...
##  $ bath_half_count         : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ bath_3qtr_count         : num  0 0 1 1 0 0 0 0 0 0 ...
##  $ year_built              : num  1953 1964 1954 1945 1983 ...
##  $ year_renovated          : num  0 1992 0 0 0 ...
##  $ sq_ft_lot               : num  120661 19556 29933 80346 65340 ...
##  $ present_use             : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ house.size              : Factor w/ 4 levels "Tiny","Small",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Cost.sqft               : num  2865 5290 3191 1585 1686 ...
##  $ mmyy                    : chr  "12-2016" "08-2007" "12-2006" "11-2016" ...
```