# Assignment: 5.2 Exercise Housing dataset

# Name: Koby-Hercsky, Theodore

# Date: 2021-04-13

1. 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
   1. **# 1. GroupBy** – I used the groupby function below to group each zip code and tally the number of houses that are in each. As seen below the largest number of houses in our data set is coming from 98052 with 7,452 houses while zip code 98059 only has one house in our data set.
      1. **Updated\_Housing\_Dataset %>% group\_by(zip5) %>% tally()**
      2. **Text

         Description automatically generated with medium confidence**
   2. **# 2. Summarize** – I used the summarize function to find the mean price of each zip code in our housing data set. The one with the highest mean price is zipcode 98074 as it is for $ 951,544.00 while it only has 73 houses in this data set as seen in our group by function.

**Table

Description automatically generated with medium confidence**

* + 1. **Updated\_Housing\_Dataset %>% group\_by(zip5) %>% summarize(Mean\_Price = mean(Sale.Price, na.rm = TRUE))**
  1. **# 3. Mutate** – Used the mutate function seen below to add another column that is for the price per square foot of living space per house in our data set. This gives an idea of how much each house is in comparison to how big the actual house is. So, if the house is small but the price per square foot is high it must be in a nice area and or be a little smaller but very nice.
     1. **Price\_Per\_square\_feet\_housingdata <- mutate(Updated\_Housing\_Dataset, price\_per\_square\_foot = Sale.Price / square\_feet\_total\_living)**

**Table

Description automatically generated**

* 1. **# 4. Filter** – Seen below is my code using the Filter function that has filter houses that are located in the zip code 98053. As seen in the screen shot below, we have narrowed down our rows to 5,339 from 12,865 in our original table. This shows that the number of houses in in the 98053 zip code is almost half the houses in this data set.
     1. **zipcode\_Housing\_Data <- filter(Updated\_Housing\_Dataset, zip5 == 98053)**

**Table

Description automatically generated**

* 1. **# 5. Select** – Seen below is my code for the select function from dplyr. I selected columns that I believe is needed to do further research on the housing data such as the price of sales, the zip code the house is located in, how large the living area is, and also the amount of bed and baths that are in the house.
     1. Updated\_Housing\_Dataset -> select(Housing\_dataset, Sale.Price, zip5, square\_feet\_total\_living, bedrooms, bath\_full\_count, bath\_half\_count, sq\_ft\_lot)

Table

Description automatically generated

* 1. **# 6. Arrange** – I used the Arrange function to sort the price of each house from highest to lowest. This is useful to see houses in order of how much they cost. As seen below from our data set, we see the highest costing house in our data set to be $ 4,400,000.00.
     1. **Housing\_Data\_By\_Price <- Updated\_Housing\_Dataset %>% arrange(desc(Sale.Price), .by\_group = FALSE)**

**Table

Description automatically generated**

1. Using the purrr package – perform 2 functions on your dataset. You could use zip\_n, keep, discard, compact, etc. This will be used when I use my Cbind in the following tasks.
   1. **Keep** – Used the Keep function to keep each line with a mean over 600 which got rid of my bedroom and bathroom columns
      1. Mean\_Keep\_Housing\_data\_set <- Updated\_Housing\_Dataset %>% keep(~ mean(.x) > 600)

Table

Description automatically generated

* 1. **Discard** – I used the discard function to get rid of all the columns except the ones that pertained to the bedrooms and bathrooms. This will be used when I use my Cbind in the following tasks.
     1. Discard\_Housingdata <- Updated\_Housing\_Dataset %>% discard(~ sum(.x) > 100000
     2. Table

        Description automatically generated

1. Use the cbind and rbind function on your dataset
   1. Cbind – I used Cbind to combine what I kept and discarded when I used the purrr package with keep and discard. As seen below I combined the zipcode, price, bed, and bath to bring my data set back together.
      1. Cbind\_Housing <- cbind(zipcode\_house, Price\_Housing\_Dataset, Discard\_Housingdata)

Table

Description automatically generated

* 1. Rbind – I have used Rbind to combine two data sets I created from the filter function that I created for zip code 98053 and 98052. This combined the rows for both zipcodes back together.
     1. rbind\_ZipHousing <- rbind(zipcode\_98052\_Housing\_Data, zipcode\_98053\_Housing\_Data)

Table

Description automatically generated

1. Split a string, then concatenate the results back together
   1. Split the date the house was sold by using the str\_split by using the code below and seen the output in the screen shot.
   2. Year\_sold <- str\_split(string = Housing\_data$Sale.Date, pattern = "-")
   3. Table

      Description automatically generated