Assaf Bitton

We will be predicting the housing price using the sahp dataset in the **r02pro** R package. Please answer the following questions. **Add your R code blocks as needed.**

You can run the following code to prepare the analysis.

```
library(r02pro)     #INSTALL IF NECESSARY

## Warning: package 'r02pro' was built under R version 4.4.3

library(tidyverse)  #INSTALL IF NECESSARY

## Warning: package 'ggplot2' was built under R version 4.4.3

## Warning: package 'tidyr' was built under R version 4.4.2

## Warning: package 'dplyr' was built under R version 4.4.2

my_sahp <- sahp %>%
  na.omit() %>%
  dplyr::select(gar_car, liv_area, kit_qual, sale_price)
my_sahp_train <- my_sahp[1:100, ]
my_sahp_test <- my_sahp[-(1:100), ]
```

Q1

Use the training data my_sahp_train to fit a simple linear regression model of sale_price on each variable (gar_car, liv_area, kit_qual) separately. For each regression,

a. Interpret the coefficients and compute the R^2. Which variable is most useful in predicting the sale_price on the training data?

```
fit_sale_price_gar_car <-
  lm(sale_price~gar_car, data= my_sahp_train)
summary(fit_sale_price_gar_car)

##
## Call:
## lm(formula = sale_price ~ gar_car, data = my_sahp_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -154.239  -43.754   -8.877   24.023  292.985
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   69.515     16.692   4.165 6.72e-05 ***
## gar_car       60.908      8.948   6.807 8.01e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 64.62 on 98 degrees of freedom
## Multiple R-squared:  0.321,  Adjusted R-squared:  0.3141
## F-statistic: 46.34 on 1 and 98 DF,  p-value: 8.011e-10

fit_sale_price_liv_area <-
  lm(sale_price~liv_area, data= my_sahp_train)
summary(fit_sale_price_liv_area)

##
## Call:
## lm(formula = sale_price ~ liv_area, data = my_sahp_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -158.979  -30.101    1.136   23.119  220.986
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.89852   18.51621   0.481    0.632
## liv_area     0.11325    0.01207   9.386 2.61e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56.91 on 98 degrees of freedom
## Multiple R-squared:  0.4734, Adjusted R-squared:  0.468
## F-statistic: 88.09 on 1 and 98 DF,  p-value: 2.606e-15

class(my_sahp_train$kit_qual)

## [1] "character"

my_sahp_train$kit_qual <- factor(my_sahp_train$kit_qual,
                       levels=c(
                         "Fair",
                         "Average",
                         "Good",
                         "Excellent"
                       ))
class(my_sahp_train$kit_qual)

## [1] "factor"

fit_sale_price_kit_qual <-
  lm(sale_price~kit_qual, data= my_sahp_train)
summary(fit_sale_price_kit_qual)

##
## Call:
## lm(formula = sale_price ~ kit_qual, data = my_sahp_train)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -94.926 -27.928  -6.065  19.990 198.298
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         117.80      28.69   4.106 8.45e-05 ***
## kit_qualAverage      19.52      29.46   0.662   0.5093
## kit_qualGood         76.13      29.96   2.541   0.0127 *
## kit_qualExcellent   229.13      33.13   6.917 5.14e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.69 on 96 degrees of freedom
## Multiple R-squared:  0.6068, Adjusted R-squared:  0.5945
## F-statistic: 49.37 on 3 and 96 DF,  p-value: < 2.2e-16
```

- **Answer (Q1, a):**

- The **coefficient** of **gar_car** is **60.908** and the R^2 value is **0.321**. This means that for every unit change in **gar_car**, **sale_price** increases by **$60,908.00**. The R^2 of **fit_sale_price_gar_car** indicates that **32.10%** of the variation of the sale price can be explained by the model using the variable **gar_car**.

- The **coefficient** of **liv_area** is **0.11325** and the R^2 value is **0.4734**. This means that for every unit change in **liv_area**, **sale_price** increases by **$113.25**. The R^2 of **fit_sale_price_liv_area** indicates that **47.34%** of the variation of the sale price can be explained by the model using the variable **liv_area**.

- For **my_sahp_train$kit_qual**, I checked it's class with **class()** and found that it was **character**. I converted it to factor with **factor()** and set the levels as **Fair, Average, Good, and Excellent**, which makes **Fair** the reference level. Therefore, the **coefficients** of **kit_qual** are **117.80** for Fair (**the reference level** or **intercept**), **19.52** for "***Average***", **76.13** for "***Good***" and **229.13** for "***Excellent***". The R^2 value is **0.6068**. This means that when all levels other than the reference levels are false (**Average**, **Good**, **Excellent**), that the observation is **Fair** and has a predicted sale price of **$117,800**. When **Average** is **True** and all other observations are **False** the predicted sale price increases by **$19,520**. When **Good** is **True** and all other observations are **False** the predicted sale price increases by **$76,130**. When **Excellent** is **True** and all other observations are **False** the predicted sale price increases by **$229,130**. The R^2 is interpreted as **60.68%** of the variation in predicted sale price is attributed to **kit_qual**.

- Given that **kit_qual** has the highest R^2 value, **kit_qual** is the variable **most useful in predicting the 'sale_price'** within the training data.

b. Compute the fitted value for the training data and make prediction for the test data, then compute the training and test MSE. Which variable gives the smallest test

MSE? Does this agree with the variable with the highest R^2? Explain your findings.

Note that, the training MSE is defined as

$$\frac{1}{n_{train}} \sum_{i \in train} \left(Y_i - \hat{Y}_i\right)^2$$

and the test MSE is defined as

$$\frac{1}{n_{test}} \sum_{i \in test} \left(Y_i - \hat{Y}_i\right)^2,$$

where n_{train} and n_{test} represent the number of observations for the training and test data, repectively.

Solution:

```
#MSE train & Test for gar_car
pred_gar_car_train <- predict(
  fit_sale_price_gar_car,
    newdata = my_sahp_train
)

pred_gar_car_test <- predict(
  fit_sale_price_gar_car,
  newdata = my_sahp_test
)

mse_gar_car_train <- mean((
  pred_gar_car_train - my_sahp_train$sale_price)^2)

mse_gar_car_test <- mean((
  pred_gar_car_test - my_sahp_test$sale_price)^2)

mse_gar_car_train
```

```
## [1] 4092.45
```

```
mse_gar_car_test
```

```
## [1] 4718.267
```

```
#MSE train & Test for liv_area
pred_liv_area_train <- predict(
  fit_sale_price_liv_area,
  newdata = my_sahp_train
)
```

```r
pred_liv_area_test <- predict(
  fit_sale_price_liv_area,
  newdata = my_sahp_test
)

mse_liv_area_tain <- mean((
  pred_liv_area_train -
my_sahp_train$sale_price)^2)

mse_live_area_test <- mean((
  pred_liv_area_test -
my_sahp_test$sale_price)^2)

mse_liv_area_tain
```

```
## [1] 3174.156
```

```r
mse_live_area_test
```

```
## [1] 4182.664
```

```r
#MSE train & Test for kit_qual
pred_kit_qual_train <- predict(
  fit_sale_price_kit_qual,
newdata = my_sahp_train
)

pred_kit_qual_test <- predict(
  fit_sale_price_kit_qual,
newdat = my_sahp_test
)

mse_kit_qual_train <- mean((
  pred_kit_qual_train -
my_sahp_train$sale_price)^2)

mse_kit_qual_test <- mean((
  pred_kit_qual_test -
my_sahp_test$sale_price)^2)

mse_kit_qual_train
```

```
## [1] 2370.283
```

```r
mse_kit_qual_test
```

```
## [1] 4839.717
```

- **Answer (Q1, b):**

- The variable with the **smallest test MSE** is **liv_area** with an **mse_test** of **4182.66**. The variable with the highest R^2 is **kit_qual** of **0.6068**. This does not agree with my findings that **kit_qual** is the most useful in predicting **sale_price**. It goes to show that a variable that fits the training data well doesn't necessarily generalize unseen (test) data well. While R^2 does a good job of explaining the percentage of variation attributed to a variable given some training data, **mse_test** is a better determinant of **usefulness** of a variable because it measures the differences between predicted values of unseen values and their true values, something that directly correlates with true predictive capability.

Q2

Use the training data `my_sahp_train` to fit a multiple linear regression model of `sale_price` on all variables, interpret the coefficients and compute the R^2. Then compute the training and test MSE. Compare the results to Q1 and explain your findings.

Solution:

```
fit_sale_price_GLK <- lm(
  sale_price ~ gar_car + liv_area +
    kit_qual, data = my_sahp_train)

summary(fit_sale_price_GLK)

##
## Call:
## lm(formula = sale_price ~ gar_car + liv_area + kit_qual, data = my_sahp_tr
ain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -85.872 -21.532  -0.614  17.439 120.635
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -6.168132  25.838420  -0.239   0.8118
## gar_car            27.559749   5.830365   4.727 7.98e-06 ***
## liv_area            0.057576   0.009551   6.028 3.23e-08 ***
## kit_qualAverage    29.978452  22.164214   1.353   0.1794
## kit_qualGood       51.618408  22.686602   2.275   0.0252 *
## kit_qualExcellent 171.607278  25.780827   6.656 1.86e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.32 on 94 degrees of freedom
## Multiple R-squared:  0.7827, Adjusted R-squared:  0.7712
## F-statistic: 67.73 on 5 and 94 DF,  p-value: < 2.2e-16
```

- **Answer (Q2):** The coefficients of **fit_sale_price_GLK** (**model using all 3 chosen variables**) are **27.559749** for **gar_car**, **0.057576** for **liv_area**, and **29.978452, 51.618408, and 171.607278** for **kit_qualAverage, kit_qualGood, and kit_qualExcellent** respectively. The factor level **kit_qualFair** doens't have a coefficient, however there is an intercept of **-6.16813** that isn't negated with an increase in value of **sale_price** given a factor level of **kit_qualFair**. This means that for every unit increase (holding everything else equal) of **gar_car**, house price increases by **$27,559.74**... **$57.57** for every unit increase of **liv_area**, and **$29,978.45, $51,618.40, and $171,607.27** if either **kit_qualAverage, kit_qualGood, and kit_qualExcellent** are ***True*** given all else are ***False***. However, the intercept of **(-6.16813)** is nonsensical since the saleprice cannot be negative, and therefore if the factor level of **kit_qual** is **fair** while every other factor is 0 and every other variable is 0, the linear model gives a nonsensical prediction that sale price would be **(-$6,168.13)**.

- The R^2 for **fit_sale_price_GLK** is **0.7827**.

```
pred_GLK_train <- predict(
  fit_sale_price_GLK, newdata =
    my_sahp_train)

pred_GLK_test <- predict(
  fit_sale_price_GLK, newdata =
    my_sahp_test)

mse_GLK_train <- mean((
  pred_GLK_train -
my_sahp_train$sale_price)^2)

mse_GLK_test <- mean((
  pred_GLK_test -
my_sahp_test$sale_price)^2)

mse_GLK_train
```

```
## [1] 1309.529
```

```
mse_GLK_test
```

```
## [1] 2535.262
```

- **Answer (Q2) continued:** The **training MSE** for **fit_sale_price_GLK** is **1309.529** and it's **test MSE** is **2535.262**.

- **Compared to question 1** (training and test MSE of models of the 3 variables individually), the **training and test MSE** of the model of all 3 variables together is substantially lower. This indicates that using all 3 predictors variable in the same

model **gives a better fit and stronger predictive power** compared to the models that use only 1 variable each.

- Additionally, the R^2 value of **0.7827** is higher than any other individual R^2 value of any model using only 1 predictor. This means that **78.27%** of the variation in sale price is predicted by the model with 3 predictors, higher than the best single variable model of **kit_qual** that had an R^2 value of **0.6068**.

- Of course, the **test MSE** for this 3 predictor model is **larger than it's training MSE**, but the increase is **not as large as** it was for some of the **single predictor models**.

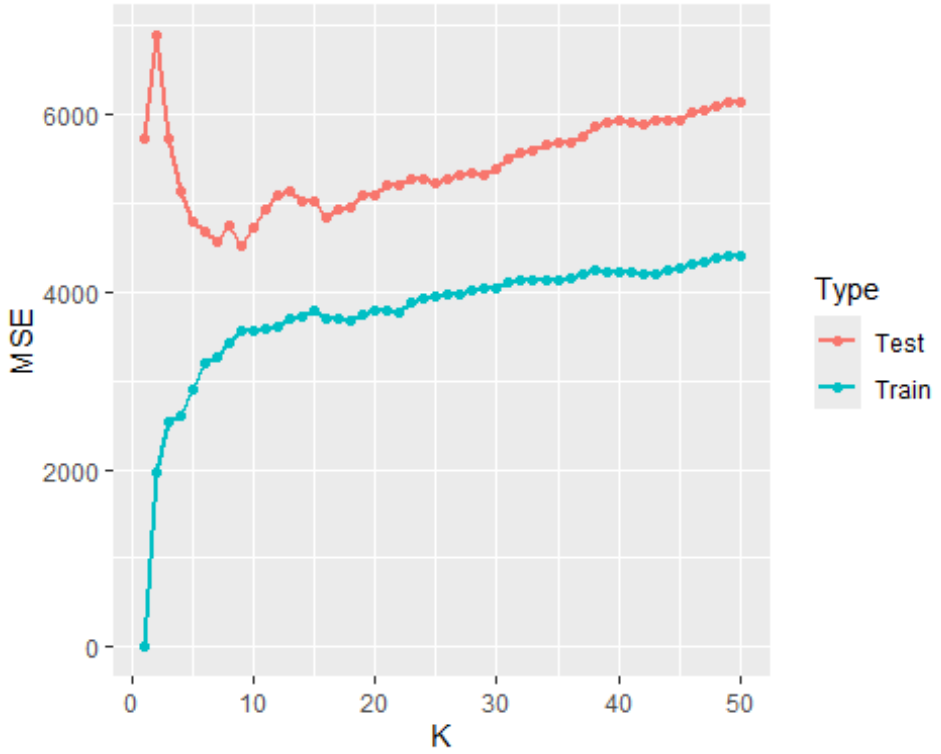
Q3

Now, use the KNN method for predicting `sale_price` using all predictors. Note: Please use the **formula format** for KNN regression, i.e., `knnreg(formula, data, k)`, so that R will automatically code `kit_qual` as dummy variables.

a. Vary the number of neighbors K from 1 to 50 with increment 1. For each K, fit the KNN regression model on the training data, and predict on the test data. Visualize the trends of training and test MSEs as functions of K (see slide #40 of Lecture 2). Discuss your findings.

```
library(caret)

## Warning: package 'caret' was built under R version 4.4.3

library(ggplot2)

k_seq <- 1:50
mse_seq_tr <- mse_seq_te <- NULL

for (i in seq_along(k_seq)) {
  fit <- knnreg(
    sale_price ~ gar_car +
      liv_area +
      kit_qual,
    data = my_sahp_train,
    k = k_seq[i]
  )

y_hat_tr <- predict(fit,
                    newdata = my_sahp_train)
mse_seq_tr[i] <- mean((
  my_sahp_train$sale_price - y_hat_tr)^2)

y_hat_te <- predict(fit, newdata =
    my_sahp_test)
mse_seq_te[i] <- mean((
  my_sahp_test$sale_price - y_hat_te)^2)
```

```
}

mse_stacked <- rbind(
  data.frame(K = k_seq,
  MSE = mse_seq_tr, Type = "Train"),
  data.frame(K = k_seq,
  MSE = mse_seq_te, Type = "Test")
)

ggplot(mse_stacked, mapping = aes(x = K, y = MSE, color = Type)) +
  geom_point() +
  geom_line(linewidth = 1)
```




```
which.min(mse_seq_te)

## [1] 9

mse_seq_tr[9]

## [1] 3566.076

mse_seq_te[9]

## [1] 4521.596
```

b. Compare the best KNN result with the linear regression result in Q2. Discuss your findings.

- **Answer (Q3, a & b):** The best number of **neighbors** to use, or **K**, is **"9"**. The **train MSE** equals **3566.076** when **K = 9** and the **test MSE** equals **4521.596** when **K = 9**. Compared to the **linear model** in question 2, the test MSE for this model is larger, which means that **KNN performs worse here** and therefore a **linear model generalizes better** for this dataset.

Q4

Answer the following questions on a sheet of paper, scan or take a photo of your answer, and upload as an attachment in your homework submission. (R is not needed for these questions)

a. Does the line $y = 3x - 5$ pass through the point $(3,4)$? Why?

b. ISLR book 2nd Edition Chapter 3.7 Question 6:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})\,(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Using the above equations, argue that in the case of simple linear regression, the least squares line always passes through the point $(\bar{x}, \bar{y})$.

Solution: