

January 14, 2024

The results below are generated from an R script.

```
library(readxl)
library(dplyr)
read_excel("Housing.xlsx")

## # A tibble: 12,865 x 24
##   `Sale Date`      `Sale Price` sale_reason sale_instrument sale_warning sitetype
##   <dtm>           <dbl>      <dbl>         <dbl> <chr>         <chr>
## 1 2006-01-03 00:00:00    698000         1           3 <NA>         R1
## 2 2006-01-03 00:00:00    649990         1           3 <NA>         R1
## 3 2006-01-03 00:00:00    572500         1           3 <NA>         R1
## 4 2006-01-03 00:00:00    420000         1           3 <NA>         R1
## 5 2006-01-03 00:00:00    369900         1           3 15          R1
## 6 2006-01-03 00:00:00    184667         1          15 18 51        R1
## 7 2006-01-04 00:00:00   1050000         1           3 <NA>         R1
## 8 2006-01-04 00:00:00    875000         1           3 <NA>         R1
## 9 2006-01-04 00:00:00    660000         1           3 <NA>         R1
## 10 2006-01-04 00:00:00    650000         1           3 <NA>         R1
## # i 12,855 more rows
## # i 18 more variables: 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>

housingData <- read_excel("Housing.xlsx")

# Using the dplyr package, use the 6 different operations to analyze/transform
# the data - GroupBy, Summarize, Mutate, Filter, Select, and Arrange - 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

# Group By number of bedrooms

housingData %>% group_by(bedrooms)

## # A tibble: 12,865 x 24
## # Groups:   bedrooms [12]
##   `Sale Date`      `Sale Price` sale_reason sale_instrument sale_warning sitetype
##   <dtm>           <dbl>      <dbl>         <dbl> <chr>         <chr>
## 1 2006-01-03 00:00:00    698000         1           3 <NA>         R1
## 2 2006-01-03 00:00:00    649990         1           3 <NA>         R1
## 3 2006-01-03 00:00:00    572500         1           3 <NA>         R1
## 4 2006-01-03 00:00:00    420000         1           3 <NA>         R1
## 5 2006-01-03 00:00:00    369900         1           3 15          R1
```

```

## 6 2006-01-03 00:00:00      184667      1      15 18 51      R1
## 7 2006-01-04 00:00:00     1050000      1          3 <NA>      R1
## 8 2006-01-04 00:00:00     875000      1          3 <NA>      R1
## 9 2006-01-04 00:00:00     660000      1          3 <NA>      R1
## 10 2006-01-04 00:00:00     650000      1          3 <NA>      R1
## # i 12,855 more rows
## # i 18 more variables: 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>

# Summarize avg price of each group

housingData %>% group_by(bedrooms) %>% summarize(AvgPrice = mean(`Sale Price`))

## # A tibble: 12 x 2
##   bedrooms AvgPrice
##   <dbl>     <dbl>
## 1       0  844059.
## 2       1  722814.
## 3       2  544946.
## 4       3  564959.
## 5       4  735910.
## 6       5  836974.
## 7       6  767494.
## 8       7 1307282.
## 9       8 1122500
## 10      9  581500
## 11     10  450000
## 12     11 1825000

# Mutate to determine if house is expensive based on median price

median(housingData$`Sale Price`)

## [1] 593000

housingData %>% select(`Sale Price`, bedrooms) %>%
  mutate(expensive = ifelse(`Sale Price` > median(`Sale Price`), "yes", "no"))

## # A tibble: 12,865 x 3
##   `Sale Price` bedrooms expensive
##   <dbl>     <dbl> <chr>
## 1     698000         4 yes
## 2     649990         4 yes
## 3     572500         4 no
## 4     420000         3 no
## 5     369900         3 no
## 6     184667         4 no
## 7    1050000         5 yes
## 8     875000         4 yes
## 9     660000         4 yes
## 10    650000         4 yes
## # i 12,855 more rows

```

```

# Filter to only include houses with 3 bedrooms

housingData %>% filter(bedrooms == 3)

## # A tibble: 4,493 x 24
##   `Sale Date`      `Sale Price` sale_reason sale_instrument sale_warning sitetype
##   <dtm>          <dbl>      <dbl>          <dbl> <chr>      <chr>
## 1 2006-01-03 00:00:00      420000          1            3 <NA>      R1
## 2 2006-01-03 00:00:00      369900          1            3 15        R1
## 3 2006-01-04 00:00:00      599950          1            3 <NA>      R1
## 4 2006-01-04 00:00:00      526787          1            3 <NA>      R1
## 5 2006-01-04 00:00:00      165000          1            3 <NA>      R1
## 6 2006-01-05 00:00:00      803000          1            3 <NA>      R1
## 7 2006-01-05 00:00:00      507950          1            3 <NA>      R1
## 8 2006-01-09 00:00:00      501000          1            3 <NA>      R1
## 9 2006-01-09 00:00:00      372500          1            3 <NA>      R1
## 10 2006-01-10 00:00:00      482000          1            3 <NA>      R1
## # i 4,483 more rows
## # i 18 more variables: 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>

# Select bedrooms, full bathrooms, and sale price

housingData %>% select(bedrooms, bath_full_count, `Sale Price`)

## # A tibble: 12,865 x 3
##   bedrooms bath_full_count `Sale Price`
##   <dbl>      <dbl>          <dbl>
## 1         4             2      698000
## 2         4             2      649990
## 3         4             1      572500
## 4         3             1      420000
## 5         3             1      369900
## 6         4             2      184667
## 7         5             3     1050000
## 8         4             2      875000
## 9         4             2      660000
## 10        4             1      650000
## # i 12,855 more rows

# Arrange in ascending order by sale price

housingData %>% arrange(`Sale Price`)

## # A tibble: 12,865 x 24
##   `Sale Date`      `Sale Price` sale_reason sale_instrument sale_warning sitetype
##   <dtm>          <dbl>      <dbl>          <dbl> <chr>      <chr>
## 1 2010-07-06 00:00:00         698          1            26 24        R1
## 2 2010-07-06 00:00:00         698          1            26 24        R1
## 3 2009-12-29 00:00:00         873          1            26 24        R1
## 4 2010-01-28 00:00:00         873          1            26 24 32        R1
## 5 2009-12-22 00:00:00         998          1            26 24        R1

```

```

## 6 2007-03-20 00:00:00      1000      1      15 18 51      R1
## 7 2010-05-14 00:00:00      1070      1      26 24      R1
## 8 2008-03-31 00:00:00      1500      1       3 51      R1
## 9 2009-09-24 00:00:00      1500      1      26 24      R1
## 10 2012-10-05 00:00:00     2031     18      15 18 31      R1
## # i 12,855 more rows
## # i 18 more variables: 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.

library(purrr)

# Map to find median

housingData %>% map(median)

## $`Sale Date`
## [1] "2011-11-17 UTC"
##
## $`Sale Price`
## [1] 593000
##
## $sale_reason
## [1] 1
##
## $sale_instrument
## [1] 3
##
## $sale_warning
## [1] NA
##
## $sitetype
## [1] "R1"
##
## $addr_full
## [1] "17822 NE 95TH CT"
##
## $zip5
## [1] 98052
##
## $ctyname
## [1] NA
##
## $postalctyn
## [1] "REDMOND"
##
## $lon
## [1] -122.1003
##

```

```

## $lat
## [1] 47.68742
##
## $building_grade
## [1] 8
##
## $square_feet_total_living
## [1] 2420
##
## $bedrooms
## [1] 4
##
## $bath_full_count
## [1] 2
##
## $bath_half_count
## [1] 1
##
## $bath_3qtr_count
## [1] 0
##
## $year_built
## [1] 1998
##
## $year_renovated
## [1] 0
##
## $current_zoning
## [1] "R6"
##
## $sq_ft_lot
## [1] 7965
##
## $prop_type
## [1] "R"
##
## $present_use
## [1] 2

# map_int

housingData %>% map_int(NROW)

##           Sale Date           Sale Price           sale_reason
##           12865           12865           12865
##   sale_instrument   sale_warning   sitetype
##           12865           12865           12865
##           addr_full           zip5           ctyname
##           12865           12865           12865
##           postalctyn           lon           lat
##           12865           12865           12865
##   building_grade square_feet_total_living bedrooms
##           12865           12865           12865
##   bath_full_count bath_half_count bath_3qtr_count

```

```

##          12865          12865          12865
##      year_built      year_renovated      current_zoning
##          12865          12865          12865
##      sq_ft_lot      prop_type      present_use
##          12865          12865          12865

# Keep houses with 2 bath

# fullBaths <- housingData$bath_full_count
# keep(fullBaths, grepl("2", fullBaths))

housingTwoBath <- purrr::keep(housingData$bath_full_count, function(x) x >=2)
housingTwoBath

##      [1] 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 3 4 3 2 2 2 2 2 2 2 3 2 2
##      [29] 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 3 2 3 2 2 2 2 2 2 2 3
##      [57] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 3 2 2 2 2
##      [85] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 2 2 2 2 2 2 2 2
##     [113] 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 3 2 2 2 2 4 2 2 2 2 2 2 2 2
##     [141] 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2
##     [169] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 3 2 2 2 2 3 2 2 2 2 2 2
##     [197] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 23 2 2 2 2 3 2 2 2 2 2 2 2 3
##     [225] 2 3 2 2 2 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2
##     [253] 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 3 2
##     [281] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2
##     [309] 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2
##     [337] 3 3 2 2 2 6 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 3 3 2 2 2 2 2
##     [365] 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 2 2 2
##     [393] 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2
##     [421] 2 3 2 2 2 2 2 2 2 2 2 2 2 3 3 2 2 2 2 2 2 2 2 2 2 3 2 2
##     [449] 2 2 2 2 2 3 2 3 3 2 2 2 2 2 2 3 2 2 3 3 2 3 3 2 2 2 2 2
##     [477] 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2
##     [505] 2 2 2 2 2 2 2 2 4 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##     [533] 3 2 2 2 2 2 2 3 4 2 2 3 2 3 3 2 2 2 2 2 2 2 3 2 2 2 2 2
##     [561] 2 2 2 2 2 2 2 2 2 5 2 2 2 2 2 3 2 2 2 3 2 2 2 2 2 2 2 2
##     [589] 2 2 2 2 2 3 2 2 2 2 3 2 3 2 3 2 3 2 2 3 2 2 3 2 2 2 2 2
##     [617] 2 2 2 2 2 2 2 2 2 2 2 2 2 5 2 2 2 2 2 2 2 2 2 2 2 2 2 3
##     [645] 2 2 2 2 2 3 3 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 3 2 2 2
##     [673] 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 3 3 2 2 2 2 3 3 2 2 2 2
##     [701] 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 3 2 2 2 2 2 3 2 2 2
##     [729] 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2
##     [757] 2 2 2 3 3 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2
##     [785] 2 3 2 2 2 3 2 2 2 2 2 3 2 2 2 3 2 5 3 3 2 2 2 2 2 2 2
##     [813] 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##     [841] 2 2 2 2 2 3 3 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##     [869] 2 2 2 2 2 3 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2
##     [897] 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2
##     [925] 2 2 2 2 2 3 2 2 3 2 2 2 2 2 2 2 2 3 2 3 3 2 3 2 2 2
##     [953] 2 2 3 2 2 2 2 2 3 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2
##     [981] 2 2 2 3 2 2 2 2 2 3 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2
##      [ reached getOption("max.print") -- omitted 8042 entries ]

# Use the cbind and rbind function on your dataset
# cbind used to combine columns (divide into halves)

```

```
ncol(housingData)

## [1] 24

housingDataFirstHalf <- housingData[1:12]
housingDataFirstHalf

## # A tibble: 12,865 x 12
##   `Sale Date`      `Sale Price` sale_reason sale_instrument sale_warning sitetype
##   <dtm>           <dbl>      <dbl>         <dbl> <chr>      <chr>
## 1 2006-01-03 00:00:00      698000         1           3 <NA>      R1
## 2 2006-01-03 00:00:00      649990         1           3 <NA>      R1
## 3 2006-01-03 00:00:00      572500         1           3 <NA>      R1
## 4 2006-01-03 00:00:00      420000         1           3 <NA>      R1
## 5 2006-01-03 00:00:00      369900         1           3 15        R1
## 6 2006-01-03 00:00:00      184667         1          15 18 51      R1
## 7 2006-01-04 00:00:00     1050000         1           3 <NA>      R1
## 8 2006-01-04 00:00:00      875000         1           3 <NA>      R1
## 9 2006-01-04 00:00:00      660000         1           3 <NA>      R1
## 10 2006-01-04 00:00:00      650000         1           3 <NA>      R1
## # i 12,855 more rows
## # i 6 more variables: addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>,
## #   lon <dbl>, lat <dbl>

housingDataSecondHalf <- housingData[13:24]
housingDataSecondHalf

## # A tibble: 12,865 x 12
##   building_grade square_feet_total_living bedrooms bath_full_count bath_half_count
##   <dbl>           <dbl>      <dbl>         <dbl>      <dbl>
## 1           9          2810         4           2           1
## 2           9          2880         4           2           0
## 3           8          2770         4           1           1
## 4           8          1620         3           1           0
## 5           7          1440         3           1           0
## 6           7          4160         4           2           1
## 7          10          3960         5           3           0
## 8          10          3720         4           2           1
## 9           9          4160         4           2           1
## 10          8          2760         4           1           0
## # i 12,855 more rows
## # i 7 more variables: bath_3qtr_count <dbl>, year_built <dbl>, year_renovated <dbl>,
## #   current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>

cbind(housingDataFirstHalf, housingDataSecondHalf)

##   Sale Date Sale Price sale_reason sale_instrument sale_warning sitetype
## 1 2006-01-03      698000         1           3      <NA>      R1
## 2 2006-01-03      649990         1           3      <NA>      R1
## 3 2006-01-03      572500         1           3      <NA>      R1
## 4 2006-01-03      420000         1           3      <NA>      R1
## 5 2006-01-03      369900         1           3       15      R1
## 6 2006-01-03      184667         1          15      18 51      R1
## 7 2006-01-04     1050000         1           3      <NA>      R1
## 8 2006-01-04      875000         1           3      <NA>      R1
```

## 9	2006-01-04	660000	1	3	<NA>	R1
## 10	2006-01-04	650000	1	3	<NA>	R1
## 11	2006-01-04	599950	1	3	<NA>	R1
## 12	2006-01-04	526787	1	3	<NA>	R1
## 13	2006-01-04	470000	1	3	<NA>	R1
## 14	2006-01-04	165000	1	3	<NA>	R1
## 15	2006-01-05	803000	1	3	<NA>	R1
## 16	2006-01-05	507950	1	3	<NA>	R1
## 17	2006-01-06	765000	1	3	<NA>	R1
## 18	2006-01-06	589950	1	3	<NA>	R1
## 19	2006-01-09	501000	1	3	<NA>	R1
## 20	2006-01-09	372500	1	3	<NA>	R1
## 21	2006-01-10	513262	1	3	<NA>	R1
## 22	2006-01-10	482000	1	3	<NA>	R1
## 23	2006-01-11	765000	1	3	<NA>	R1
## 24	2006-01-11	372500	1	3	<NA>	R2
## 25	2006-01-11	265000	1	3	<NA>	R1
## 26	2006-01-12	1392000	1	3	<NA>	R1
## 27	2006-01-12	717390	1	3	<NA>	R1
## 28	2006-01-12	552000	1	3	<NA>	R1
## 29	2006-01-12	470000	1	3	<NA>	R1
## 30	2006-01-13	523935	1	3	<NA>	R1
## 31	2006-01-13	399900	1	3	<NA>	R1
## 32	2006-01-13	335105	1	3	<NA>	R1
## 33	2006-01-16	572950	1	3	<NA>	R1
## 34	2006-01-17	949950	1	3	<NA>	R1
## 35	2006-01-17	905000	1	3	41	R1
## 36	2006-01-17	750073	1	3	<NA>	R1
## 37	2006-01-17	526718	1	3	<NA>	R1
## 38	2006-01-18	690749	1	3	<NA>	R1
## 39	2006-01-18	640000	1	3	<NA>	R1
## 40	2006-01-18	485000	1	3	<NA>	R1
## 41	2006-01-18	270000	1	3	<NA>	R1
##		addr_full	zip5	ctyname	postalctyn	lon lat
## 1		17021 NE 113TH CT	98052	REDMOND	REDMOND	-122.1124 47.70139
## 2		11927 178TH PL NE	98052	REDMOND	REDMOND	-122.1022 47.70731
## 3		13315 174TH AVE NE	98052	<NA>	REDMOND	-122.1085 47.71986
## 4		3303 178TH AVE NE	98052	REDMOND	REDMOND	-122.1037 47.63914
## 5		16126 NE 108TH CT	98052	REDMOND	REDMOND	-122.1242 47.69748
## 6		8101 229TH DR NE	98053	<NA>	REDMOND	-122.0341 47.67545
## 7		21634 NE 87TH PL	98053	<NA>	REDMOND	-122.0507 47.68053
## 8		21404 NE 67TH ST	98053	<NA>	REDMOND	-122.0555 47.66510
## 9		7525 238TH AVE NE	98053	<NA>	REDMOND	-122.0227 47.67208
## 10		17703 NE 26TH ST	98052	REDMOND	REDMOND	-122.1039 47.63341
## 11		14924 NE 74TH CT	98052	REDMOND	REDMOND	-122.1411 47.67142
## 12		7858 148TH CT NE	98052	REDMOND	REDMOND	-122.1425 47.67407
## 13		17905 NE 26TH ST	98052	REDMOND	REDMOND	-122.1010 47.63319
## 14		2921 288TH AVE NE	98053	<NA>	REDMOND	-121.9577 47.63382
## 15		3624 264TH AVE NE	98053	<NA>	REDMOND	-121.9857 47.64184
## 16		7850 148TH CT NE	98052	REDMOND	REDMOND	-122.1425 47.67390
## 17		8944 237TH PL NE	98053	<NA>	REDMOND	-122.0230 47.68150
## 18		11922 173RD PL NE	98052	REDMOND	REDMOND	-122.1086 47.70678
## 19		3201 176TH CT NE	98052	REDMOND	REDMOND	-122.1055 47.63872
## 20		26920 NE 50TH ST	98053	<NA>	REDMOND	-121.9795 47.65158



## 21	11807 242ND PL NE 98053	<NA>	REDMOND	-122.0162	47.70323
## 22	9166 226TH PL NE 98053	<NA>	REDMOND	-122.0376	47.68249
## 23	4811 228TH AVE NE 98053	<NA>	REDMOND	-122.0365	47.65149
## 24	8606 134TH CT NE 98052	REDMOND	REDMOND	-122.1606	47.67987
## 25	25149 NE PATTERSON WAY 98053	<NA>	REDMOND	-122.0032	47.65814
## 26	2428 W LAKE SAMMAMISH PKWY NE 98052	REDMOND	REDMOND	-122.0879	47.63205
## 27	16565 NE 119TH WAY 98052	REDMOND	REDMOND	-122.1197	47.70583
## 28	6128 145TH CT NE 98052	REDMOND	REDMOND	-122.1463	47.66187
## 29	14876 NE 78TH WAY 98052	REDMOND	REDMOND	-122.1425	47.67380
## 30	11823 242ND PL NE 98053	<NA>	REDMOND	-122.0160	47.70348
## 31	24307 NE VINE MAPLE WAY 98053	<NA>	REDMOND	-122.0151	47.70263
## 32	2543 174TH AVE NE 98052	REDMOND	REDMOND	-122.1077	47.63326
## 33	17264 NE 119TH WAY 98052	REDMOND	REDMOND	-122.1095	47.70728
## 34	11878 175TH PL NE 98052	REDMOND	REDMOND	-122.1056	47.70671
## 35	24215 NE 59TH LN 98053	<NA>	REDMOND	-122.0145	47.65899
## 36	12240 243RD PL NE 98053	<NA>	REDMOND	-122.0148	47.70771
## 37	23844 NE ADAIR RD 98053	<NA>	REDMOND	-122.0202	47.70903
## 38	12223 243RD PL NE 98053	<NA>	REDMOND	-122.0155	47.70756
## 39	22233 NE 46TH ST 98053	<NA>	REDMOND	-122.0424	47.64993
## 40	17609 NE 141ST ST 98052	<NA>	REDMOND	-122.1040	47.72588
## 41	15918 NE 53RD ST 98052	REDMOND	REDMOND	-122.1274	47.65571
##	building_grade	square_feet_total_living	bedrooms	bath_full_count	bath_half_count
## 1	9	2810	4	2	1
## 2	9	2880	4	2	0
## 3	8	2770	4	1	1
## 4	8	1620	3	1	0
## 5	7	1440	3	1	0
## 6	7	4160	4	2	1
## 7	10	3960	5	3	0
## 8	10	3720	4	2	1
## 9	9	4160	4	2	1
## 10	8	2760	4	1	0
## 11	9	2180	3	2	1
## 12	8	2480	3	2	1
## 13	8	2230	4	1	0
## 14	9	1850	3	2	0
## 15	10	3180	3	2	1
## 16	8	2480	3	2	1
## 17	9	4000	4	2	1
## 18	8	2570	4	2	1
## 19	8	2620	3	1	0
## 20	7	1620	3	1	0
## 21	8	1930	2	2	0
## 22	7	2360	3	2	1
## 23	9	3520	4	3	0
## 24	7	1640	2	1	1
## 25	10	4920	4	4	1
## 26	9	3740	4	3	2
## 27	9	3090	3	2	1
## 28	9	2050	3	2	0
## 29	8	2120	3	2	1
## 30	8	1680	2	2	0
## 31	8	1350	2	2	0
## 32	8	1690	3	2	1

## 33	8		2530	4	2	1
## 34	10		3840	5	3	1
## 35	10		3520	4	2	1
## 36	8		2300	3	2	1
## 37	8		1680	2	2	0
## 38	8		2305	3	2	1
## 39	9		3010	4	2	0
## 40	8		1920	3	2	0
## 41	7		2280	5	2	0
##	bath_3qtr_count	year_built	year_renovated	current_zoning	sq_ft_lot	prop_type
## 1	0	2003	0	R4	6635	R
## 2	1	2006	0	R4	5570	R
## 3	1	1987	0	R6	8444	R
## 4	1	1968	0	R4	9600	R
## 5	1	1980	0	R6	7526	R
## 6	1	2005	0	URPS0	7280	R
## 7	1	1993	0	RA5	97574	R
## 8	0	1988	0	RA5	30649	R
## 9	1	1978	0	RA5	42688	R
## 10	1	1976	0	R3	94889	R
## 11	0	1988	0	R5	7949	R
## 12	0	2005	0	R5	2647	R
## 13	1	1975	0	R3	12070	R
## 14	0	2011	0	RA5	278891	R
## 15	0	1990	0	RA5	95013	R
## 16	0	2005	0	R5	3099	R
## 17	1	2005	0	URPS0	7611	R
## 18	0	2005	0	R4	4737	R
## 19	2	1972	0	R4	9649	R
## 20	1	1977	0	RA5	47480	R
## 21	0	2005	0	URPS0	4958	R
## 22	0	2003	0	URPS0	4080	R
## 23	0	1987	0	RA5	35348	R
## 24	0	1986	0	R5	3279	R
## 25	0	2007	0	RA5	112650	R
## 26	1	1998	0	R4	17291	R
## 27	0	2006	0	R4	5760	R
## 28	0	1986	0	R5	10827	R
## 29	0	2005	0	R5	3962	R
## 30	0	2005	0	URPS0	4764	R
## 31	0	2005	0	URPS0	4781	R
## 32	0	1979	0	R5	2752	R
## 33	0	2006	0	R4	3792	R
## 34	0	2005	0	R4	6204	R
## 35	1	1998	0	RA2.5	46270	R
## 36	0	2006	0	URPS0	5697	R
## 37	0	2005	0	URPS0	6386	R
## 38	0	2006	0	URPS0	7583	R
## 39	1	1986	0	RA5	35006	R
## 40	0	1986	0	R6	7560	R
## 41	1	1966	0	R5	11180	R
##	present_use					
## 1	2					
## 2	2					

```

## 3      2
## 4      2
## 5      2
## 6      2
## 7      2
## 8      2
## 9      2
## 10     2
## 11     2
## 12     2
## 13     2
## 14     2
## 15     2
## 16     2
## 17     2
## 18     2
## 19     2
## 20     2
## 21     2
## 22     2
## 23     2
## 24     2
## 25     2
## 26     2
## 27     2
## 28     2
## 29     2
## 30     2
## 31     29
## 32     29
## 33     2
## 34     2
## 35     2
## 36     2
## 37     2
## 38     2
## 39     2
## 40     2
## 41     2
## [ reached 'max' / getOption("max.print") -- omitted 12824 rows ]

# rbind used to combine rows (first 10 and last 10)

nrow(housingData)

## [1] 12865

housingDataFirst10 <- head(housingData, 10)
housingDataFirst10

## # A tibble: 10 x 24
##   `Sale Date`      `Sale Price` sale_reason sale_instrument sale_warning sitetype
##   <dtm>           <dbl>      <dbl>          <dbl> <chr>      <chr>
## 1 2006-01-03 00:00:00    698000         1            3 <NA>      R1
## 2 2006-01-03 00:00:00    649990         1            3 <NA>      R1

```

```
## 3 2006-01-03 00:00:00      572500      1      3 <NA>      R1
## 4 2006-01-03 00:00:00      420000      1      3 <NA>      R1
## 5 2006-01-03 00:00:00      369900      1      3 15      R1
## 6 2006-01-03 00:00:00      184667      1     15 18 51      R1
## 7 2006-01-04 00:00:00     1050000      1      3 <NA>      R1
## 8 2006-01-04 00:00:00      875000      1      3 <NA>      R1
## 9 2006-01-04 00:00:00      660000      1      3 <NA>      R1
## 10 2006-01-04 00:00:00      650000      1      3 <NA>      R1
## # i 18 more variables: 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>

housingDataLast10 <- tail(housingData, 10)
housingDataLast10

## # A tibble: 10 x 24
##   `Sale Date`      `Sale Price` sale_reason sale_instrument sale_warning sitetype
##   <dtm>          <dbl>      <dbl>          <dbl> <chr>      <chr>
## 1 2016-12-14 00:00:00      180637      1              3 <NA>      R2
## 2 2016-12-15 00:00:00      865000      1              3 <NA>      R1
## 3 2016-12-15 00:00:00      840000      1              3 <NA>      R1
## 4 2016-12-15 00:00:00      833000      1              3 <NA>      R1
## 5 2016-12-15 00:00:00      824000      1              3 <NA>      R1
## 6 2016-12-15 00:00:00      798930      1              3 <NA>      R1
## 7 2016-12-15 00:00:00      750000      1              3 <NA>      R1
## 8 2016-12-15 00:00:00      629000      1              3 <NA>      R1
## 9 2016-12-16 00:00:00      835000      1              3 <NA>      R1
## 10 2016-12-16 00:00:00      455500      1              3 <NA>      R1
## # i 18 more variables: 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>

rbind(housingDataFirst10, housingDataLast10)

## # A tibble: 20 x 24
##   `Sale Date`      `Sale Price` sale_reason sale_instrument sale_warning sitetype
##   <dtm>          <dbl>      <dbl>          <dbl> <chr>      <chr>
## 1 2006-01-03 00:00:00      698000      1              3 <NA>      R1
## 2 2006-01-03 00:00:00      649990      1              3 <NA>      R1
## 3 2006-01-03 00:00:00      572500      1              3 <NA>      R1
## 4 2006-01-03 00:00:00      420000      1              3 <NA>      R1
## 5 2006-01-03 00:00:00      369900      1              3 15      R1
## 6 2006-01-03 00:00:00      184667      1     15 18 51      R1
## 7 2006-01-04 00:00:00     1050000      1              3 <NA>      R1
## 8 2006-01-04 00:00:00      875000      1              3 <NA>      R1
## 9 2006-01-04 00:00:00      660000      1              3 <NA>      R1
## 10 2006-01-04 00:00:00      650000      1              3 <NA>      R1
## 11 2016-12-14 00:00:00      180637      1              3 <NA>      R2
## 12 2016-12-15 00:00:00      865000      1              3 <NA>      R1
## 13 2016-12-15 00:00:00      840000      1              3 <NA>      R1
## 14 2016-12-15 00:00:00      833000      1              3 <NA>      R1
```

```
## 15 2016-12-15 00:00:00      824000      1      3 <NA>      R1
## 16 2016-12-15 00:00:00      798930      1      3 <NA>      R1
## 17 2016-12-15 00:00:00      750000      1      3 <NA>      R1
## 18 2016-12-15 00:00:00      629000      1      3 <NA>      R1
## 19 2016-12-16 00:00:00      835000      1      3 <NA>      R1
## 20 2016-12-16 00:00:00      455500      1      3 <NA>      R1
## # i 18 more variables: addr_full <chr>, zip5 <dbl>, ctynome <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>

# Split a string, then concatenate the results back together
library(stringr)
fullAddrListSplit <- str_split(housingData$addr_full, pattern = " ")
head(fullAddrListSplit)

## [[1]]
## [1] "17021" "NE"      "113TH" "CT"
##
## [[2]]
## [1] "11927" "178TH" "PL"      "NE"
##
## [[3]]
## [1] "13315" "174TH" "AVE"      "NE"
##
## [[4]]
## [1] "3303"  "178TH" "AVE"      "NE"
##
## [[5]]
## [1] "16126" "NE"      "108TH" "CT"
##
## [[6]]
## [1] "8101"  "229TH" "DR"      "NE"

strConcatenate <- paste(fullAddrListSplit[[1]], collapse = " ")
strConcatenate

## [1] "17021 NE 113TH CT"
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 4.3.2 (2023-10-31)
## Platform: x86_64-apple-darwin20 (64-bit)
## Running under: macOS Ventura 13.5.1
##
## Matrix products: default
## BLAS:      /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions/A/
## LAPACK:    /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib; LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
```

```
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics  grDevices utils      datasets  methods   base
##
## other attached packages:
## [1] stringr_1.5.1 purrr_1.0.2  dplyr_1.1.4  readxl_1.4.3
##
## loaded via a namespace (and not attached):
## [1] utf8_1.2.4      R6_2.5.1        xfun_0.41       tidyselect_1.2.0
## [5] cellranger_1.1.0 magrittr_2.0.3   glue_1.6.2      tibble_3.2.1
## [9] knitr_1.45       pkgconfig_2.0.3 generics_0.1.3   lifecycle_1.0.4
## [13] cli_3.6.1        fansi_1.0.5      vctrs_0.6.4     withr_2.5.2
## [17] compiler_4.3.2   highr_0.10       rstudioapi_0.15.0 tools_4.3.2
## [21] evaluate_0.23    pillar_1.9.0     rlang_1.1.2     stringi_1.8.2

Sys.time()

## [1] "2024-01-14 13:50:37 EST"
```