**Student: Seif Kungulio**

**Date: 02/23/2025**

**Subject: Project 6**

**Class: DSCI 502**

**Section: 01W**

**Instructor: Sean Yang**

**File Name: Project6\_Kungulio\_Seif.docx**

1. Load the dataset in kc\_house\_data.csv into R. Call the loaded data kc\_house\_data. Make sure that you have the directory set to the correct location for the data.

>

> ## 1. Load the dataset in kc\_house\_data.csv into R. Call the loaded data

> ## kc\_house\_data. Make sure that you have the directory set to the correct

> ## location for the data.

>

> # Set the working directory to the correct location for the dataset.

> setwd("C:/PROJECTS/Maryville/DSCI 502/Project6")

>

> # Load the data from loan.csv

> kc\_house\_data <- read.csv("kc\_house\_data.csv")

>

> # Display the dimensions (rows and columns) of the dataframe

> dim(kc\_house\_data) # Shows the number of rows and columns in the dataset.

[1] 21613 21

>

> # Display column names

> colnames(kc\_house\_data)

[1] "id" "date" "price" "bedrooms" "bathrooms" "sqft\_living"

[7] "sqft\_lot" "floors" "waterfront" "view" "condition" "grade"

[13] "sqft\_above" "sqft\_basement" "yr\_built" "yr\_renovated" "zipcode" "lat"

[19] "long" "sqft\_living15" "sqft\_lot15"

>

> # Displays the structure of the kc\_house\_data object.

> str(kc\_house\_data)

'data.frame': 21613 obs. of 21 variables:

$ id : num 7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...

$ date : chr "20141013T000000" "20141209T000000" "20150225T000000" "20141209T000000" ...

$ price : num 221900 538000 180000 604000 510000 ...

$ bedrooms : int 3 3 2 4 3 4 3 3 3 3 ...

$ bathrooms : num 1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...

$ sqft\_living : int 1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...

$ sqft\_lot : int 5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...

$ floors : num 1 2 1 1 1 1 2 1 1 2 ...

$ waterfront : int 0 0 0 0 0 0 0 0 0 0 ...

$ view : int 0 0 0 0 0 0 0 0 0 0 ...

$ condition : int 3 3 3 5 3 3 3 3 3 3 ...

$ grade : int 7 7 6 7 8 11 7 7 7 7 ...

$ sqft\_above : int 1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...

$ sqft\_basement: int 0 400 0 910 0 1530 0 0 730 0 ...

$ yr\_built : int 1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...

$ yr\_renovated : int 0 1991 0 0 0 0 0 0 0 0 ...

$ zipcode : int 98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...

$ lat : num 47.5 47.7 47.7 47.5 47.6 ...

$ long : num -122 -122 -122 -122 -122 ...

$ sqft\_living15: int 1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...

$ sqft\_lot15 : int 5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...

>

A screenshot of a computer program

AI-generated content may be incorrect.

1. Build a linear model to forecast the price using bedrooms, bathrooms, and sqft\_living.

>

> ## 2. Build a linear model to forecast the price using bedrooms, bathrooms,

> ## and sqft\_living.

>

> # Convert the following features to factor

> features <- c("waterfront", "floors", "view", "condition")

>

> # Loop over each feature and convert it to a factor

> for (feature in features) {

+ kc\_house\_data[[feature]] <- as.factor(kc\_house\_data[[feature]])

+ }

>

> # Build a basic multiple linear regression model

> qn2.fit <- lm(price ~ bedrooms + bathrooms + sqft\_living, data = kc\_house\_data)

>

> # Display statistical summary of the model

> summary(qn2.fit)

Call:

lm(formula = price ~ bedrooms + bathrooms + sqft\_living, data = kc\_house\_data)

Residuals:

Min 1Q Median 3Q Max

-1644794 -144361 -22891 102420 4178611

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 74662.099 6917.977 10.792 <2e-16 \*\*\*

bedrooms -57906.631 2336.062 -24.788 <2e-16 \*\*\*

bathrooms 7928.708 3512.744 2.257 0.024 \*

sqft\_living 309.605 3.089 100.238 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 258000 on 21609 degrees of freedom

Multiple R-squared: 0.5069, Adjusted R-squared: 0.5069

F-sta tistic: 7406 on 3 and 21609 DF, p-value: < 2.2e-16

A computer screen shot of a computer program

AI-generated content may be incorrect.

1. Then write down the corresponding math formula.

Math formula:

**price = β0 + β1\*bedrooms + β2\*bathrooms + β3\*sqft\_living**

1. Is it a good model based on R square or adjusted R square?

>

> ##### b. Is it a good model based on R square or adjusted R square?

>

> # Retrieve R-squared

> summary(qn2.fit)$r.squared

[1] 0.5069364

>

A black screen with blue and purple text

AI-generated content may be incorrect.

The model's R-squared value of 0.5069364 indicates a moderate fit, explaining 50.7% of house price variance using bedrooms, bathrooms, and sqft\_living. However, a significant portion remains unexplained. Improving the model by adding other predictors could enhance accuracy.

1. Build a linear model to forecast the price using bedrooms, bathrooms, sqft\_living, and all the cross effects between them.

>

> ## 3. Build a linear model to forecast the price using bedrooms, bathrooms,

> ## sqft\_living, and all the cross effects between them.

>

> # Build a model with interaction effects

> qn3.fit <- lm(price ~ bedrooms \* bathrooms \* sqft\_living, data = kc\_house\_data)

>

> # Display statistical summary of the model

> summary(qn3.fit)

Call:

lm(formula = price ~ bedrooms \* bathrooms \* sqft\_living, data = kc\_house\_data)

Residuals:

Min 1Q Median 3Q Max

-3763654 -136129 -28111 95553 3367308

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 4.339e+05 2.422e+04 17.916 < 2e-16 \*\*\*

bedrooms -4.670e+04 7.338e+03 -6.364 2.01e-10 \*\*\*

bathrooms -1.203e+05 1.382e+04 -8.708 < 2e-16 \*\*\*

sqft\_living -2.490e+01 1.597e+01 -1.559 0.119

bedrooms:bathrooms -4.825e+03 3.570e+03 -1.352 0.177

bedrooms:sqft\_living 3.227e+01 3.899e+00 8.278 < 2e-16 \*\*\*

bathrooms:sqft\_living 1.121e+02 5.014e+00 22.360 < 2e-16 \*\*\*

bedrooms:bathrooms:sqft\_living -1.015e+01 9.673e-01 -10.493 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 248200 on 21605 degrees of freedom

Multiple R-squared: 0.5436, Adjusted R-squared: 0.5435

F-statistic: 3677 on 7 and 21605 DF, p-value: < 2.2e-16

A computer screen shot of a black screen

AI-generated content may be incorrect.

1. Then write down the corresponding math formula.

Math formula:

**price = β0 + β1\*bedrooms + β2\*bathrooms + β3\*sqft\_living +**

**β4\*(bedrooms\*bathrooms) + β5\*(bedrooms\*sqft\_living) +**

**β6\*(bathrooms\*sqft\_living) + β7\*(bedrooms\*bathrooms\*sqft\_living)**

1. Is it a better model than the model in Question 2 based on adjusted R square?

>

> ##### b. Is it a better model than the model in Question 2 based on

> ##### adjusted R square?

>

> # Compares adjusted R-squared values to determine which

> # model (Q2 or Q3) performs better.

> adjusted\_r\_squared\_values <- c(

+ summary(qn2.fit)$adj.r.squared, summary(qn3.fit)$adj.r.squared

+ )

>

> best\_model\_index <- which(adjusted\_r\_squared\_values ==

+ max(adjusted\_r\_squared\_values))

>

> # Display the best model index

> best\_model\_index

[1] 2

>

A computer screen shot of a black screen

AI-generated content may be incorrect.

The model (qn3.fit) is superior based on Adjusted R-Squared (0.5435 vs. 0.5069). This indicates it explains more variance in house prices, making it a better fit. Including interaction effects enhances the model’s predictive power.

1. Build a linear model to forecast the price using bedrooms, bathrooms, sqft\_living, waterfront, and grade.

>

> ## 4. Build a linear model to forecast the price using bedrooms, bathrooms,

> ## sqft\_living, waterfront, and grade.

>

> # Build a basic multiple linear regression model by adding more predictors

> qn4.fit <- lm(price ~ bedrooms + bathrooms + sqft\_living + waterfront + grade,

+ data = kc\_house\_data)

>

> # Display statistical summary of the model

> summary(qn4.fit)

Call:

lm(formula = price ~ bedrooms + bathrooms + sqft\_living + waterfront +

grade, data = kc\_house\_data)

Residuals:

Min 1Q Median 3Q Max

-1263684 -130226 -20778 98943 4755524

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -4.883e+05 1.433e+04 -34.09 < 2e-16 \*\*\*

bedrooms -3.198e+04 2.210e+03 -14.47 < 2e-16 \*\*\*

bathrooms -2.554e+04 3.347e+03 -7.63 2.44e-14 \*\*\*

sqft\_living 2.134e+02 3.455e+00 61.78 < 2e-16 \*\*\*

waterfront1 7.991e+05 1.893e+04 42.22 < 2e-16 \*\*\*

grade 9.669e+04 2.227e+03 43.42 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 238600 on 21607 degrees of freedom

Multiple R-squared: 0.5782, Adjusted R-squared: 0.5781

F-statistic: 5923 on 5 and 21607 DF, p-value: < 2.2e-16

A screenshot of a computer program

AI-generated content may be incorrect.

1. Then write down the corresponding math formula.

Math formula:

**price = β0 + β1\*bedrooms + β2\*bathrooms + β3\*sqft\_living +**

**β4\*waterfront + β5\*grade**

1. Is it a better model than the model in Question 3 based on adjusted R square?

>

> ##### b. Is it a better model than the model in Question 3 based on

> ##### adjusted R square?

>

> # Better model recommendation

> models <- list(qn3.fit = qn3.fit, qn4.fit = qn4.fit)

>

> # Iterates through models and extracts adjusted R-squared values for comparison

> for (i in seq\_along(models)) {

+ adjusted\_r\_squared\_values[i] <- summary(models[[i]])$adj.r.squared

+ }

> best\_model\_index <- which(adjusted\_r\_squared\_values ==

+ max(adjusted\_r\_squared\_values))

>

> # Identifies and prints the best model based on adjusted R-squared.

> cat("Best Model is", names(models)[best\_model\_index],

+ "with Adjusted R-Squared of",

+ summary(models[[best\_model\_index]])$adj.r.squared)

Best Model is qn4.fit with Adjusted R-Squared of 0.578082>

>

A computer screen shot of white text

AI-generated content may be incorrect.

Model qn4.fit outperforms qn3.fit based on Adjusted R-Squared (0.5781 vs. 0.5435), indicating it explains more variance in house prices and is a better fit.

1. Build a linear model to forecast the price using all other columns except id, date, zipcode, lat, and long without a y-intercept. If we only consider the models defined in Q2, Q3, Q4 and Q5, which model do you recommend based on the adjusted R squared value?

>

> ## 5. Build a linear model to forecast the price using all other columns except

> ## id, date, zipcode, lat, and long without a y-intercept.

> ## If we only consider the models defined in Q2, Q3, Q4 and Q5,

> ## which model do you recommend based on the adjusted R squared value?

>

> # Uses all variables except id, date, zipcode, lat, and long, ensuring

> # better prediction by removing non-informative or redundant variables.

> qn5.fit <- lm(price ~ . -id -date -zipcode -lat -long -1, data = kc\_house\_data)

>

> # Display statistical summary of the model

> summary(qn5.fit)

Call:

lm(formula = price ~ . - id - date - zipcode - lat - long - 1,

data = kc\_house\_data)

Residuals:

Min 1Q Median 3Q Max

-1264731 -109707 -10444 88970 4186439

Coefficients: (1 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

bedrooms -3.797e+04 2.024e+03 -18.762 < 2e-16 \*\*\*

bathrooms 4.880e+04 3.484e+03 14.005 < 2e-16 \*\*\*

sqft\_living 1.581e+02 4.664e+00 33.899 < 2e-16 \*\*\*

sqft\_lot 6.832e-04 5.097e-02 0.013 0.98931

floors1 6.262e+06 1.530e+05 40.917 < 2e-16 \*\*\*

floors1.5 6.257e+06 1.514e+05 41.339 < 2e-16 \*\*\*

floors2 6.253e+06 1.540e+05 40.598 < 2e-16 \*\*\*

floors2.5 6.389e+06 1.531e+05 41.734 < 2e-16 \*\*\*

floors3 6.385e+06 1.552e+05 41.148 < 2e-16 \*\*\*

floors3.5 6.534e+06 1.718e+05 38.027 < 2e-16 \*\*\*

waterfront1 4.911e+05 2.099e+04 23.397 < 2e-16 \*\*\*

view1 1.037e+05 1.202e+04 8.633 < 2e-16 \*\*\*

view2 4.680e+04 7.299e+03 6.412 1.46e-10 \*\*\*

view3 1.007e+05 1.001e+04 10.059 < 2e-16 \*\*\*

view4 2.820e+05 1.538e+04 18.331 < 2e-16 \*\*\*

condition2 -1.427e+04 4.251e+04 -0.336 0.73713

condition3 -1.498e+04 3.941e+04 -0.380 0.70381

condition4 -4.677e+03 3.941e+04 -0.119 0.90553

condition5 3.368e+04 3.965e+04 0.849 0.39570

grade 1.178e+05 2.245e+03 52.463 < 2e-16 \*\*\*

sqft\_above 9.647e+00 4.649e+00 2.075 0.03798 \*

sqft\_basement NA NA NA NA

yr\_built -3.554e+03 7.759e+01 -45.796 < 2e-16 \*\*\*

yr\_renovated 1.094e+01 3.909e+00 2.799 0.00514 \*\*

sqft\_living15 2.984e+01 3.602e+00 8.284 < 2e-16 \*\*\*

sqft\_lot15 -5.534e-01 7.784e-02 -7.109 1.20e-12 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 214500 on 21588 degrees of freedom

Multiple R-squared: 0.8923, Adjusted R-squared: 0.8922

F-statistic: 7156 on 25 and 21588 DF, p-value: < 2.2e-16

A screenshot of a computer program

AI-generated content may be incorrect.

>

> # Model recommendation

> models <- setNames(

+ list(qn2.fit, qn3.fit, qn4.fit, qn5.fit),

+ c("qn2.fit", "qn3.fit", "qn4.fit", "qn5.fit")

+ )

>

> # Iterates through models and extracts adjusted R-squared values for comparison

> for (i in seq\_along(models)) {

+ adjusted\_r\_squared\_values[i] <- summary(models[[i]])$adj.r.squared

+ }

> best\_model\_index <- which(adjusted\_r\_squared\_values ==

+ max(adjusted\_r\_squared\_values))

>

> # Identifies and prints the best model based on adjusted R-squared.

> cat("Best Model is", names(models)[best\_model\_index],

+ "with Adjusted R-Squared of",

+ summary(models[[best\_model\_index]])$adj.r.squared)

Best Model is qn5.fit with Adjusted R-Squared of 0.892199>

>

A computer screen shot of a black screen

AI-generated content may be incorrect.

Model Q5 (qn5.fit) is the best, with the highest Adjusted R-Squared (0.8922), outperforming Q2, Q3, and Q4. This indicates that using all relevant features except id, date, zipcode, lat, and long without an intercept significantly enhances predictive accuracy.

1. You are asked to build a linear model to forecast price using bedrooms, bathrooms, sqft\_living, sqft\_lot, floors, waterfront, view, condition, and grade. Then you are given the flowing new house info:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| bedrooms | bathrooms | sqft\_living | sqft\_lot | floor | waterfront | view | condition | grade |
| 4 | 2 | 2560 | 7650 | 1.5 | 1 | 3 | 5 | 10 |

>

> ## 6. You are asked to build a linear model to forecast price using bedrooms,

> ## bathrooms, sqft\_living, sqft\_lot, floors, waterfront, view, condition,

> # and grade. Then you are given the flowing new house info:

>

> # | bedrooms | bathrooms | sqft\_living | sqft\_lot | floors | waterfront | view | condition | grade |

> # |----------|-----------|-------------|----------|--------|------------|------|-----------|-------|

> # | 4 | 2 | 2560 | 7650 | 1.5 | 1 | 3 | 5 | 10 |

>

> # Linear model to forecast price

> qn6.fit <- lm(price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot + floors +

+ waterfront + view + condition + grade, data = kc\_house\_data)

>

> # Statistical summary of the model

> summary(qn6.fit)

Call:

lm(formula = price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot +

floors + waterfront + view + condition + grade, data = kc\_house\_data)

Residuals:

Min 1Q Median 3Q Max

-1273136 -121300 -15773 93698 4404586

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -4.763e+05 4.305e+04 -11.062 <2e-16 \*\*\*

bedrooms -3.689e+04 2.137e+03 -17.258 <2e-16 \*\*\*

bathrooms 2.621e+03 3.459e+03 0.758 0.4485

sqft\_living 1.944e+02 3.445e+00 56.421 <2e-16 \*\*\*

sqft\_lot -3.561e-01 3.821e-02 -9.320 <2e-16 \*\*\*

floors1.5 7.923e+04 5.696e+03 13.909 <2e-16 \*\*\*

floors2 -5.489e+04 4.213e+03 -13.030 <2e-16 \*\*\*

floors2.5 1.719e+05 1.829e+04 9.401 <2e-16 \*\*\*

floors3 2.535e+04 1.002e+04 2.530 0.0114 \*

floors3.5 2.038e+05 8.031e+04 2.537 0.0112 \*

waterfront1 5.041e+05 2.217e+04 22.743 <2e-16 \*\*\*

view1 1.440e+05 1.263e+04 11.398 <2e-16 \*\*\*

view2 8.768e+04 7.622e+03 11.503 <2e-16 \*\*\*

view3 1.490e+05 1.044e+04 14.279 <2e-16 \*\*\*

view4 3.320e+05 1.616e+04 20.549 <2e-16 \*\*\*

condition2 -5.852e+04 4.493e+04 -1.303 0.1927

condition3 -1.046e+05 4.160e+04 -2.513 0.0120 \*

condition4 -6.373e+04 4.163e+04 -1.531 0.1258

condition5 8.090e+03 4.188e+04 0.193 0.8469

grade 1.064e+05 2.213e+03 48.058 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 226900 on 21593 degrees of freedom

Multiple R-squared: 0.6189, Adjusted R-squared: 0.6186

F-statistic: 1846 on 19 and 21593 DF, p-value: < 2.2e-16

A screenshot of a computer program

AI-generated content may be incorrect.

* 1. Predict the average sales price for this house.

>

> ##### a. Predict the average sales price for this house.

>

> # New house information

> new\_house <- data.frame(bedrooms = 4,

+ bathrooms = 2,

+ sqft\_living = 2560,

+ sqft\_lot = 7650,

+ floors = 1.5,

+ waterfront = 1,

+ view = 3,

+ condition = 5,

+ grade = 10

+ )

>

> # Convert the following features to factor

> features <- c("waterfront", "floors", "view", "condition")

>

> # Loop over each feature and convert it to a factor

> for (feature in features) {

+ new\_house[[feature]] <- as.factor(new\_house[[feature]])

+ }

>

> # Predict and display the average price

> predicted\_price <- predict(qn6.fit, newdata = new\_house)

> cat("The average price for the new house is", predicted\_price)

The average price for the new house is 1680306

>

A screenshot of a computer program

AI-generated content may be incorrect.

* 1. Predict the 95% predicted interval for this house.

>

> ##### b. Predict the 95% predicted interval for this house.

>

> # Provides a 95% prediction interval

> prediction\_interval <- predict(qn6.fit, newdata = new\_house,

+ interval = "prediction")

> # Display the predicted intervals

> prediction\_interval

fit lwr upr

1 1680306 1232883 2127729

>

> cat(

+ "\nThe predicted price for the new house:",

+ prediction\_interval[1],

+ "\nThe lower end price of the 95% prediction interval:",

+ prediction\_interval[2],

+ "\nThe upper end price of the 95% prediction interval:",

+ prediction\_interval[3]

+ )

The predicted price for the new house: 1680306

The lower end price of the 95% prediction interval: 1232883

The upper end price of the 95% prediction interval: 2127729

>

A computer screen with text and numbers

AI-generated content may be incorrect.