Assignment 4

Predictive Analytics

Meruyert Balgabek

MET AD 571

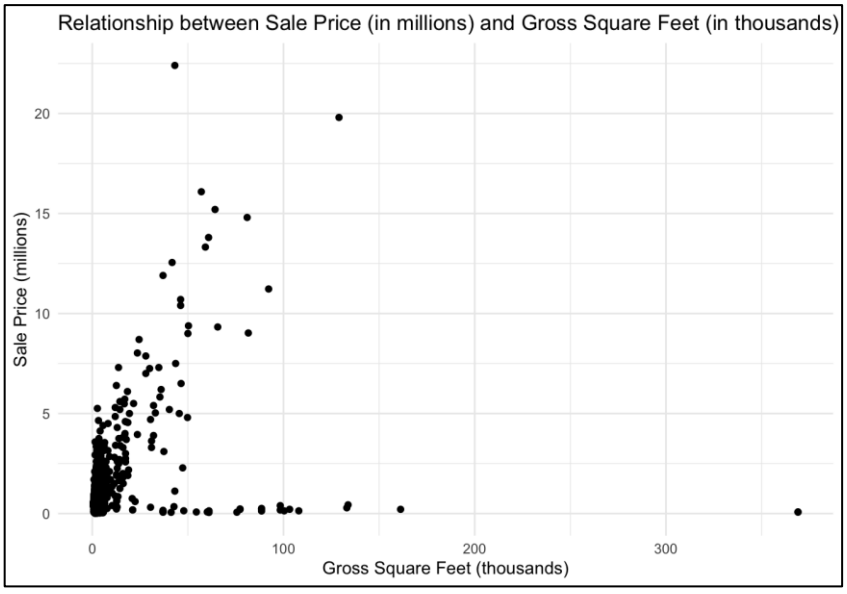
1. Make a scatterplot that shows SALE_PRICE on the y-axis and GROSS_SQUARE_FEET on the x-axis. What can you say about the relationship between these two variables?

Code:

My neighborhood is Bay Ridge.

```
#1
ggplot(bayridge, aes(x = GROSS_SQUARE_FEET / 1000, y = SALE_PRICE / 1e6)) +
  geom_point() +
  labs(
    title = "Relationship between Sale Price (in millions) and Gross Square Feet (in thousands)",
    x = "Gross Square Feet (thousands)",
    y = "Sale Price (millions)"
  ) +
  theme_minimal()
```

Results:



Interpretation:

There is a positive relationship between sale price and property size in square feet. However, some large units are priced lower, and smaller units can sell for higher prices. This suggests that other factors—such as year built, amenities, and location—also play significant roles in determining price.

2. Generate a simple linear regression model to predict SALE_PRICE using GROSS_SQUARE_FEET.

Code:

```
#2
model <- lm(SALE_PRICE ~ GROSS_SQUARE_FEET, data = bayridge)
```

3. Show a summary of your model.

Code:

```
#3
summary(model)
```

Results:

```
> summary(model)

Call:
lm(formula = SALE_PRICE ~ GROSS_SQUARE_FEET, data = bayridge)

Residuals:
      Min        1Q    Median        3Q       Max
-11122085   -299085   -117057    140931  20338499

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       851690.93   13358.28   63.76   <2e-16 ***
GROSS_SQUARE_FEET     28.02       1.22   22.97   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 899300 on 4999 degrees of freedom
Multiple R-squared:  0.09544,   Adjusted R-squared:  0.09526
F-statistic: 527.4 on 1 and 4999 DF,  p-value: < 2.2e-16
```

Interpretation:

GROSS_SQUARE_FEET explains 9.544 % of sale price, and is a statistically important independent variable.

4. Generate a prediction using this model, being sure to use a GROSS_SQUARE_FEET value from within the dataset range. Show that prediction, and explain it using the model equation.

Code:

```
#4
range(bayridge$GROSS_SQUARE_FEET)
predict(model, data.frame(GROSS_SQUARE_FEET=1500))

851690.93+28.02*1500
```

Results:

```
> #4
> range(bayridge$GROSS_SQUARE_FEET)
[1]    490 369046
> predict(model, data.frame(GROSS_SQUARE_FEET=1500))
       1
893715.7
> 851690.93+28.02*1500
[1] 893720.9
```

Interpretation:

GROSS_SQUARE_FEET values range between 490 and 369046 square feet in the 'bayridge' dataset. When it is 1500, sale price is predicted to be 893715.7.

Using the model outputs and coefficients, 851690.93 is an intercept, 28.02 is a beta coefficient for GROSS_SQUARE_FEET. It gives us 893720.9, close to the predicted value.

5. Remove the SALE_DATE variable.

Code:

```
#5
bayridge <- bayridge  %>% select(-SALE_DATE)
```

6. For the remaining variables in your dataset, sort them all as either numeric or categorical. To determine whether a variable is numeric, use the same two-part test that you used on the previous assignment:
    - Are its values represented by numbers?
    - If so, do those numbers have quantitative/numeric meaning?

Code:

```r
#6
numeric_vars <- bayridge %>%
  select(RESIDENTIAL_UNITS, COMMERCIAL_UNITS, GROSS_SQUARE_FEET, SALE_PRICE)

bayridge <- bayridge %>%
  mutate(across(c(SALE_ID, NEIGHBORHOOD_ID, BUILDING_CLASS_FINAL_ROLL, ADDRESS,
                  APARTMENT_NUMBER, ZIP_CODE, YEAR_BUILT, DESCRIPTION,
                  TYPE, NEIGHBORHOOD_NAME, BOROUGH_ID), as.factor))

categoric_vars <- bayridge %>%
  select(SALE_ID, NEIGHBORHOOD_ID, BUILDING_CLASS_FINAL_ROLL, ADDRESS, APARTMENT_NUMBER,
         ZIP_CODE, YEAR_BUILT, DESCRIPTION, TYPE,  NEIGHBORHOOD_NAME, BOROUGH_ID)
```

Results & Interpretation:

There are 15 varies in total. Only RESIDENTIAL_UNITS, COMMERCIAL_UNITS, GROSS_SQUARE_FEET, SALE_PRICE are numeric and have quantitative meaning. The rest I identified as categorical variables and converted them into factors.

7. If you have any categorical variables currently seen by R as a numeric or int type, convert them to factors now.

Code:

```r
#7
bayridge <- bayridge %>%
  mutate(
    SALE_ID = as.factor(SALE_ID),
    NEIGHBORHOOD_ID = as.factor(NEIGHBORHOOD_ID),
    YEAR_BUILT = as.factor(YEAR_BUILT),
    ZIP_CODE = as.factor(ZIP_CODE)
  )
str(bayridge)
```

Results:

```
> str(bayridge)
'data.frame':   5001 obs. of  15 variables:
 $ SALE_ID                 : Factor w/ 5001 levels "1531","2305",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ NEIGHBORHOOD_ID         : Factor w/ 1 level "16": 1 1 1 1 1 1 1 1 1 1 ...
 $ BUILDING_CLASS_FINAL_ROLL: Factor w/ 25 levels "A1","A2","A3",..: 5 1 5 3 7 8 13 9 19 1 ...
 $ ADDRESS                 : Factor w/ 4344 levels "1 78TH STREET",..: 710 773 167 2261 132 1920 2818 1412 4299 300 ...
 $ APARTMENT_NUMBER        : Factor w/ 64 levels "1","10","101",..: NA NA NA NA NA NA NA NA NA NA ...
 $ ZIP_CODE                : Factor w/ 4 levels "11209","11219",..: 1 1 1 1 3 3 3 1 1 1 ...
 $ RESIDENTIAL_UNITS       : num  1 1 1 1 2 2 6 2 91 1 ...
 $ COMMERCIAL_UNITS        : num  0 0 0 0 0 0 0 0 0 0 ...
 $ GROSS_SQUARE_FEET       : num  1216 2334 1328 3590 2128 ...
 $ YEAR_BUILT              : Factor w/ 81 levels "0","1899","1900",..: 23 9 23 33 23 72 21 18 46 9 ...
 $ SALE_PRICE              : num  380000 150000 450000 1275000 540000 ...
 $ DESCRIPTION             : Factor w/ 25 levels "CITY RESIDENCE ONE FAMILY",..: 16 23 16 13 20 22 11 21 10 23 ...
 $ TYPE                    : Factor w/ 1 level "RESIDENTIAL": 1 1 1 1 1 1 1 1 1 1 ...
 $ NEIGHBORHOOD_NAME       : Factor w/ 1 level "BAY RIDGE": 1 1 1 1 1 1 1 1 1 1 ...
 $ BOROUGH_ID              : Factor w/ 1 level "3": 1 1 1 1 1 1 1 1 1 1 ...
```

Interpretation:

I have already converted all the variables other than meaningful numeric ones to factors, as I found it to simplify things for me. Repeated the mutation for this question too.

8. For your categorical variables, how many just contain a single unique value? Remove all of these from your dataset.

Code:

```
#8 R
bayridge <- bayridge %>%
  select(where(~ n_distinct(.) > 1 | !is.factor(.)))
str(bayridge)
```

Results:

```
> str(bayridge)
'data.frame':   5001 obs. of  11 variables:
 $ SALE_ID                  : Factor w/ 5001 levels "1531","2305",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ BUILDING_CLASS_FINAL_ROLL: Factor w/ 25 levels "A1","A2","A3",..: 5 1 5 3 7 8 13 9 19 1 ...
 $ ADDRESS                  : Factor w/ 4344 levels "1 78TH STREET",..: 710 773 167 2261 132 1920 2818 1412 4299 300 ...
 $ APARTMENT_NUMBER         : Factor w/ 64 levels "1","10","101",..: NA NA NA NA NA NA NA NA NA NA ...
 $ ZIP_CODE                 : Factor w/ 4 levels "11209","11219",..: 1 1 1 1 3 3 3 1 1 1 ...
 $ RESIDENTIAL_UNITS        : num  1 1 1 1 2 2 6 2 91 1 ...
 $ COMMERCIAL_UNITS         : num  0 0 0 0 0 0 0 0 0 0 ...
 $ GROSS_SQUARE_FEET        : num  1216 2334 1328 3590 2128 ...
 $ YEAR_BUILT               : Factor w/ 81 levels "0","1899","1900",..: 23 9 23 33 23 72 21 18 46 9 ...
 $ SALE_PRICE               : num  380000 150000 450000 1275000 540000 ...
 $ DESCRIPTION              : Factor w/ 25 levels "CITY RESIDENCE ONE FAMILY",..: 16 23 16 13 20 22 11 21 10 23 ...
```

Interpretation:

As we can see from 'bayridge' structure in Question 7, there are 4 variables with single unique value – NEIGHBORHOOD_ID, TYPE, NEIGHBORHOOD_NAME and BOROUGH_ID. We could remove them separately or do as in my code above. After that we are left with 11 variables.

9. For your remaining categorical variables, how many contain more than 50 unique values? Remove all of these from your dataset.

Code:

```
#9
bayridge <- bayridge %>%
  select(where(~ n_distinct(.) <= 50 | !is.factor(.)))
str(bayridge)
```

Results:

```
> str(bayridge)
'data.frame':   5001 obs. of  7 variables:
 $ BUILDING_CLASS_FINAL_ROLL: Factor w/ 25 levels "A1","A2","A3",..: 5 1 5 3 7 8 13 9 19 1 ...
 $ ZIP_CODE                 : Factor w/ 4 levels "11209","11219",..: 1 1 1 1 3 3 3 1 1 1 ...
 $ RESIDENTIAL_UNITS        : num  1 1 1 1 2 2 6 2 91 1 ...
 $ COMMERCIAL_UNITS         : num  0 0 0 0 0 0 0 0 0 0 ...
 $ GROSS_SQUARE_FEET        : num  1216 2334 1328 3590 2128 ...
 $ SALE_PRICE               : num  380000 150000 450000 1275000 540000 ...
 $ DESCRIPTION              : Factor w/ 25 levels "CITY RESIDENCE ONE FAMILY",..: 16 23 16 13 20 22 11 21 10 23 ...
```

Interpretation:

After dropping categorical variables that contain more than 50 unique values, there are 7 variables left in the sample.

10. For any categorical variables still remaining, filter your dataset so that only the 6 most common types remain.

Code:

```
# 10.
top_6_building <- bayridge %>%
  count(BUILDING_CLASS_FINAL_ROLL, sort = TRUE) %>%
  filter(BUILDING_CLASS_FINAL_ROLL %in% unique(bayridge$BUILDING_CLASS_FINAL_ROLL)) %>%
  slice_max(n, n = 6) %>%
  pull(BUILDING_CLASS_FINAL_ROLL)

top_6_desc <- bayridge %>%
  count(DESCRIPTION, sort = TRUE) %>%
  filter(DESCRIPTION %in% unique(bayridge$DESCRIPTION)) %>%
  slice_max(n, n = 6) %>%
  pull(DESCRIPTION)

bayridge <- bayridge %>%
  filter(
    BUILDING_CLASS_FINAL_ROLL %in% top_6_building,
    DESCRIPTION %in% top_6_desc
  )
bayridge <- droplevels(bayridge)
str(bayridge)
```

Results:

```
> str(bayridge)
'data.frame':   3993 obs. of  7 variables:
 $ BUILDING_CLASS_FINAL_ROLL: Factor w/ 6 levels "A1","A5","B1",..: 2 1 2 3 4 5 1 3 6 2 ...
 $ ZIP_CODE                 : Factor w/ 4 levels "11209","11219",..: 1 1 1 3 3 1 1 3 4 1 ...
 $ RESIDENTIAL_UNITS        : num  1 1 1 2 2 2 1 2 3 1 ...
 $ COMMERCIAL_UNITS         : num  0 0 0 0 0 0 0 0 0 0 ...
 $ GROSS_SQUARE_FEET        : num  1216 2334 1328 2128 5508 ...
 $ SALE_PRICE               : num  380000 150000 450000 540000 500000 ...
 $ DESCRIPTION              : Factor w/ 6 levels "ONE FAMILY ATTACHED OR SEMI-DETACHED",..: 1 6 1 3 5 4 6 3 2 1 ...
```

Interpretation:

Since there are only 4 types in zip codes, I filtered dataset with 6 most common types remaining for BUILDING_CLASS_FINAL_ROLL and DESCRIPTION.

11. Next, let's check for the correlations among your numeric inputs. Generate a correlation table for your numeric input variables. Do any show correlations higher than 0.70? If so, remove one member from any highly-correlated pair.

Code:

```
#11

cor_matrix <- cor(numeric_vars, use = "complete.obs")
print(cor_matrix)

bayridge <- bayridge %>% select(-RESIDENTIAL_UNITS)
numeric_vars <- numeric_vars %>% select(-RESIDENTIAL_UNITS)

cor_matrix <- cor(numeric_vars, use = "complete.obs")
print(cor_matrix)
str(bayridge)
```

Results:

```
> cor_matrix <- cor(numeric_vars, use = "complete.obs")
> print(cor_matrix)
                  RESIDENTIAL_UNITS COMMERCIAL_UNITS GROSS_SQUARE_FEET SALE_PRICE
RESIDENTIAL_UNITS         1.00000000       0.09375812        0.98165464  0.3054745
COMMERCIAL_UNITS          0.09375812       1.00000000        0.09824653  0.1957221
GROSS_SQUARE_FEET         0.98165464       0.09824653        1.00000000  0.3089272
SALE_PRICE                0.30547453       0.19572213        0.30892717  1.0000000
> bayridge <- bayridge %>% select(-RESIDENTIAL_UNITS)
> numeric_vars <- numeric_vars %>% select(-RESIDENTIAL_UNITS)
> cor_matrix <- cor(numeric_vars, use = "complete.obs")
> print(cor_matrix)
                  COMMERCIAL_UNITS GROSS_SQUARE_FEET SALE_PRICE
COMMERCIAL_UNITS         1.00000000        0.09824653  0.1957221
GROSS_SQUARE_FEET        0.09824653        1.00000000  0.3089272
SALE_PRICE               0.19572213        0.30892717  1.0000000
```

Interpretation:

Since there is high correlation (>0.98) between RESIDENTIAL_UNITS and GROSS_SQUARE_FEET, I removed one of them - RESIDENTIAL_UNITS. Now there are 3 numeric variables remaining in 'bayridge'.

12. Use a multiple regression model to determine the SALE_PRICE of a given residential property in your neighborhood. As your inputs, include all the variables that still remain.

Code:

Before running the multiple regression, I needed to perform some checking and prepare the data.

```
#12

summary(bayridge)
table(bayridge$COMMERCIAL_UNITS)
table(bayridge$BUILDING_CLASS_FINAL_ROLL)
table(bayridge$DESCRIPTION)
table(bayridge$ZIP_CODE)

table(bayridge$BUILDING_CLASS_FINAL_ROLL, bayridge$DESCRIPTION)


bayridge <- bayridge %>% select(-COMMERCIAL_UNITS)

bayridge <- bayridge %>% select(-DESCRIPTION)
model2 <- lm(SALE_PRICE ~ ., data = bayridge)
```

Results:

```
> summary(bayridge)
 BUILDING_CLASS_FINAL_ROLL  ZIP_CODE    COMMERCIAL_UNITS GROSS_SQUARE_FEET   SALE_PRICE
 A1: 607                   11209:3098   Min.   :0        Min.   : 720      Min.   :  12000
 A5:1084                   11219:   2   1st Qu.:0        1st Qu.:1632      1st Qu.: 610000
 B1:1350                   11220: 571   Median :0        Median :2152      Median : 797000
 B2: 359                   11228: 322   Mean   :0        Mean   :2186      Mean   : 846726
 B3: 304                                3rd Qu.:0        3rd Qu.:2634      3rd Qu.: 999000
 C0: 289                                Max.   :0        Max.   :6877      Max.   :5255000
                              DESCRIPTION
 ONE FAMILY ATTACHED OR SEMI-DETACHED:1084
 THREE FAMILIES                     : 289
 TWO FAMILY BRICK                   :1350
 TWO FAMILY CONVERTED FROM ONE FAMILY: 304
 TWO FAMILY FRAME                   : 359
 TWO STORIES - DETACHED SM OR MID   : 607
```

```
> table(bayridge$COMMERCIAL_UNITS)

   0
3993
> table(bayridge$BUILDING_CLASS_FINAL_ROLL)

  A1   A5   B1   B2   B3   C0
 607 1084 1350  359  304  289
> table(bayridge$DESCRIPTION)

ONE FAMILY ATTACHED OR SEMI-DETACHED                    THREE FAMILIES                    TWO FAMILY BRICK
                                1084                               289                              1350
TWO FAMILY CONVERTED FROM ONE FAMILY                    TWO FAMILY FRAME    TWO STORIES - DETACHED SM OR MID
                                 304                               359                               607
```

```
> table(bayridge$ZIP_CODE)

11209 11219 11220 11228
 3098     2   571   322
> table(bayridge$BUILDING_CLASS_FINAL_ROLL, bayridge$DESCRIPTION)

    ONE FAMILY ATTACHED OR SEMI-DETACHED THREE FAMILIES TWO FAMILY BRICK TWO FAMILY CONVERTED FROM ONE FAMILY
  A1                                    0              0                0                                     0
  A5                                 1084              0                0                                     0
  B1                                    0              0             1350                                     0
  B2                                    0              0                0                                     0
  B3                                    0              0                0                                   304
  C0                                    0            289                0                                     0

    TWO FAMILY FRAME TWO STORIES - DETACHED SM OR MID
  A1                0                             607
  A5                0                               0
  B1                0                               0
  B2              359                               0
  B3                0                               0
  C0                0                               0
```

Interpretation:

As you can see, COMMERCIAL_UNITS contains only 0s after all filtering applied, thus I had to drop it. From both frequency tables and contingency table we observe perfect one-to-one relationship between BUILDING_CLASS_FINAL_ROLL and DESCRIPTION, we drop one of them. Model includes all the remaining variables.

13. Show your model summary. Which numeric variables, if any, show a p-value that suggests statistical insignificance?

Code:

```
#13
summary(model2)
```

Results:

```
> summary(model2)

Call:
lm(formula = SALE_PRICE ~ ., data = bayridge)

Residuals:
     Min       1Q   Median       3Q      Max
-1269255  -199522   -51379   192982  4415067

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                  819981.97   25352.64  32.343  < 2e-16 ***
BUILDING_CLASS_FINAL_ROLLA5 -285831.63   18669.75 -15.310  < 2e-16 ***
BUILDING_CLASS_FINAL_ROLLB1 -271528.30   18160.50 -14.952  < 2e-16 ***
BUILDING_CLASS_FINAL_ROLLB2 -231070.18   23962.38  -9.643  < 2e-16 ***
BUILDING_CLASS_FINAL_ROLLB3 -235285.63   24973.55  -9.421  < 2e-16 ***
BUILDING_CLASS_FINAL_ROLLC0 -247097.55   27524.85  -8.977  < 2e-16 ***
ZIP_CODE11219               -393618.10  250009.79  -1.574 0.115472
ZIP_CODE11220                -59926.25   16256.65  -3.686 0.000231 ***
ZIP_CODE11228                -62602.29   20900.93  -2.995 0.002760 **
GROSS_SQUARE_FEET               121.91      10.15  12.008  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 352500 on 3983 degrees of freedom
Multiple R-squared:   0.13,     Adjusted R-squared:  0.1281
F-statistic: 66.14 on 9 and 3983 DF,  p-value: < 2.2e-16
```

Interpretation:

After all filtering applied, the number of observations is approximately 4,000. Various building classes are associated with decreases in sale price compared to the reference category (A1). Location (ZIP codes) affects sale price too. The ZIP codes listed (11219, 11220, and 11228) have negative coefficients, which indicates that properties in these areas tend to be cheaper compared to the reference ZIP code 11209. ZIP code 11219 decreases sale price by 393,618, but this effect is not statistically significant ($p = 0.115$). ZIP codes 11220 and 11228 also reduce sale price, with significant effects.

A positive coefficient for GROSS_SQUARE_FEET (121.91) indicates that for each additional square foot, the sale price increases by approximately $121.91. This effect is highly significant, supporting a positive relationship between size and sale price.

Model explains only a small portion of sale price variability – 13%, suggesting other factors (e.g., amenities, proximity to transit, market conditions) may play important roles in determining sale prices.

14. Using your model, generate a prediction for a fictional real estate listing in your neighborhood, being sure to use input values from within the dataset range. Show that prediction, and explain it using the model equation.

Code:

```
#14
range(bayridge$GROSS_SQUARE_FEET)

new_data <- data.frame(
  GROSS_SQUARE_FEET = 1500,
  ZIP_CODE = factor("11219", levels = c("11219", "11220", "11228")),
  BUILDING_CLASS_FINAL_ROLL = factor("B2", levels = c("A5", "B1", "B2", "B3", "C0"))
)
predicted_price <- predict(model2, new_data)
predicted_price


819981.97 + 1500*121.91 + 1*(-231070.18) + 1*(-393618.10)
```

Results:

```
> #14
> range(bayridge$GROSS_SQUARE_FEET)
[1]  720 6877
> new_data <- data.frame(
+   GROSS_SQUARE_FEET = 1500,
+   ZIP_CODE = factor("11219", levels = c("11219", "11220", "11228")),
+   BUILDING_CLASS_FINAL_ROLL = factor("B2", levels = c("A5", "B1", "B2", "B3", "C0"))
+ )
> predicted_price <- predict(model2, new_data)
> predicted_price
       1
378165.7
> 819981.97 + 1500*121.91 + 1*(-231070.18) + 1*(-393618.10)
[1] 378158.7
```

Interpretation:

GROSS_SQUARE_FEET ranges between 720 to 6877. Using 1500 GROSS_SQUARE_FEET, ZIP CODE 11219, B2 Building Class, we get a predicted price of 378,165.7. Applying the relevant coefficients, the model equation confirms this prediction, yielding a similar value."

15. According to your model, which three properties were the biggest bargains and which three were the most overpriced? How might you account for these disparities?

Code:

```r
#15
bayridge$residuals <- residuals(model2)

bayridge_sorted <- bayridge[order(bayridge$residuals), ]

print(bayridge_sorted[1:3, ])

print(bayridge_sorted[nrow(bayridge_sorted): (nrow(bayridge_sorted) - 2), ])
```

Results:

```
> #15
> bayridge$residuals <- residuals(model2)
> bayridge_sorted <- bayridge[order(bayridge$residuals), ]
> print(bayridge_sorted[1:3, ])
     BUILDING_CLASS_FINAL_ROLL ZIP_CODE GROSS_SQUARE_FEET SALE_PRICE residuals
1202                        A1    11209              3882      24000  -1269255
563                         A1    11209              2724      50000  -1102078
3643                        A1    11209              2240      50000  -1043071
> print(bayridge_sorted[nrow(bayridge_sorted): (nrow(bayridge_sorted) - 2), ])
     BUILDING_CLASS_FINAL_ROLL ZIP_CODE GROSS_SQUARE_FEET SALE_PRICE residuals
167                         C0    11220              2682    5255000   4415067
3993                        A1    11209              4110    4125000   2803949
3497                        A1    11209              2947    3480000   2300735
```

Interpretation:

To find the biggest bargains and most overpriced properties, we need to look at the residuals from the model. The residuals represent the difference between the actual sale price and the predicted sale price.

The biggest bargains will be the properties with the most negative residuals, as they were sold for less than their predicted price. The most overpriced properties will be those with the most positive residuals, as they were sold for more than their predicted price.

A few possible explanations for the disparities between the actual and predicted sale prices might be:

- Condition of the property
- Amenities (such as a view, a backyard, or a pool)
- Location within the neighborhood
- Time of year
- Negotiation: The model does not account for negotiation between the buyer and seller. A skilled negotiator may be able to get a better price for the property than the predicted price.
- Market trends
- Emotional Purchases: Sometimes properties are bought or sold at emotional prices, not reflecting market value.
- Economic Factors and Investment Climate

- Condition and Age

Model wise:

- Non-linear Relationships: The model assumes linear relationships, but property values might not always follow this pattern.
- Measurement Errors: Errors in recording `GROSS_SQUARE_FEET` or other variables could lead to incorrect predictions.
- Outliers: Extreme values in the dataset might skew the model's predictions.

There may be other factors at play as well.