rocchio-assign3

July 11, 2023

1 Part 1

[8]:
```
import pandas as pd
import numpy as np

df=pd.read_excel('ames_housing_data.xlsx')
len(df)
```

[8]: 2930

1.1 Model 1 Question 1

There are 2930 observations in the sample data.

1.2 Model 1 Question 2

1.2.1 Null Hypothesis (H0):

The coefficient Beta1 equals zero, meaning X1 has no effect on the dependent variable. Mathematically, this can be written as H0: Beta1 = 0.

1.2.2 Alternative Hypothesis (Ha):

The coefficient Beta1 does not equal zero, meaning X1 does have an effect on the dependent variable. Mathematically, this can be written as Ha: Beta1 0.

1.2.3 Summary:

In this case, since the p-value associated with the t-value for Beta1 is less than 0.0001, we can reject the null hypothesis and accept the alternate hypothesis. This means we have sufficient evidence to conclude that the predictor variable X1 does have a statistically significant effect on the dependent variable.

1.3 Model 1 Question 3

$$t = (Estimate - NullHypothesis)/Std.Error$$

The estimate for Beta1 is given as 2.186, the standard error is 0.4104, and the null hypothesis is that Beta1 equals 0.

Inputting these values into the equation yields:

$$t = (2.186 - 0)/0.4104 \approx 5.33$$

The calculated t-value matches with the one in the given model summary.

Generally is the absolute val of the t-value is greater than 2, it's typically considered significant at the p < 0.05 level. Here, the t-value is approximately 5.33, so it is significant.

In this case, we also have the p-value directly given as <0.0001 which is much less than the usual alpha level of 0.05.

This means that you would reject the null hypothesis and conclude that the variable X1 has a statistically significant effect on the response variable. This means that changes in X1 are associated with changes in the dependent variable.

1.4 Model 1 Question 4

R-squared can be computed using the ANOVA table by dividing the sum of squares of the regression (Model sum of squares) by the total sum of squares:

$$R^2 = \frac{SSR}{SST}$$

From the provided ANOVA table:

$$Let\ SSR = 2126$$

$$Let\ SST = 2756.37$$

$$\frac{2126}{2756.37} \approx 0.7713$$

This matches the provided R-squared value, suggesting that approximately 77.13% of the variance in the dependent variable can be explained by the model.

This means that 77.13% of the variability in the outcome variable can be accounted for by the variables X1, X2, X3, and X4 in your model. This is a relatively high value, indicating that your model provides a good fit to your data.

[9]:
```
## Rounding to match values of provided output.
print(round(2126/2756.37, 4))
```

```
0.7713
```

1.5 Model 1 Question 5

The Adjusted R-squared can be calculated using the following formula:

$$Let\ n = total\ number\ of\ observations$$

$$Let\ k = number\ of\ predictors$$

$$Adj.\ R^2 = 1 - \frac{(1-R^2)(n-1)}{(n-k-1)}$$

From the provided table:

$$n = 72\ (dof + 1)$$

$$k = 4$$

$$R^2 = 0.7713$$

$$Adj.\ R^2 = 1 - \frac{(1-0.7713)(72-1)}{(72-4-1)} \approx 0.7576$$

This matches the given Adjusted R-squared value, confirming the calculation.

R-squared and Adjusted R-squared values are different because Adjusted R-squared accounts for the number of predictors in the model. If you add more and more useless variables to a model, R-squared will tend to increase. However, Adjusted R-squared will penalize you for adding ineffective predictors, leading to a decrease if the additional variables do not contribute enough to the improvement of the model.

[10]:
```
## Rounding to match values of provided output.
print(round(1-((1-.7713)*(72-1)/(72-4-1)),4))
```

```
0.7576
```

1.6 Model 1 Question 6

The null and alternate hypotheses for the overall F-test are:

Null Hypothesis (H0): All the regression coefficients are equal to zero. In the context of your model, this means that none of the predictor variables (X1, X2, X3, X4) have any effect on the dependent variable.

This can be written as:

$$H0: \ Beta1 \ = Beta2 \ = Beta3 \ = Beta4 \ = 0$$

Alternative Hypothesis (Ha): At least one regression coefficient is not equal to zero. This means that at least one predictor variable has a significant effect on the dependent variable. This can be written as:

$$Ha: \ At\ least\ one\ Beta\ i \neq 0$$

$$ Where i represents the predictor variables X1, X2, X3, X4.

1.7 Model 1 Question 7

The F-statistic can be calculated using the following formula which has been simplified for use in this scenario:

$$Let\ SSR = sum\ of\ squares\ due\ to\ regression$$
$$Let\ SSE = sum\ of\ squares\ due\ to\ error$$
$$Let\ n = total\ number\ of\ observations$$
$$Let\ k = number\ of\ predictors$$
$$F = \frac{(SSR/k) * (n - k - 1)}{SSE}$$

From the ANOVA table:

$$SSR = 2126$$
$$SSE = 630.36$$
$$n = 72$$
$$k = 4$$

$$F = \frac{(2126/4) * (72 - 4 - 1)}{630.36} \approx 56.49$$

If the p-value associated with this F-statistic is less than our significance level (typically 0.05), we reject the null hypothesis.

The p-value provided for the F-test is < 0.0001, which is less than 0.05.

Therefore, we reject the null hypothesis that all regression coefficients are equal to zero. We conclude that at least one predictor (X1, X2, X3, X4) has a significant effect on the dependent variable. This indicates that the model with the predictors (X1, X2, X3, X4) is statistically significantly better at explaining the variance in the dependent variable than an empty model.

[11]:
```
## Rounding to match values of providedround( output.
print(round((2126/4)*(72-4-1)/(630.36),2))
```

```
56.49
```

1.8 Model 2 Question 8

Comparing Model 1 and Model 2:

Model 1 includes predictors X1, X2, X3, X4. Model 2 includes predictors X1, X2, X3, X4, but also additional predictors X5 and X6.

Therefore:

$$Model1 \subset Model2$$

Since Model 1 is part of Model 2 and you can obtain Model 2 by adding X5 and X6 to Model 1, Model 1 nests Model 2.

1.9 Model 2 Question 9

Null Hypothesis (H0): The reduced model (Model 1) is adequate, and the additional predictors in the full model (Model 2) do not significantly improve the fit. This implies that the coefficients of the additional predictors (X5 and X6) in Model 2 are zero.

Alternative Hypothesis (Ha): The full model (Model 2) provides a significantly better fit than the reduced model (Model 1). This means that at least one of the additional predictors in Model 2 (X5 or X6) has a significant effect on the dependent variable, meaning that at least one of their coefficients is not equal to zero.

This can be represented by:

$$H0 : Beta5 = Beta6 = 0 \ (coefficients\ of\ X5\ and\ X6\ in\ Model\ 2)$$

$$Ha : \ At\ least\ one\ of\ Beta5\ and\ Beta6\ is\ not\ equal\ to\ zero.$$

1.10 Model 2 Question 10:

The F-statistic for a nested F-test can be computed using the simplified following formula:

$$Let\ SSE1 = squares\ of\ the\ residuals\ of\ the\ smaller\ model\ (Model\ 1)$$

$$SSE2 = squares\ of\ the\ residuals\ of\ the\ larger\ model\ (Model\ 2)$$

$$DF1 = dof\ of\ the\ smaller\ model\ (Number\ obs.\ minus\ the\ number\ of\ predictors\ in\ model\ 1 - 1)$$

$$DF2 = dof\ of\ the\ larger\ model\ (Number\ obs.\ minus\ the\ number\ of\ predictors\ in\ model\ 2 - 1)$$

$$F = \frac{SSE1 - SSE2/(DF1 - DF2)}{(SSE2/DF2)}$$

Input with the values provided:

$$SSE1 = 630.36$$

$$SSE2 = 572.6091$$

$$DF1 = 71 - 4 - 1 = 66$$

$$DF2 = 71 - 6 - 1 = 64$$

$$F = \frac{630.36 - 572.6091}{(66 - 64)/(572.6091/64)} \approx 3.23$$

[12]:
```
print(round((64 * (630.36 - 572.6091))/(572.6091 * (66 - 64)),2))
```

```
3.23
```

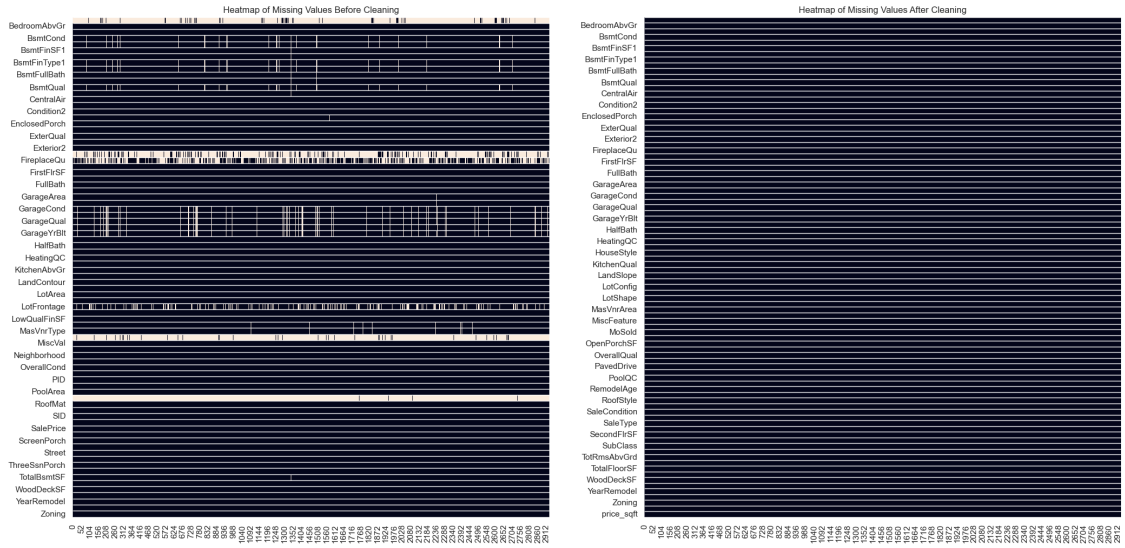
2 Part 2

2.1 Prep Work

2.1.1 Adding in data quality work from assignment 1.

[13]:
```
import seaborn as sns
import matplotlib.pyplot as plt


##
sns.set()
fig, (ax1, ax2) = plt.subplots(1,2,figsize=(25, 12))

df_heat1=df.sort_index(axis=1, ascending=False)
sns.heatmap(df_heat1.T.isnull(), ax=ax1, cbar=False).invert_yaxis()
ax1.hlines(range(len(df_heat1)), *ax1.get_xlim(), color='white', linewidths=1)
ax1.vlines([], [], [])
ax1.set_title('Heatmap of Missing Values Before Cleaning')
plt.yticks(rotation = 360)
df['TotalFloorSF'] = df['FirstFlrSF'] + df['SecondFlrSF']
df['HouseAge'] = df['YrSold'] - df['YearBuilt']
df['QualityIndex'] = df['OverallQual'] * df['OverallCond']
df['logSalePrice'] = np.log(df['SalePrice'])
df['price_sqft'] = df['SalePrice'] / df['TotalFloorSF']
Nulls=[]
for i in df.columns:
    if df[i].isnull().sum() > 0:
        Nulls.append(i)
df['LotFrontage']=df['LotFrontage'].fillna(df['LotFrontage'].median())
df['Alley']=df['Alley'].fillna('No alley')
df['MasVnrType']=df['MasVnrType'].fillna('None')
df['MasVnrArea']=df['MasVnrArea'].fillna(0)
df['RemodelAge']=df['YrSold']-df['YearRemodel']
for col in ['BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1',␣
 ↪'BsmtFinType2']:
    df[col].fillna('No basement')
for col in ['GarageType', 'GarageFinish', 'GarageQual', 'GarageCond']:
    if df[col].dtype == 'object':
        df[col]=df[col].fillna('No garage')
    else:
        df[col]=df[col].fillna('None')
df['GarageYrBlt']=df['GarageYrBlt'].fillna(df['GarageYrBlt'].median())
df['GarageCars']=df['GarageCars'].fillna(0)
df['GarageArea']=df['GarageArea'].fillna(0)
df['PoolQC']=df['PoolQC'].fillna('No pool')
df['Fence']=df['Fence'].fillna('No fence')
df['MiscFeature']=df['MiscFeature'].fillna('No feature')
```

```
df['Electrical']=df['Electrical'].fillna(df['Electrical'].mode()[0])
df['FireplaceQu']=df['FireplaceQu'].fillna('No fireplace')
for col in ['BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1',␣
 ↪'BsmtFinType2']:
    df[col]=df[col].fillna('No basement')
for col in ['BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF',␣
 ↪'BsmtFullBath', 'BsmtHalfBath']:
    df[col]=df[col].fillna(0)
df.to_csv('ames_housing_data.csv', index=False)
sns.set()
df_heat2=df.sort_index(axis=1, ascending=False)
sns.heatmap(df_heat2.T.isnull(), ax=ax2, cbar=False).invert_yaxis()
ax2.hlines(range(len(df_heat2)), *ax2.get_xlim(), color='white', linewidths=1)
ax2.set_title('Heatmap of Missing Values After Cleaning')
ax2.vlines([], [], [])
plt.yticks(rotation = 360)
plt.show()
```



2.2 Variable Selection

[ ]:
```
### Before I switch to R I am going to write code to do a stepwise comparison␣
 ↪of the variables..
### Sorry to use Python but I dont know how to write something this complicated␣
 ↪in R without spending hours on it.
### This will find the best variables to use in the model and count for␣
 ↪colinearity... most likely.

import warnings
```

```
from sklearn.datasets import load_boston
import statsmodels.api as sm
warnings.filterwarnings('ignore')

def get_continuous_variables(dataset, target_except):
    continuous_vars = []
    for column in dataset.columns:
        if column not in target_except:
            if dataset[column].dtype in [int, float]:
                continuous_vars.append(column)
    return continuous_vars

def stepwise_selection(X, y,
                       initial_list=[],
                       threshold_in=0.01,
                       threshold_out = 0.05,
                       max_vars=10,
                       verbose=True):
    included = list(initial_list)
    while True:
        changed=False
        excluded = list(set(X.columns)-set(included))
        new_pval = pd.Series(index=excluded)
        for new_column in excluded:
            model = sm.OLS(y, sm.add_constant(pd.
 ↪DataFrame(X[included+[new_column]]))).fit()
            new_pval[new_column] = model.pvalues[new_column]
        best_pval = new_pval.min()
        if best_pval < threshold_in:
            best_feature = new_pval.idxmin()
            included.append(best_feature)
            changed=True
            if verbose:
                print('Add  {:30} with p-value {:.6}'.format(best_feature,␣
 ↪best_pval))
        model = sm.OLS(y, sm.add_constant(pd.DataFrame(X[included]))).fit()
        pvalues = model.pvalues.iloc[1:]
        worst_pval = pvalues.max()
        if worst_pval > threshold_out:
            changed=True
            worst_feature = pvalues.idxmax()
            included.remove(worst_feature)
            if verbose:
                print('Drop {:30} with p-value {:.6}'.format(worst_feature,␣
 ↪worst_pval))
        if not changed:
            break
```

```
    final_model = sm.OLS(y, sm.add_constant(pd.DataFrame(X[included]))).fit()
    coef = final_model.params.iloc[1:]
    coef_abs = np.abs(coef)
    top_vars = coef_abs.nlargest(max_vars).index.tolist()
    return top_vars
```

[8]:
```
numeric_vars = get_continuous_variables(df, ['SalePrice', 'logSalePrice',␣
 ↪'price_sqft'])
print(numeric_vars)
X = df[numeric_vars]
y = df['SalePrice']
best_variables10 = stepwise_selection(X, y, max_vars=10)
```

```
['SID', 'PID', 'SubClass', 'LotFrontage', 'LotArea', 'OverallQual',
'OverallCond', 'YearBuilt', 'YearRemodel', 'MasVnrArea', 'BsmtFinSF1',
'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'FirstFlrSF', 'SecondFlrSF',
'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'TotRmsAbvGrd', 'Fireplaces',
'GarageYrBlt', 'GarageCars', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF',
'EnclosedPorch', 'ThreeSsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal',
'MoSold', 'YrSold', 'TotalFloorSF', 'HouseAge', 'QualityIndex', 'RemodelAge']
Add  OverallQual                  with p-value 0.0
Add  TotalFloorSF                 with p-value 1.05506e-196
Add  BsmtFinSF1                   with p-value 2.01534e-99
Add  GarageArea                   with p-value 6.46371e-48
Add  SubClass                     with p-value 2.96896e-32
Add  HouseAge                     with p-value 5.24779e-32
Add  BedroomAbvGr                 with p-value 2.48733e-16
Add  MiscVal                      with p-value 1.59602e-15
Add  MasVnrArea                   with p-value 2.70263e-14
Add  RemodelAge                   with p-value 1.45905e-14
Add  LotArea                      with p-value 1.28436e-10
Add  ScreenPorch                  with p-value 1.58393e-07
Add  TotalBsmtSF                  with p-value 4.7736e-07
Add  OverallCond                  with p-value 1.81085e-08
Add  BsmtFullBath                 with p-value 2.54932e-06
Add  WoodDeckSF                   with p-value 0.000110822
Add  TotRmsAbvGrd                 with p-value 0.000180604
Add  Fireplaces                   with p-value 0.000477301
Add  GarageYrBlt                  with p-value 0.000532192
Add  KitchenAbvGr                 with p-value 0.00112476
Add  GarageCars                   with p-value 0.00251997
Drop GarageArea                   with p-value 0.0793156
```

[9]:
```
print("Best 10 Variables: ")
for i in best_variables10:
    print(i)
```

```
Best 10 Variables:
OverallQual
KitchenAbvGr
BedroomAbvGr
GarageCars
BsmtFullBath
OverallCond
Fireplaces
TotRmsAbvGrd
HouseAge
RemodelAge
```

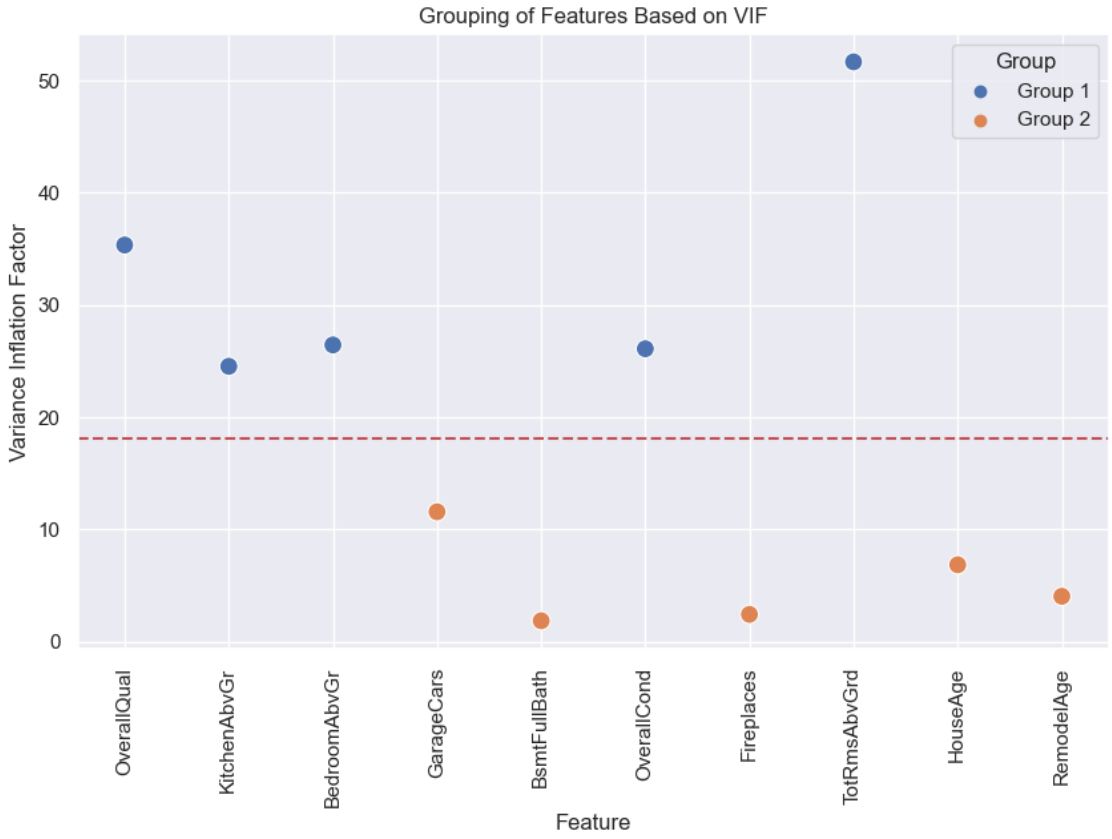
2.2.1 Variable Grouping:

I would use K means clustering to group variables into 2 or more sets. But I will cluster on variance inflation factor to eliminate collinearity. I then will divide the features into two groups, one with lower VIF values and one with higher VIF values, representing lower and higher multicollinearity.

[11]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.cluster import KMeans

def calculate_vif(X):
    vif = pd.DataFrame()
    vif["variables"] = X.columns
    vif["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.
 ↪shape[1])]
    return vif

def group_variables(X, selected_variables):
    vif = calculate_vif(X[selected_variables])
    vif = vif.set_index('variables').reindex(selected_variables)

    kmeans = KMeans(n_clusters=2, random_state=0).fit(vif.values.reshape(-1,1))
    labels = kmeans.labels_
    group_1 = [var for var, label in zip(selected_variables, labels) if label␣
 ↪== 0]
    group_2 = [var for var, label in zip(selected_variables, labels) if label␣
 ↪== 1]

    if len(group_1) < 2:
        group_1.append(group_2.pop())
    elif len(group_2) < 2:
        group_2.append(group_1.pop())

    return group_1, group_2, vif
```

```
group_1, group_2, vif = group_variables(X, best_variables10)

print('Group 1:')
print(group_1)

print('Group 2:')
print(group_2)

plot_data = pd.DataFrame({
    'Feature': best_variables10,
    'VIF': vif['VIF'],
    'Group': ['Group 1' if feature in group_1 else 'Group 2' for feature in␣
 ↪best_variables10]
})
plt.figure(figsize=(10, 6))
sns.scatterplot(data=plot_data, x='Feature', y='VIF', hue='Group', s=100)
plt.axhline(y=vif['VIF'].median(), color='r', linestyle='--')
plt.xticks(rotation=90)
plt.title('Grouping of Features Based on VIF')
plt.ylabel('Variance Inflation Factor')
plt.grid(True)

plt.show()
```

```
Group 1:
['OverallQual', 'KitchenAbvGr', 'BedroomAbvGr', 'OverallCond', 'TotRmsAbvGrd']
Group 2:
['GarageCars', 'BsmtFullBath', 'Fireplaces', 'HouseAge', 'RemodelAge']
```



[12]:
```
df.to_csv('ames_housing_data_clean.csv', index=False)
```

2.3 Switching to R for regression model fitting.

2.4 Model 3

[ ]:
```
library(tidyverse)
```

[13]:
```
df <- read.csv('ames_housing_data_clean.csv')
group1 <- c('OverallQual', 'KitchenAbvGr', 'BedroomAbvGr', 'OverallCond',␣
 ↪'TotRmsAbvGrd')
group2 <- c('GarageCars', 'BsmtFullBath', 'Fireplaces', 'HouseAge',␣
 ↪'RemodelAge')
full_group <- c(group1, group2)
head(df)
```

A data.frame: 6 × 88

	SID <int>	PID <int>	SubClass <int>	Zoning <chr>	LotFrontage <dbl>	LotArea <int>	Street <chr>	Alley <chr>
1	1	526301100	20	RL	141	31770	Pave	No alley
2	2	526350040	20	RH	80	11622	Pave	No alley
3	3	526351010	20	RL	81	14267	Pave	No alley
4	4	526353030	20	RL	93	11160	Pave	No alley
5	5	527105010	60	RL	74	13830	Pave	No alley
6	6	527105030	60	RL	78	9978	Pave	No alley

[14]:
```
model3 <- lm(SalePrice ~., data=df[, c("SalePrice", group1)])
summary(model3)
summary(model3)$coefficients[,4]
anova(model3)
```

```
Call:
lm(formula = SalePrice ~ ., data = df[, c("SalePrice", group1)])

Residuals:
    Min      1Q  Median      3Q     Max
-301314  -25729   -2799   21233  377430

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -87323.6     7811.8 -11.178  < 2e-16 ***
OverallQual   37399.3      687.2  54.425  < 2e-16 ***
KitchenAbvGr -33788.5     4217.7  -8.011 1.62e-15 ***
BedroomAbvGr -10087.6     1394.0  -7.236 5.86e-13 ***
OverallCond   -1224.9      744.2  -1.646   0.0999 .
TotRmsAbvGrd  17238.9      831.0  20.745  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 44250 on 2924 degrees of freedom
Multiple R-squared:  0.6937,        Adjusted R-squared:  0.6932
F-statistic:  1324 on 5 and 2924 DF,  p-value: < 2.2e-16
```

(Intercept) 1.946236164486e-28 **OverallQual** 0 **KitchenAbvGr** 1.62484462436099e-15 **BedroomAbvGr** 5.86261907837284e-13 **OverallCond** 0.0998788082977956 **TotRmsAbvGrd** 2.71550979153532e-89

A anova: 6 × 5

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
OverallQual	1	1.194116e+13	1.194116e+13	6098.183430	0.000000e+00
KitchenAbvGr	1	1.186188e+09	1.186188e+09	0.605770	4.364479e-01
BedroomAbvGr	1	1.672986e+11	1.672986e+11	85.437106	4.475482e-20
OverallCond	1	1.458339e+10	1.458339e+10	7.447536	6.390356e-03
TotRmsAbvGrd	1	8.426837e+11	8.426837e+11	430.346945	2.715510e-89
Residuals	2924	5.725630e+12	1.958150e+09	NA	NA

2.4.1 The model was constructed using the dependent variable SalePrice and the following explanatory variables:

OverallQual, KitchenAbvGr, BedroomAbvGr, OverallCond, TotRmsAbvGrd.

2.4.2 Coefficients:

These are the estimates for the parameters of the model. For example, the OverallQual coefficient of 37399.3 suggests that for every one-unit increase in OverallQual, the SalePrice increases by approximately $ 37,399.30, holding all other factors constant. The p-values (Pr(>|t|)) are all less than 0.05 for all variables except OverallCond, indicating that we reject the null hypothesis that the parameter equals zero for these variables. This suggests that these variables significantly affect SalePrice. The OverallCond variable is not statistically significant at the 0.05 level (p = 0.0999), so we fail to reject the null hypothesis for OverallCond.

2.4.3 Residual standard error:

This is the standard deviation of the residuals, which are the differences between the observed and predicted responses. It's estimated to be 44250 on 2924 degrees of freedom.

2.4.4 Multiple R-squared:

This is the proportion of variance in SalePrice that can be explained by the predictors. An R-squared of 0.6937 means that 69.37 % of the variation in SalePrice can be explained by the predictors in the model.

2.4.5 Adjusted R-squared:

This is the adjusted R-squared which accounts for the number of predictors in the model. This value is often more reliable than the R-squared when comparing models with different numbers of predictors. It's 0.6932, which is quite close to the Multiple R-squared.

2.4.6 F-statistic:

This is a statistic for an overall significance test that all the regression coefficients are zero. The F-statistic is 1324 on 5 and 2924 DF, and its corresponding p-value is less than 2.2e-16, which is practically zero. This means that we reject the null hypothesis that all regression coefficients are zero.

2.4.7 ANOVA Table:

The predictor KitchenAbvGr has a p-value of 0.436 which suggests that it's not a significant predictor at the 0.05 level when considering other variables in the model. Other variables have p-values close to or equal to zero, indicating that they're significant predictors in the model.

2.4.8 Conclusions:

OverallQual, KitchenAbvGr, BedroomAbvGr, and TotRmsAbvGrd are significant predictors for SalePrice, but OverallCond is not. The model explains about 69.37% of the variance in SalePrice, and the model significantly improves the prediction of SalePrice over a single variable model such as the one trained in assignment 2.

2.5 Model 4

[15]:
```
model4 <- lm(SalePrice ~., data=df[, c("SalePrice", full_group)])
summary(model4)
summary(model4)$coefficients[,4]
anova(model4)
```

```
Call:
lm(formula = SalePrice ~ ., data = df[, c("SalePrice", full_group)])

Residuals:
    Min      1Q  Median      3Q     Max
-285963  -22592   -3447   16850  390794

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -60043.74    7733.49  -7.764 1.13e-14 ***
OverallQual   24293.80     807.74  30.076  < 2e-16 ***
KitchenAbvGr -28408.63    3762.59  -7.550 5.77e-14 ***
BedroomAbvGr  -4743.36    1250.96  -3.792 0.000153 ***
OverallCond    2127.06     765.66   2.778 0.005503 **
TotRmsAbvGrd  14245.39     778.50  18.299  < 2e-16 ***
GarageCars    16787.23    1274.68  13.170  < 2e-16 ***
BsmtFullBath  20233.89    1441.85  14.033  < 2e-16 ***
Fireplaces    14596.36    1276.68  11.433  < 2e-16 ***
HouseAge       -259.69      39.64  -6.551 6.72e-11 ***
RemodelAge     -225.07      49.93  -4.507 6.82e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 38890 on 2919 degrees of freedom
Multiple R-squared:  0.7639,        Adjusted R-squared:  0.7631
F-statistic: 944.2 on 10 and 2919 DF,  p-value: < 2.2e-16
```

(Intercept) 1.1289960097964e-14 **OverallQual** 2.36699765308556e-173 **KitchenAbvGr** 5.77166327214362e-14 **BedroomAbvGr** 0.000152596301320088 **OverallCond** 0.00550349979340394 **TotRmsAbvGrd** 6.74427407298064e-71 **GarageCars** 1.6074495806682e-38 **BsmtFullBath** 2.42201760670655e-43 **Fireplaces** 1.210138159316 65e-29 **HouseAge** 6.71511993456806e-11 **RemodelAge** 6.81784783675497e-06

A anova: 11 × 5

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
OverallQual	1	1.194116e+13	1.194116e+13	7896.6649396	0.000000e+00
KitchenAbvGr	1	1.186188e+09	1.186188e+09	0.7844242	3.758644e-01
BedroomAbvGr	1	1.672986e+11	1.672986e+11	110.6342914	2.015354e-25
OverallCond	1	1.458339e+10	1.458339e+10	9.6439696	1.917976e-03
TotRmsAbvGrd	1	8.426837e+11	8.426837e+11	557.2652368	6.620825e-113
GarageCars	1	5.545508e+11	5.545508e+11	366.7234360	4.270372e-77
BsmtFullBath	1	4.238995e+11	4.238995e+11	280.3240022	3.742256e-60
Fireplaces	1	1.497466e+11	1.497466e+11	99.0271494	5.757428e-23
HouseAge	1	1.526641e+11	1.526641e+11	100.9564862	2.244828e-23
RemodelAge	1	3.072369e+10	3.072369e+10	20.3175244	6.817848e-06
Residuals	2919	4.414045e+12	1.512177e+09	NA	NA

[16]:
```
anova(model3, model4)
```

A anova: 2 × 6

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	2924	5.725630e+12	NA	NA	NA	NA
2	2919	4.414045e+12	5	1.311585e+12	173.4697	5.7662e-162

2.6 a) Hypothesis Test:

$$H0\ (Null\ Hypothesis) : \beta1 = \beta2 = \beta3 = \beta4 = \beta5 = 0$$

This signifies that the additional variables (GarageCars, BsmtFullBath, Fireplaces, HouseAge, RemodelAge) in Model 4 provide no improvement in predicting SalePrice over Model 3.

H1 (Alternative Hypothesis): At least one i 0 for i = 1, 2, 3, 4, 5 This signifies that at least one of the additional variables in Model 4 improves the prediction of SalePrice over Model 3.

2.7 Coefficent Hypothesis Tests:

2.7.1 OverallQual:

H0: There is no relationship between OverallQual and SalePrice. H1: There is a relationship. The p-value is near 0, thus we reject H0. There is strong evidence that OverallQual affects SalePrice.

2.7.2 KitchenAbvGr:

H0: There is no relationship between KitchenAbvGr and SalePrice. H1: There is a relationship. The p-value is near 0, thus we reject H0. There is strong evidence that KitchenAbvGr affects SalePrice.

2.7.3 BedroomAbvGr:

H0: There is no relationship between BedroomAbvGr and SalePrice. H1: There is a relationship. The p-value is 0.00015, thus we reject H0. There is strong evidence that BedroomAbvGr affects SalePrice.

2.7.4 OverallCond:

H0: There is no relationship between OverallCond and SalePrice. H1: There is a relationship. The p-value is 0.0055, thus we reject H0. There is strong evidence that OverallCond affects SalePrice.

2.7.5 TotRmsAbvGrd:

H0: There is no relationship between TotRmsAbvGrd and SalePrice. H1: There is a relationship. The p-value is near 0, thus we reject H0. There is strong evidence that TotRmsAbvGrd affects SalePrice.

2.7.6 GarageCars:

H0: There is no relationship between GarageCars and SalePrice. H1: There is a relationship. The p-value is near 0, thus we reject H0. There is strong evidence that GarageCars affects SalePrice.

2.7.7 BsmtFullBath:

H0: There is no relationship between BsmtFullBath and SalePrice. H1: There is a relationship. The p-value is near 0, thus we reject H0. There is strong evidence that BsmtFullBath affects SalePrice.

2.7.8 Fireplaces:

H0: There is no relationship between Fireplaces and SalePrice. H1: There is a relationship. The p-value is near 0, thus we reject H0. There is strong evidence that Fireplaces affects SalePrice.

2.7.9 HouseAge:

H0: There is no relationship between HouseAge and SalePrice. H1: There is a relationship. The p-value is near 0, thus we reject H0. There is strong evidence that HouseAge affects SalePrice.

2.7.10 RemodelAge:

H0: There is no relationship between RemodelAge and SalePrice. H1: There is a relationship. The p-value is near 0, thus we reject H0. There is strong evidence that RemodelAge affects SalePrice.

2.8 b) Omnibus Overall F-test

The null hypothesis (H0) for the overall F-test is that all regression coefficients (excluding the intercept) are zero, which implies that none of the predictors are related to the response, and the model does not improve the prediction of SalePrice over an intercept-only model. The alternative hypothesis (H1) is that at least one regression coefficient is not zero, which implies that at least one of the predictors is related to the response, and the model improves the prediction of SalePrice over an intercept-only model.

The F-statistic is 944.2 on 10 and 2919 degrees of freedom, and its associated p-value is virtually 0. As this p-value is less than 0.05, we reject the null hypothesis. There is strong evidence that at least one of the predictors has an improvement the prediction of SalePrice over an single variable model.

2.9 Nested Model

The null and alternative hypotheses for the nested F-test using Model 3 and Model 4 are:

H0 (Null Hypothesis): The additional variables in Model 4 (GarageCars, BsmtFullBath, Fireplaces, HouseAge, RemodelAge) provide no improvement in prediction of SalePrice over Model 3. This is written as:

$$\beta_{GarageCars} = \beta_{BsmtFullBath} = \beta_{Fireplaces} = \beta_{HouseAge} = \beta_{RemodelAge} = 0$$

H1 (Alternative Hypothesis): At least one of the additional variables in Model 4 improves the prediction of SalePrice over Model 3. This is written as:

$$\beta_{GarageCars}, \beta_{BsmtFullBath}, \beta_{Fireplaces}, \beta_{HouseAge}, \beta_{RemodelAge} \neq 0$$

The F-statistic for this nested F-test is 173.4697. This statistic measures how much the sum of squares of residuals (a measure of the discrepancy between the data and the estimation) is reduced when we go from Model 3 to Model 4.

The p-value associated with this F-statistic is virtually 0 (5.7662e-162). This p-value is much less than 0.05 (or any reasonable significance level). Therefore, we reject the null hypothesis.

In conclusion, there is very strong evidence that at least one of the additional variables included in Model 4 (GarageCars, BsmtFullBath, Fireplaces, HouseAge, RemodelAge) significantly improves the prediction of SalePrice over Model 3.

The nested F-test produces an F-statistic of 173.4697 and a p-value of 5.7662e-162.

Using a significance level ( ) of 0.05:

$$If\ F \leq \alpha,\ we\ fail\ to\ reject\ the\ null\ hypothesis\ (H0)$$

$$If\ F > \alpha, we\ reject\ the\ null\ hypothesis\ (H0)\ and\ accept\ the\ alternative\ hypothesis\ (H1)$$

Since our p-value (5.7662e-162) is much smaller than our chosen significance level of 0.05, we reject the null hypothesis (H0).

Mathematically, this can be represented as:

$$F = 173.4697$$

$$\alpha = 0.05$$

$$p - value = 5.7662e - 162$$

$$Since\ p - value\ < \alpha, we\ reject\ H0$$

So, we conclude that there is strong evidence to suggest that at least one of the additional variables in Model 4 (GarageCars, BsmtFullBath, Fireplaces, HouseAge, RemodelAge) significantly improves the prediction of SalePrice over Model 3.