Assignment: ASSIGNMENT 5

Name: Shekhar, Manish

Date: 2021-04-13

```
# Read the housing data set
library(readxl)
mydata <- read_excel("week-6-housing.xlsx")

# check the structure of data and some basic stats
str(mydata)
```

```
## tibble[,24] [12,865 x 24] (S3: tbl_df/tbl/data.frame)
##  $ Sale Date               : POSIXct[1:12865], format: "2006-01-03" "2006-01-03" ...
##  $ Sale Price              : num [1:12865] 698000 649990 572500 420000 369900 ...
##  $ sale_reason             : num [1:12865] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sale_instrument         : num [1:12865] 3 3 3 3 3 15 3 3 3 3 ...
##  $ sale_warning            : chr [1:12865] NA NA NA NA ...
##  $ sitetype                : chr [1:12865] "R1" "R1" "R1" "R1" ...
##  $ addr_full               : chr [1:12865] "17021 NE 113TH CT" "11927 178TH PL NE" "13315 174TH AVE N
##  $ zip5                    : num [1:12865] 98052 98052 98052 98052 98052 ...
##  $ ctyname                 : chr [1:12865] "REDMOND" "REDMOND" NA "REDMOND" ...
##  $ postalctyn              : chr [1:12865] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ lon                     : num [1:12865] -122 -122 -122 -122 -122 ...
##  $ lat                     : num [1:12865] 47.7 47.7 47.7 47.6 47.7 ...
##  $ building_grade          : num [1:12865] 9 9 8 8 7 7 10 10 9 8 ...
##  $ square_feet_total_living: num [1:12865] 2810 2880 2770 1620 1440 4160 3960 3720 4160 2760 ...
##  $ bedrooms                : num [1:12865] 4 4 4 3 3 4 5 4 4 4 ...
##  $ bath_full_count         : num [1:12865] 2 2 1 1 1 2 3 2 2 1 ...
##  $ bath_half_count         : num [1:12865] 1 0 1 0 0 1 0 1 1 0 ...
##  $ bath_3qtr_count         : num [1:12865] 0 1 1 1 1 1 1 0 1 1 ...
##  $ year_built              : num [1:12865] 2003 2006 1987 1968 1980 ...
##  $ year_renovated          : num [1:12865] 0 0 0 0 0 0 0 0 0 0 ...
##  $ current_zoning          : chr [1:12865] "R4" "R4" "R6" "R4" ...
##  $ sq_ft_lot               : num [1:12865] 6635 5570 8444 9600 7526 ...
##  $ prop_type               : chr [1:12865] "R" "R" "R" "R" ...
##  $ present_use             : num [1:12865] 2 2 2 2 2 2 2 2 2 2 ...
```

```
summary(mydata)
```

```
##    Sale Date                     Sale Price       sale_reason
##  Min.   :2006-01-03 00:00:00   Min.   :    698   Min.   : 0.00
##  1st Qu.:2008-07-07 00:00:00   1st Qu.: 460000   1st Qu.: 1.00
##  Median :2011-11-17 00:00:00   Median : 593000   Median : 1.00
##  Mean   :2011-07-28 15:07:32   Mean   : 660738   Mean   : 1.55
##  3rd Qu.:2014-06-05 00:00:00   3rd Qu.: 750000   3rd Qu.: 1.00
##  Max.   :2016-12-16 00:00:00   Max.   :4400000   Max.   :19.00
##  sale_instrument  sale_warning         sitetype          addr_full
##  Min.   : 0.000   Length:12865       Length:12865       Length:12865
##  1st Qu.: 3.000   Class :character   Class :character   Class :character
##  Median : 3.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 3.678
##  3rd Qu.: 3.000
```

```
##  Max.   :27.000
##       zip5         ctyname           postalctyn             lon
##  Min.   :98052   Length:12865       Length:12865       Min.   :-122.2
##  1st Qu.:98052   Class :character   Class :character   1st Qu.:-122.1
##  Median :98052   Mode  :character   Mode  :character   Median :-122.1
##  Mean   :98053                                         Mean   :-122.1
##  3rd Qu.:98053                                         3rd Qu.:-122.0
##  Max.   :98074                                         Max.   :-121.9
##       lat        building_grade  square_feet_total_living    bedrooms
##  Min.   :47.46   Min.   : 2.00   Min.   :  240            Min.   : 0.000
##  1st Qu.:47.67   1st Qu.: 8.00   1st Qu.: 1820            1st Qu.: 3.000
##  Median :47.69   Median : 8.00   Median : 2420            Median : 4.000
##  Mean   :47.68   Mean   : 8.24   Mean   : 2540            Mean   : 3.479
##  3rd Qu.:47.70   3rd Qu.: 9.00   3rd Qu.: 3110            3rd Qu.: 4.000
##  Max.   :47.73   Max.   :13.00   Max.   :13540            Max.   :11.000
##  bath_full_count  bath_half_count  bath_3qtr_count   year_built
##  Min.   : 0.000   Min.   :0.0000   Min.   :0.000   Min.   :1900
##  1st Qu.: 1.000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1979
##  Median : 2.000   Median :1.0000   Median :0.000   Median :1998
##  Mean   : 1.798   Mean   :0.6134   Mean   :0.494   Mean   :1993
##  3rd Qu.: 2.000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:2007
##  Max.   :23.000   Max.   :8.0000   Max.   :8.000   Max.   :2016
##  year_renovated    current_zoning       sq_ft_lot        prop_type
##  Min.   :   0.00   Length:12865       Min.   :    785   Length:12865
##  1st Qu.:   0.00   Class :character   1st Qu.:   5355   Class :character
##  Median :   0.00   Mode  :character   Median :   7965   Mode  :character
##  Mean   :  26.24                      Mean   :  22229
##  3rd Qu.:   0.00                      3rd Qu.:  12632
##  Max.   :2016.00                      Max.   :1631322
##   present_use
##  Min.   :  0.000
##  1st Qu.:  2.000
##  Median :  2.000
##  Mean   :  6.598
##  3rd Qu.:  2.000
##  Max.   :300.000
```

```
head(mydata)
```

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

```r
# Change the column names to remove spaces
colnames(mydata)[1] <- "Sale_Date"
colnames(mydata)[2] <- "Sale_Price"

# 1 a. Using the dplyr package, use the 6 different operations to analyze/transform the data - Group By

# Using ddplyr package's group_by to get mean sale_price by year_built
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
```

```r
mydata %>%
    group_by(year_built) %>%
    summarize("Avg_Sale_Price" = mean(Sale_Price))
```

```
## # A tibble: 109 x 2
##    year_built Avg_Sale_Price
##         <dbl>          <dbl>
##  1       1900        394500.
##  2       1903        430000
##  3       1905        620000
##  4       1906        550000
##  5       1909          1070
##  6       1910        150000
##  7       1912        619667.
##  8       1913        457500
##  9       1914        835000
## 10       1915        228150
## # ... with 99 more rows
```

```r
# Using dplyr package's group_by and summarize to get mean sale_price by no of bedrooms
mydata %>%
    group_by(bedrooms) %>%
    summarize("Avg_Sale_Price" = mean(Sale_Price))
```

```
## # A tibble: 12 x 2
##    bedrooms Avg_Sale_Price
##       <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
```

```
# select square_feet_total_living, sq_ft_lot, bedrooms, bath_full_count, bath_half_count, sale_price
mydata %>%
    select(square_feet_total_living,
           sq_ft_lot,bedrooms,
           bath_full_count,
           bath_half_count,
           Sale_Price)
```

```
## # A tibble: 12,865 x 6
##    square_feet_total_living sq_ft_lot bedrooms bath_full_count bath_half_count
##                       <dbl>     <dbl>    <dbl>           <dbl>           <dbl>
##  1                     2810      6635        4               2               1
##  2                     2880      5570        4               2               0
##  3                     2770      8444        4               1               1
##  4                     1620      9600        3               1               0
##  5                     1440      7526        3               1               0
##  6                     4160      7280        4               2               1
##  7                     3960     97574        5               3               0
##  8                     3720     30649        4               2               1
##  9                     4160     42688        4               2               1
## 10                     2760     94889        4               1               0
## # ... with 12,855 more rows, and 1 more variable: Sale_Price <dbl>
```

```
# select all columns whose names start with 'b'
mydata %>%
    select(starts_with('b'))
```

```
## # A tibble: 12,865 x 5
##    building_grade bedrooms bath_full_count bath_half_count bath_3qtr_count
##             <dbl>    <dbl>           <dbl>           <dbl>           <dbl>
##  1              9        4               2               1               0
##  2              9        4               2               0               1
##  3              8        4               1               1               1
##  4              8        3               1               0               1
##  5              7        3               1               0               1
##  6              7        4               2               1               1
##  7             10        5               3               0               1
##  8             10        4               2               1               0
##  9              9        4               2               1               1
## 10              8        4               1               0               1
## # ... with 12,855 more rows
```

```
# Use mutate() to derive year of sale from sale date and add it to original data frame
# using magrittr package's assignment pipe
library(magrittr)
mydata %<>%
    mutate("year_of_sale"=substr(Sale_Date,1,4))
str(mydata)
```

```
## tibble[,25] [12,865 x 25] (S3: tbl_df/tbl/data.frame)
##  $ Sale_Date               : POSIXct[1:12865], format: "2006-01-03" "2006-01-03" ...
##  $ Sale_Price              : num [1:12865] 698000 649990 572500 420000 369900 ...
```

```
##  $ sale_reason             : num [1:12865] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sale_instrument         : num [1:12865] 3 3 3 3 3 15 3 3 3 3 ...
##  $ sale_warning            : chr [1:12865] NA NA NA NA ...
##  $ sitetype                : chr [1:12865] "R1" "R1" "R1" "R1" ...
##  $ addr_full               : chr [1:12865] "17021 NE 113TH CT" "11927 178TH PL NE" "13315 174TH AVE
##  $ zip5                    : num [1:12865] 98052 98052 98052 98052 98052 ...
##  $ ctyname                 : chr [1:12865] "REDMOND" "REDMOND" NA "REDMOND" ...
##  $ postalctyn              : chr [1:12865] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ lon                     : num [1:12865] -122 -122 -122 -122 -122 ...
##  $ lat                     : num [1:12865] 47.7 47.7 47.7 47.6 47.7 ...
##  $ building_grade          : num [1:12865] 9 9 8 8 7 7 10 10 9 8 ...
##  $ square_feet_total_living: num [1:12865] 2810 2880 2770 1620 1440 4160 3960 3720 4160 2760 ...
##  $ bedrooms                : num [1:12865] 4 4 4 3 3 4 5 4 4 4 ...
##  $ bath_full_count         : num [1:12865] 2 2 1 1 1 2 3 2 2 1 ...
##  $ bath_half_count         : num [1:12865] 1 0 1 0 0 1 0 1 1 0 ...
##  $ bath_3qtr_count         : num [1:12865] 0 1 1 1 1 1 1 0 1 1 ...
##  $ year_built              : num [1:12865] 2003 2006 1987 1968 1980 ...
##  $ year_renovated          : num [1:12865] 0 0 0 0 0 0 0 0 0 0 ...
##  $ current_zoning          : chr [1:12865] "R4" "R4" "R6" "R4" ...
##  $ sq_ft_lot               : num [1:12865] 6635 5570 8444 9600 7526 ...
##  $ prop_type               : chr [1:12865] "R" "R" "R" "R" ...
##  $ present_use             : num [1:12865] 2 2 2 2 2 2 2 2 2 2 ...
##  $ year_of_sale            : chr [1:12865] "2006" "2006" "2006" "2006" ...
```

```
# use filter() to filter the data
# filter houses built in or after year 2000 and count them
mydata %>%
    filter(year_built >= 2000) %>% NROW()
```

```
## [1] 6321
```

```
# filter houses built before year 2000 and count them
mydata %>%
    filter(year_built < 2000) %>% NROW()
```

```
## [1] 6544
```

```
# get total sale price by year of sale and then order / arrange it in descending order of total sale
mydata %>%
    group_by(year_of_sale) %>%
    summarize("Total_Sale"=sum(Sale_Price)) %>%
    arrange(desc(Total_Sale))
```

```
## # A tibble: 11 x 2
##    year_of_sale Total_Sale
##    <chr>             <dbl>
##  1 2016          919598273
##  2 2006          919005546
##  3 2015          915474082
##  4 2013          860712529
##  5 2007          850285247
##  6 2014          792182425
##  7 2008          755045696
##  8 2012          747585171
##  9 2011          693256612
## 10 2010          592828305
```

```
## 11 2009          454417263
```

```
# 1 b. Using the purrr package - perform 2 functions on your dataset.  You could use zip_n, keep, disca

library(purrr)
```

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

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

```
# Use keep() to keep all the houses renovated in and after year 2000
reno_after_2000 <- keep(mydata$year_renovated, ~ .x >= 2000)
str(reno_after_2000)
```

```
##  num [1:85] 2004 2004 2006 2002 2000 ...
```

```
# Use discard() to discard all the houses built before 1990
built_before_1990 <- discard(mydata$year_built, ~ .x < 1990)
str(built_before_1990)
```

```
##  num [1:7613] 2003 2006 2005 1993 2005 ...
```

```
# using negate() to check which elements of the list are not NA
is_not_na <- negate(is.na)
sl_wrng_non_na <- map_lgl(mydata$sale_warning, is_not_na)
str(sl_wrng_non_na)
```

```
##  logi [1:12865] FALSE FALSE FALSE FALSE TRUE TRUE ...
```

```
# using partial() to create modified mean function with na_rm = TRUE in-built
# later use the new function to calculate mean without worrying about NAs
my_mean = partial(mean, na.rm = TRUE)
# using new mean function to get average selling price of the houses
my_mean(mydata$Sale_Price)
```

```
## [1] 660737.7
```

```
# using possibly() to handle errors
my_str_concat <- function(x,y){
    str_c(x,y,sep = ",")
}
# create a modified string concatenation function using possibly
my_str_concat_m <- possibly(my_str_concat, otherwise = "I am not valid string")
# use the function to create city_state
city_state <- my_str_concat_m(mydata$ctyname, "WA")
str(city_state)
```

```
##  chr "I am not valid string"
```

```
# 1 c.Use the cbind and rbind function on your dataset

# Using cbind to add sale warning indicator
my_housing_data <- cbind(mydata, "Sale_Warning_in" = !(is.na(mydata$sale_warning)))
str(my_housing_data)
```

```
## 'data.frame':    12865 obs. of  26 variables:
##  $ Sale_Date              : POSIXct, format: "2006-01-03" "2006-01-03" ...
```

```
##  $ Sale_Price              : num  698000 649990 572500 420000 369900 ...
##  $ 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" ...
##  $ addr_full               : chr  "17021 NE 113TH CT" "11927 178TH PL NE" "13315 174TH AVE NE" "3303
##  $ zip5                    : num  98052 98052 98052 98052 98052 ...
##  $ ctyname                 : chr  "REDMOND" "REDMOND" NA "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          : num  9 9 8 8 7 7 10 10 9 8 ...
##  $ square_feet_total_living: num  2810 2880 2770 1620 1440 4160 3960 3720 4160 2760 ...
##  $ bedrooms                : num  4 4 4 3 3 4 5 4 4 4 ...
##  $ bath_full_count         : num  2 2 1 1 1 2 3 2 2 1 ...
##  $ bath_half_count         : num  1 0 1 0 0 1 0 1 1 0 ...
##  $ bath_3qtr_count         : num  0 1 1 1 1 1 1 0 1 1 ...
##  $ year_built              : num  2003 2006 1987 1968 1980 ...
##  $ year_renovated          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ current_zoning          : chr  "R4" "R4" "R6" "R4" ...
##  $ sq_ft_lot               : num  6635 5570 8444 9600 7526 ...
##  $ prop_type               : chr  "R" "R" "R" "R" ...
##  $ present_use             : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ year_of_sale            : chr  "2006" "2006" "2006" "2006" ...
##  $ Sale_Warning_in         : logi  FALSE FALSE FALSE FALSE TRUE TRUE ...
```

```r
# split data into two using year_built > 1990 into two
hs_data_before_1990 <- mydata %>% filter(year_built < 1990)
str(hs_data_before_1990)
```

```
## tibble[,25] [5,252 x 25] (S3: tbl_df/tbl/data.frame)
##  $ Sale_Date               : POSIXct[1:5252], format: "2006-01-03" "2006-01-03" ...
##  $ Sale_Price              : num [1:5252] 572500 420000 369900 875000 660000 ...
##  $ sale_reason             : num [1:5252] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sale_instrument         : num [1:5252] 3 3 3 3 3 3 3 3 3 3 ...
##  $ sale_warning            : chr [1:5252] NA NA "15" NA ...
##  $ sitetype                : chr [1:5252] "R1" "R1" "R1" "R1" ...
##  $ addr_full               : chr [1:5252] "13315 174TH AVE NE" "3303 178TH AVE NE" "16126 NE 108TH C
##  $ zip5                    : num [1:5252] 98052 98052 98052 98053 98053 ...
##  $ ctyname                 : chr [1:5252] NA "REDMOND" "REDMOND" NA ...
##  $ postalctyn              : chr [1:5252] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ lon                     : num [1:5252] -122 -122 -122 -122 -122 ...
##  $ lat                     : num [1:5252] 47.7 47.6 47.7 47.7 47.7 ...
##  $ building_grade          : num [1:5252] 8 8 7 10 9 8 9 8 8 7 ...
##  $ square_feet_total_living: num [1:5252] 2770 1620 1440 3720 4160 2760 2180 2230 2620 1620 ...
##  $ bedrooms                : num [1:5252] 4 3 3 4 4 4 3 4 3 3 ...
##  $ bath_full_count         : num [1:5252] 1 1 1 2 2 1 2 1 1 1 ...
##  $ bath_half_count         : num [1:5252] 1 0 0 1 1 0 1 0 0 0 ...
##  $ bath_3qtr_count         : num [1:5252] 1 1 1 0 1 1 0 1 2 1 ...
##  $ year_built              : num [1:5252] 1987 1968 1980 1988 1978 ...
##  $ year_renovated          : num [1:5252] 0 0 0 0 0 0 0 0 0 0 ...
##  $ current_zoning          : chr [1:5252] "R6" "R4" "R6" "RA5" ...
##  $ sq_ft_lot               : num [1:5252] 8444 9600 7526 30649 42688 ...
##  $ prop_type               : chr [1:5252] "R" "R" "R" "R" ...
##  $ present_use             : num [1:5252] 2 2 2 2 2 2 2 2 2 2 ...
```

```
##  $ year_of_sale            : chr [1:5252] "2006" "2006" "2006" "2006" ...
```

```
hs_data_after_1990 <- mydata %>% filter(year_built >= 1990)
str(hs_data_after_1990)
```

```
## tibble[,25] [7,613 x 25] (S3: tbl_df/tbl/data.frame)
##  $ Sale_Date               : POSIXct[1:7613], format: "2006-01-03" "2006-01-03" ...
##  $ Sale_Price              : num [1:7613] 698000 649990 184667 1050000 526787 ...
##  $ sale_reason             : num [1:7613] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sale_instrument         : num [1:7613] 3 3 15 3 3 3 3 3 3 3 ...
##  $ sale_warning            : chr [1:7613] NA NA "18 51" NA ...
##  $ sitetype                : chr [1:7613] "R1" "R1" "R1" "R1" ...
##  $ addr_full               : chr [1:7613] "17021 NE 113TH CT" "11927 178TH PL NE" "8101 229TH DR NE"
##  $ zip5                    : num [1:7613] 98052 98052 98053 98053 98052 ...
##  $ ctyname                 : chr [1:7613] "REDMOND" "REDMOND" NA NA ...
##  $ postalctyn              : chr [1:7613] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ lon                     : num [1:7613] -122 -122 -122 -122 -122 ...
##  $ lat                     : num [1:7613] 47.7 47.7 47.7 47.7 47.7 ...
##  $ building_grade          : num [1:7613] 9 9 7 10 8 9 10 8 9 8 ...
##  $ square_feet_total_living: num [1:7613] 2810 2880 4160 3960 2480 1850 3180 2480 4000 2570 ...
##  $ bedrooms                : num [1:7613] 4 4 4 5 3 3 3 3 4 4 ...
##  $ bath_full_count         : num [1:7613] 2 2 2 3 2 2 2 2 2 2 ...
##  $ bath_half_count         : num [1:7613] 1 0 1 0 1 0 1 1 1 1 ...
##  $ bath_3qtr_count         : num [1:7613] 0 1 1 1 0 0 0 0 1 0 ...
##  $ year_built              : num [1:7613] 2003 2006 2005 1993 2005 ...
##  $ year_renovated          : num [1:7613] 0 0 0 0 0 0 0 0 0 0 ...
##  $ current_zoning          : chr [1:7613] "R4" "R4" "URPSO" "RA5" ...
##  $ sq_ft_lot               : num [1:7613] 6635 5570 7280 97574 2647 ...
##  $ prop_type               : chr [1:7613] "R" "R" "R" "R" ...
##  $ present_use             : num [1:7613] 2 2 2 2 2 2 2 2 2 2 ...
##  $ year_of_sale            : chr [1:7613] "2006" "2006" "2006" "2006" ...
```

```
# using rbind to add new housing record
housing_data <- rbind(hs_data_before_1990, hs_data_after_1990)
str(housing_data)
```

```
## tibble[,25] [12,865 x 25] (S3: tbl_df/tbl/data.frame)
##  $ Sale_Date               : POSIXct[1:12865], format: "2006-01-03" "2006-01-03" ...
##  $ Sale_Price              : num [1:12865] 572500 420000 369900 875000 660000 ...
##  $ sale_reason             : num [1:12865] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sale_instrument         : num [1:12865] 3 3 3 3 3 3 3 3 3 3 ...
##  $ sale_warning            : chr [1:12865] NA NA "15" NA ...
##  $ sitetype                : chr [1:12865] "R1" "R1" "R1" "R1" ...
##  $ addr_full               : chr [1:12865] "13315 174TH AVE NE" "3303 178TH AVE NE" "16126 NE 108TH C
##  $ zip5                    : num [1:12865] 98052 98052 98052 98053 98053 ...
##  $ ctyname                 : chr [1:12865] NA "REDMOND" "REDMOND" NA ...
##  $ postalctyn              : chr [1:12865] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ lon                     : num [1:12865] -122 -122 -122 -122 -122 ...
##  $ lat                     : num [1:12865] 47.7 47.6 47.7 47.7 47.7 ...
##  $ building_grade          : num [1:12865] 8 8 7 10 9 8 9 8 8 7 ...
##  $ square_feet_total_living: num [1:12865] 2770 1620 1440 3720 4160 2760 2180 2230 2620 1620 ...
##  $ bedrooms                : num [1:12865] 4 3 3 4 4 4 3 4 3 3 ...
##  $ bath_full_count         : num [1:12865] 1 1 1 2 2 1 2 1 1 1 ...
##  $ bath_half_count         : num [1:12865] 1 0 0 1 1 0 1 0 0 0 ...
##  $ bath_3qtr_count         : num [1:12865] 1 1 1 0 1 1 0 1 2 1 ...
```

```
##  $ year_built              : num [1:12865] 1987 1968 1980 1988 1978 ...
##  $ year_renovated          : num [1:12865] 0 0 0 0 0 0 0 0 0 0 ...
##  $ current_zoning          : chr [1:12865] "R6" "R4" "R6" "RA5" ...
##  $ sq_ft_lot               : num [1:12865] 8444 9600 7526 30649 42688 ...
##  $ prop_type               : chr [1:12865] "R" "R" "R" "R" ...
##  $ present_use             : num [1:12865] 2 2 2 2 2 2 2 2 2 2 ...
##  $ year_of_sale            : chr [1:12865] "2006" "2006" "2006" "2006" ...
```

```
# Split a string, then concatenate the results back together
# For this exercise I have created another data frame from mtcars
# I have added a column cars which is basically row names of each row
mtcars2 <- mtcars
mtcars2["cars"] <- c(rownames(mtcars))
# split cars variable by space in the mtcars2 data frame to get the name of make of the car
library(stringr)
cars_split_list <- str_split(string = mtcars2$cars, pattern = " ")
# get unique car models
cars_make <- sapply(cars_split_list, FUN = function(x) x[1])
cars_make
```

```
##  [1] "Mazda"    "Mazda"    "Datsun"   "Hornet"   "Hornet"   "Valiant"
##  [7] "Duster"   "Merc"     "Merc"     "Merc"     "Merc"     "Merc"
## [13] "Merc"     "Merc"     "Cadillac" "Lincoln"  "Chrysler" "Fiat"
## [19] "Honda"    "Toyota"   "Toyota"   "Dodge"    "AMC"      "Camaro"
## [25] "Pontiac"  "Fiat"     "Porsche"  "Lotus"    "Ford"     "Ferrari"
## [31] "Maserati" "Volvo"
```

```
cars_model <- sapply(cars_split_list, FUN = function(x) x[2])
cars_model
```

```
##  [1] "RX4"         "RX4"         "710"         "4"           "Sportabout"
##  [6] NA            "360"         "240D"        "230"         "280"
## [11] "280C"        "450SE"       "450SL"       "450SLC"      "Fleetwood"
## [16] "Continental" "Imperial"    "128"         "Civic"       "Corolla"
## [21] "Corona"      "Challenger"  "Javelin"     "Z28"         "Firebird"
## [26] "X1-9"        "914-2"       "Europa"      "Pantera"     "Dino"
## [31] "Bora"        "142E"
```

```
cars_submodel <- sapply(cars_split_list, FUN = function(x) x[3])
cars_submodel
```

```
##  [1] NA      "Wag"   NA      "Drive" NA      NA      NA      NA      NA
## [10] NA      NA      NA      NA      NA      NA      NA      NA      NA
## [19] NA      NA      NA      NA      NA      NA      NA      NA      NA
## [28] NA      "L"     NA      NA      NA
```

```
# combining strings together
# replacing all NAs in each list of strings with blank so they can be combined
# where values are missing.
cars2 <- str_c(str_replace_na(cars_make,""),
               str_replace_na(cars_model,""),
               str_replace_na(cars_submodel,""), sep = " ")
cars2
```

```
##  [1] "Mazda RX4 "           "Mazda RX4 Wag"        "Datsun 710 "
##  [4] "Hornet 4 Drive"       "Hornet Sportabout "   "Valiant  "
##  [7] "Duster 360 "          "Merc 240D "           "Merc 230 "
```

```
## [10] "Merc 280 "            "Merc 280C "           "Merc 450SE "
## [13] "Merc 450SL "          "Merc 450SLC "         "Cadillac Fleetwood "
## [16] "Lincoln Continental " "Chrysler Imperial "   "Fiat 128 "
## [19] "Honda Civic "         "Toyota Corolla "      "Toyota Corona "
## [22] "Dodge Challenger "    "AMC Javelin "         "Camaro Z28 "
## [25] "Pontiac Firebird "    "Fiat X1-9 "           "Porsche 914-2 "
## [28] "Lotus Europa "        "Ford Pantera L"       "Ferrari Dino "
## [31] "Maserati Bora "       "Volvo 142E "
```