Assignment 5

Surenther

2024-09-28

Load XLSX file

```
#Load XLSX
library(readxl)
mydata <- read_excel("week-6-housing.xlsx", 1)
```

Dplyr Functions

Mutate - Create Year column

```
#Extract the Year column from the Date and Create a new column
library(magrittr) #Library for Pipes
library(dplyr,warn.conflicts = FALSE)
mydata2 <- mydata
mydata2 %<>%
  select('Sale Date','Sale Price','zip5','square_feet_total_living','bedrooms') %>%
  mutate('Sale Year' = format(mydata$'Sale Date','%Y'))
print(mydata2,width=Inf,n=5)
```

```
## # A tibble: 12,865 x 6
##   ‘Sale Date‘         ‘Sale Price‘  zip5 square_feet_total_living bedrooms
##   <dttm>                     <dbl> <dbl>                    <dbl>    <dbl>
## 1 2006-01-03 00:00:00       698000 98052                     2810        4
## 2 2006-01-03 00:00:00       649990 98052                     2880        4
## 3 2006-01-03 00:00:00       572500 98052                     2770        4
## 4 2006-01-03 00:00:00       420000 98052                     1620        3
## 5 2006-01-03 00:00:00       369900 98052                     1440        3
##   ‘Sale Year‘
##   <chr>
## 1 2006
## 2 2006
## 3 2006
## 4 2006
## 5 2006
## # i 12,860 more rows
```

Summarize - Avg Sale Price, Avg Sqft, Avg Bedroom

```r
#Calculate Avg Saleprice, SQFT and Bedroom
mydata2 %>%
  summarize_at(
   vars(AvgSalePrice="Sale Price",AvgSqft="square_feet_total_living",
        AvgBedroom="bedrooms"),mean, na.rm = TRUE
   )
```

```
## # A tibble: 1 x 3
##   AvgSalePrice AvgSqft AvgBedroom
##          <dbl>   <dbl>      <dbl>
## 1      660738.   2540.       3.48
```

Group by - Yearsold, Zip

```r
#Group by Year,Zip & Calculate Avg Saleprice, SQFT and Bedroom
mydata2 %>%
  group_by(
    Year = mydata2$'Sale Year',Zip = mydata2$'zip5'
    ) %>%
  summarize_at(
    vars(AvgSalePrice="Sale Price",AvgSqft="square_feet_total_living",
         AvgBedroom="bedrooms"),mean, na.rm = TRUE
    )
```

```
## # A tibble: 34 x 5
## # Groups:   Year [11]
##    Year    Zip AvgSalePrice AvgSqft AvgBedroom
##    <chr> <dbl>        <dbl>   <dbl>      <dbl>
##  1 2006  98052      607307.   2491.       3.70
##  2 2006  98053      638009.   2564.       3.06
##  3 2006  98074     1233529.   4617.       4.57
##  4 2007  98052      687686.   2495.       3.66
##  5 2007  98053      639144.   2380.       2.93
##  6 2007  98074      823808.   3419.       4
##  7 2008  98052      629368.   2563.       3.70
##  8 2008  98053     1048070.   2337.       3.06
##  9 2008  98074      673750    2768.       3.75
## 10 2009  98052      545385.   2474.       3.59
## # i 24 more rows
```

Select - Sale Date, Sale Price, Zip & Square Footage

```r
#Select Sale data, Sale Price, Zip and Square feet
mydata %>%
  select('Sale Date','Sale Price','zip5','square_feet_total_living')
```

```
## # A tibble: 12,865 x 4
##    ‘Sale Date‘         ‘Sale Price‘  zip5 square_feet_total_living
##    <dttm>                     <dbl> <dbl>                    <dbl>
##  1 2006-01-03 00:00:00       698000 98052                     2810
##  2 2006-01-03 00:00:00       649990 98052                     2880
##  3 2006-01-03 00:00:00       572500 98052                     2770
##  4 2006-01-03 00:00:00       420000 98052                     1620
##  5 2006-01-03 00:00:00       369900 98052                     1440
##  6 2006-01-03 00:00:00       184667 98053                     4160
##  7 2006-01-04 00:00:00      1050000 98053                     3960
##  8 2006-01-04 00:00:00       875000 98053                     3720
##  9 2006-01-04 00:00:00       660000 98053                     4160
## 10 2006-01-04 00:00:00       650000 98052                     2760
## # i 12,855 more rows
```

Filter - based on Sale Year and Price and Sqft

```
#Find home with more than 3500sqft with less than 650k sold at 2016
mydata2 %>%
  print(filter(mydata2$'Sale Year' == "2016" & mydata2$'Sale Price' <= 650000
               & mydata2$'square_feet_total_living' > 3500),width=Inf,n=5)
```

```
## # A tibble: 12,865 x 6
##   ‘Sale Date‘         ‘Sale Price‘  zip5 square_feet_total_living bedrooms
##   <dttm>                     <dbl> <dbl>                    <dbl>    <dbl>
## 1 2006-01-03 00:00:00       698000 98052                     2810        4
## 2 2006-01-03 00:00:00       649990 98052                     2880        4
## 3 2006-01-03 00:00:00       572500 98052                     2770        4
## 4 2006-01-03 00:00:00       420000 98052                     1620        3
## 5 2006-01-03 00:00:00       369900 98052                     1440        3
##   ‘Sale Year‘
##   <chr>
## 1 2006
## 2 2006
## 3 2006
## 4 2006
## 5 2006
## # i 12,860 more rows
```

Arrange - by Avg Saleprice in desc

```
#Group by Year,Zip & Calculate Avg Saleprice, SQFT and Bedroom, Display by Avg Saleprice
mydata2 %>%
  group_by(
    Year = mydata2$'Sale Year',Zip = mydata2$'zip5'
    ) %>%
  summarize_at(
    vars(AvgSalePrice="Sale Price",AvgSqft="square_feet_total_living",
         AvgBedroom="bedrooms"),mean, na.rm = TRUE
    ) %>%
  arrange(desc(AvgSalePrice))
```

```
## # A tibble: 34 x 5
## # Groups:   Year [11]
##    Year    Zip AvgSalePrice AvgSqft AvgBedroom
##    <chr> <dbl>        <dbl>   <dbl>      <dbl>
##  1 2006  98074     1233529.   4617.       4.57
##  2 2012  98074     1171000    4238.       3.83
##  3 2013  98074     1127200    4102.       4.17
##  4 2008  98053     1048070.   2337.       3.06
##  5 2010  98074     1042000    4046        3.8
##  6 2011  98074     1024280    4386        4.2
##  7 2015  98074      964450    3550        3.6
##  8 2007  98074      823808.   3419.       4
##  9 2014  98074      823400    3644.       4.29
## 10 2016  98053      794810.   2560.       3.12
## # i 24 more rows
```

purrr Functions

keep - Only Numeric Columns

```
library(purrr,warn.conflicts = FALSE)
numeric_data <- keep(mydata, is.numeric)
numeric_data
```

```
## # A tibble: 12,865 x 16
##    ‘Sale Price‘ sale_reason sale_instrument  zip5   lon   lat building_grade
##           <dbl>       <dbl>           <dbl> <dbl> <dbl> <dbl>          <dbl>
##  1       698000           1               3 98052 -122.  47.7              9
##  2       649990           1               3 98052 -122.  47.7              9
##  3       572500           1               3 98052 -122.  47.7              8
##  4       420000           1               3 98052 -122.  47.6              8
##  5       369900           1               3 98052 -122.  47.7              7
##  6       184667           1              15 98053 -122.  47.7              7
##  7      1050000           1               3 98053 -122.  47.7             10
##  8       875000           1               3 98053 -122.  47.7             10
##  9       660000           1               3 98053 -122.  47.7              9
## 10       650000           1               3 98052 -122.  47.6              8
## # i 12,855 more rows
## # i 9 more variables: 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>, sq_ft_lot <dbl>, present_use <dbl>
```

Findings: It removed non numeric columns like Sale Date,addr_full,ctyname etc

discard - Remove Colunmn with NA

```
cleaned_data <- discard(mydata, ~ any(is.na(.)))
cleaned_data
```

```
## # A tibble: 12,865 x 22
##    `Sale Date`         `Sale Price` sale_reason sale_instrument sitetype
##    <dttm>                     <dbl>       <dbl>           <dbl> <chr>
##  1 2006-01-03 00:00:00       698000           1               3 R1
##  2 2006-01-03 00:00:00       649990           1               3 R1
##  3 2006-01-03 00:00:00       572500           1               3 R1
##  4 2006-01-03 00:00:00       420000           1               3 R1
##  5 2006-01-03 00:00:00       369900           1               3 R1
##  6 2006-01-03 00:00:00       184667           1              15 R1
##  7 2006-01-04 00:00:00      1050000           1               3 R1
##  8 2006-01-04 00:00:00       875000           1               3 R1
##  9 2006-01-04 00:00:00       660000           1               3 R1
## 10 2006-01-04 00:00:00       650000           1               3 R1
## # i 12,855 more rows
## # i 17 more variables: addr_full <chr>, zip5 <dbl>, 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>
```

Findings: It removed columns like sale_warning,ctyname which is having missing value

compact - Remove NULL values from the data

```
clean_without_null <- compact(mydata)
clean_without_null
```

```
## # 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>
## # i 12,855 more rows
## # i 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>
```

Cbind - Column Bind

```r
# Extract Home details from the data
home_details <-
  mydata %>%
  select(year_built,square_feet_total_living,bedrooms)
# Extract Sale details from the data
sale_detail <-
  mydata %>%
  select('Sale Date','Sale Price',sale_reason)
#Cbind
c_merge <- cbind(home_details,sale_detail)
head(c_merge,n=5)
```

```
##   year_built square_feet_total_living bedrooms  Sale Date Sale Price
## 1       2003                     2810        4 2006-01-03     698000
## 2       2006                     2880        4 2006-01-03     649990
## 3       1987                     2770        4 2006-01-03     572500
## 4       1968                     1620        3 2006-01-03     420000
## 5       1980                     1440        3 2006-01-03     369900
##   sale_reason
## 1           1
## 2           1
## 3           1
## 4           1
## 5           1
```

Rbind - Row Bind

```r
#2006 Sale details
sale_2006 <-
  mydata2 %>%
  filter(mydata2$'Sale Year' == "2006")
#2016 Sale details
sale_2016 <-
  mydata2 %>%
  filter(mydata2$'Sale Year' == "2016")
#Rbind
r_merge <- rbind(sale_2006,sale_2016)
head(r_merge,n=5)
```

```
## # A tibble: 5 x 6
##   ‘Sale Date‘         ‘Sale Price‘  zip5 square_feet_total_living bedrooms
##   <dttm>                     <dbl> <dbl>                    <dbl>    <dbl>
## 1 2006-01-03 00:00:00       698000 98052                     2810        4
## 2 2006-01-03 00:00:00       649990 98052                     2880        4
## 3 2006-01-03 00:00:00       572500 98052                     2770        4
## 4 2006-01-03 00:00:00       420000 98052                     1620        3
## 5 2006-01-03 00:00:00       369900 98052                     1440        3
## # i 1 more variable: ‘Sale Year‘ <chr>
```

Split- Extract house no & Street name from address

```r
library(stringr)
#Split the full address based on first space
mydata[c('Home_No', 'Street Name')]  <- str_split_fixed(string=mydata$addr_full,
                                                       pattern=" ",2)
#display the data
mydata %>% select(addr_full,Home_No,'Street Name')
```

```
## # A tibble: 12,865 x 3
##    addr_full          Home_No ‘Street Name‘
##    <chr>              <chr>   <chr>
##  1 17021 NE 113TH CT  17021   NE 113TH CT
##  2 11927 178TH PL NE  11927   178TH PL NE
##  3 13315 174TH AVE NE 13315   174TH AVE NE
##  4 3303 178TH AVE NE  3303    178TH AVE NE
##  5 16126 NE 108TH CT  16126   NE 108TH CT
##  6 8101 229TH DR NE   8101    229TH DR NE
##  7 21634 NE 87TH PL   21634   NE 87TH PL
##  8 21404 NE 67TH ST   21404   NE 67TH ST
##  9 7525 238TH AVE NE  7525    238TH AVE NE
## 10 17703 NE 26TH ST   17703   NE 26TH ST
## # i 12,855 more rows
```

concatenate - address with proper format

```r
mydata %>%
  mutate('modified_addr' = sprintf("%s,%s %s-%s",mydata$Home_No,mydata$'Street Name',
                                   mydata$postalctyn,mydata$zip5)) %>%
  select(addr_full,modified_addr)
```

```
## # A tibble: 12,865 x 2
##    addr_full          modified_addr
##    <chr>              <chr>
##  1 17021 NE 113TH CT  17021,NE 113TH CT REDMOND-98052
##  2 11927 178TH PL NE  11927,178TH PL NE REDMOND-98052
##  3 13315 174TH AVE NE 13315,174TH AVE NE REDMOND-98052
##  4 3303 178TH AVE NE  3303,178TH AVE NE REDMOND-98052
##  5 16126 NE 108TH CT  16126,NE 108TH CT REDMOND-98052
##  6 8101 229TH DR NE   8101,229TH DR NE REDMOND-98053
##  7 21634 NE 87TH PL   21634,NE 87TH PL REDMOND-98053
##  8 21404 NE 67TH ST   21404,NE 67TH ST REDMOND-98053
##  9 7525 238TH AVE NE  7525,238TH AVE NE REDMOND-98053
## 10 17703 NE 26TH ST   17703,NE 26TH ST REDMOND-98052
## # i 12,855 more rows
```