CS510-Midterm Coding Project

Howard Nguyen

10/11/2021

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

Correlation Analysis

Loading dataset

This code is to predict housing prices

dataset from zillow datasets: 21,613 observations and 21 varialbes

```
data <- read.csv("housing_data.csv", header = TRUE)
head(data)
```

```
##           id            date   price bedrooms bathrooms sqft_living sqft_lot
## 1 7129300520 20141013T000000  221900        3      1.00        1180     5650
## 2 6414100192 20141209T000000  538000        3      2.25        2570     7242
## 3 5631500400 20150225T000000  180000        2      1.00         770    10000
## 4 2487200875 20141209T000000  604000        4      3.00        1960     5000
## 5 1954400510 20150218T000000  510000        3      2.00        1680     8080
## 6 7237550310 20140512T000000 1225000        4      4.50        5420   101930
##   floors waterfront view condition grade sqft_above sqft_basement yr_built
## 1      1          0    0         3     7       1180             0     1955
## 2      2          0    0         3     7       2170           400     1951
## 3      1          0    0         3     6        770             0     1933
## 4      1          0    0         5     7       1050           910     1965
## 5      1          0    0         3     8       1680             0     1987
## 6      1          0    0         3    11       3890          1530     2001
##   yr_renovated zipcode     lat     long sqft_living15 sqft_lot15
## 1            0   98178 47.5112 -122.257          1340       5650
## 2         1991   98125 47.7210 -122.319          1690       7639
## 3            0   98028 47.7379 -122.233          2720       8062
## 4            0   98136 47.5208 -122.393          1360       5000
## 5            0   98074 47.6168 -122.045          1800       7503
## 6            0   98053 47.6561 -122.005          4760     101930
```

Create scatter plots with house price data and see what kind of relationship we can quantify using the Pearson correlation.

Dependent variable: price

Independent variable: sqft_living

Create vectors with Y-dependent and X-independent

Scatterplot

```
plot(x,y, main="House price vs. Living space", xlab="Living space (sqft)",
     ylab="House price ($)", pch=18, cex=0.3, col="blue")
# Add a fit line to show the relationship direction
abline(lm(y~x))  # regression line (y~x)
lines(lowess(x,y), col="green")  # lowess line (x,y)
```

The plot shows the scatter plot bettwen Price and Living Space. The curved line is a locally smoothed fitted line. It can be seen that there is a linear relationship among the variables.

Report the correlation coefficient of this relation

```
cat("The correlation among House Price and Living Space is ", cor(x,y))
```

```
## The correlation among House Price and Living Space is  0.7020351
```

From the above plot, we can observe as follow:

The relationship is in a positive direction, so on average the house price increases with the size of the store. This is an intuitive relationship, hence we can draw causality. The bigger the living space, the better the house, which means it's more costly.

The correlation is 0.70. This is a pretty strong relationship on a linear scale.

The curved line is a LOWESS (Locally Weighted Scatterplot Smoothing) plot, which shows that it is not very different from the linear regression line. Hence, the linear relationship is worth exploring for a model.

Simple Linear Regression Analysis

Linear model using: Ordinary Least Square (OLS) technique, the lm()

Depenent variable: House price

Independent variable: Living space

Further our correlation analysis showed that these two variables have a positive linear relation and hence we will expect a positive sign to the parameter estimates of Living Space

```
# fit the model
fitted_model <- lm(y~x)
# display yhe summary of the model
summary(fitted_model)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1476062  -147486   -24043   106182  4362067
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -43580.743   4402.690  -9.899   <2e-16 ***
## x              280.624      1.936 144.920   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 261500 on 21611 degrees of freedom
## Multiple R-squared:  0.4929, Adjusted R-squared:  0.4928
## F-statistic: 2.1e+04 on 1 and 21611 DF,  p-value: < 2.2e-16
```

The estimated equation in this case is:

$y = 43580.743 + (280.624)x$

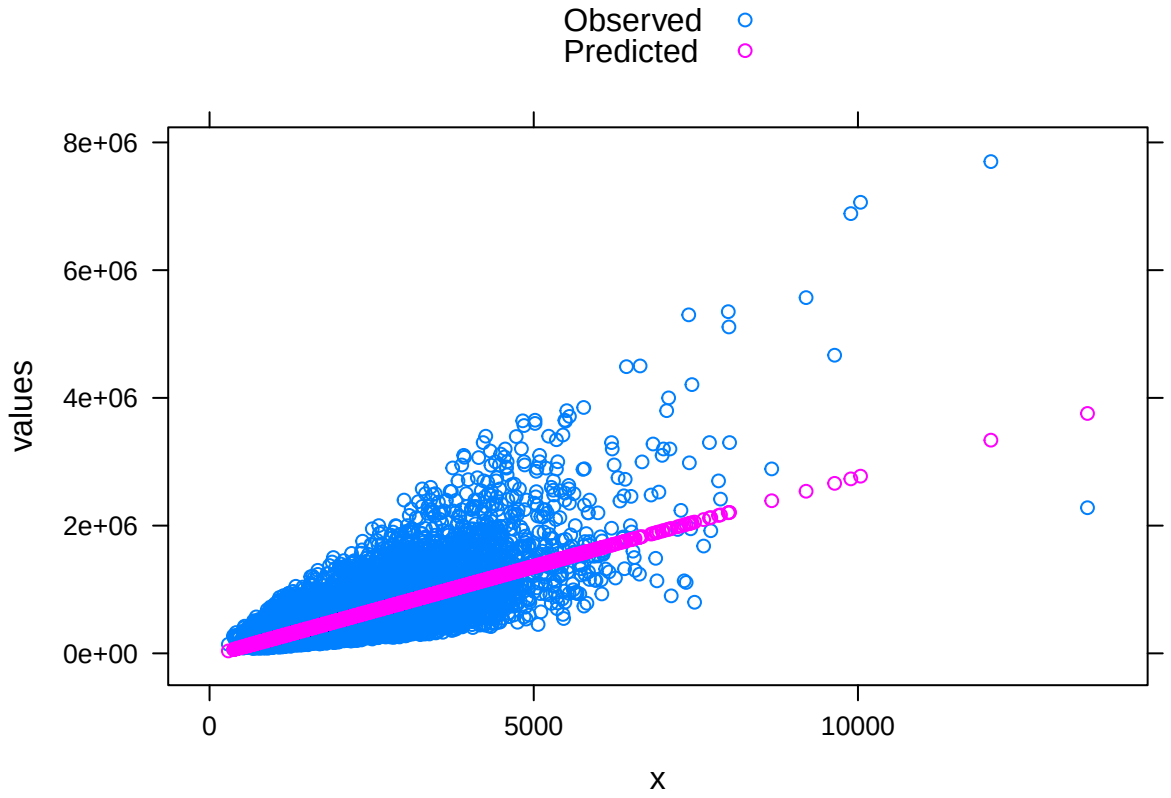
where y is House Price and x is Living Space. This implies for a unit increase in living space, the house price will be increased by $280.624.

Next, to see how the model fits the actual value, this is done by plotting actual values against the predicted values:

```
res <- stack(data.frame(Observed=y, Predicted=fitted(fitted_model)))
res <- cbind(res, x=rep(x, 2))
```

Plot using lattice xyplot function

```
library("lattice")
xyplot(values ~x, data=res, group=ind, auto.key=TRUE)
```



The above plot shows the fitted values with the actual values, we can see that the plot shows the linear relationship predicted by the model, stacked with the scatter plot of the original.

Now, this is a model with only one explanatory variable (sqft_living), but there are other variables show significant relationship with Price. The Regression framwork allow us to add multiple variable or independent variables to the regression analysis.

Multiple Linear Regression

Will use these variables: bedrooms, bathrooms, sqft_living, waterfront, view, condition, grade, and yr_built

```
lm_model <- data[,c("id","price","bedrooms","bathrooms","sqft_living",
                    "waterfront","view","condition","grade","yr_built")]
```

Check in for NA values

```
sapply(lm_model, function(x) sum(is.na(x)))
```

```
##          id       price    bedrooms   bathrooms sqft_living  waterfront
##           0           0           0           0           0           0
##        view   condition       grade    yr_built
##           0           0           0           0
```

In the case of any NA value, I use na.omit to remove these NA values off from the dataset for analysis

```
lm_model <- na.omit(lm_model)
rownames(lm_model) <- NULL
```

I need to factor those categorical variables: grade and condition

```
lm_model$grade <- factor(lm_model$grade)
lm_model$condition <- factor(lm_model$condition)
```

Now, the dataset is clean, I can run the lm() function to fit the multiple linear regression model.

```
fitted_model_multiple <- lm(price ~sqft_living + waterfront + bedrooms +
                              bathrooms + grade + condition, data = lm_model)
summary(fitted_model_multiple)
```

```
##
## Call:
```

```
## lm(formula = price ~ sqft_living + waterfront + bedrooms + bathrooms +
##     grade + condition, data = lm_model)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1518694  -123575   -20248    91564  3974525
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.564e+04  2.240e+05   0.427 0.669379
## sqft_living  1.599e+02  3.433e+00  46.567  < 2e-16 ***
## waterfront   7.526e+05  1.782e+04  42.225  < 2e-16 ***
## bedrooms    -1.919e+04  2.124e+03  -9.032  < 2e-16 ***
## bathrooms    2.032e+03  3.244e+03   0.626 0.531074
## grade3       2.523e+04  2.621e+05   0.096 0.923316
## grade4       5.787e+04  2.314e+05   0.250 0.802517
## grade5       2.478e+04  2.281e+05   0.109 0.913481
## grade6       6.292e+04  2.279e+05   0.276 0.782511
## grade7       1.017e+05  2.279e+05   0.446 0.655338
## grade8       1.720e+05  2.280e+05   0.755 0.450487
## grade9       3.019e+05  2.281e+05   1.324 0.185550
## grade10      4.883e+05  2.282e+05   2.140 0.032361 *
## grade11      7.610e+05  2.284e+05   3.331 0.000866 ***
## grade12      1.228e+06  2.295e+05   5.351 8.83e-08 ***
## grade13      2.531e+06  2.370e+05  10.680  < 2e-16 ***
## condition2  -3.889e+04  4.509e+04  -0.863 0.388395
## condition3  -3.945e+04  4.195e+04  -0.940 0.347060
## condition4   1.822e+04  4.198e+04   0.434 0.664353
## condition5   1.026e+05  4.222e+04   2.431 0.015073 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 224000 on 21593 degrees of freedom
## Multiple R-squared:  0.6281, Adjusted R-squared:  0.6278
## F-statistic:  1920 on 19 and 21593 DF,  p-value: < 2.2e-16
```

From the result, we can see that sqft_living, waterfront and bedrooms are significant at 95% confidence level, i.e., statistically different from zero. While many grades and conditions are insignificant, hence statistically they are equal zero. The higher gradings (11,12,13) are significant but not the lower ones. I will drop the condition and will re-estimate to keep only significant variables.

Now, to see the actual vs. predicted values for this model by plotting them after ordering the series by price.

Get the fitted values and create a data frame of actual and predicted get predicted values

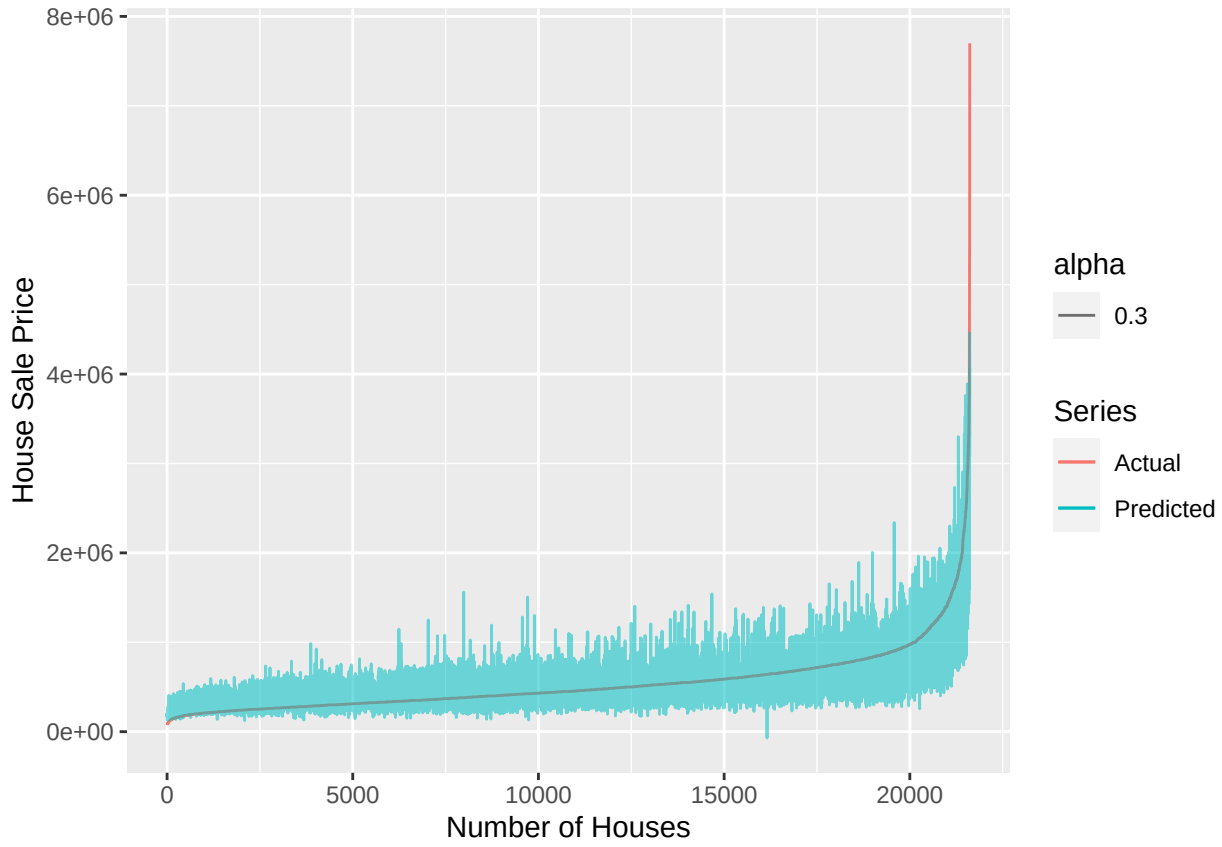
```
actual_predicted <- as.data.frame(cbind(lm_model$id,lm_model$price,
                                        fitted(fitted_model_multiple)))
names(actual_predicted) <- c("id","Actual","Predicted")
```

Order the house by increasing Actual price

```
actual_predicted <- actual_predicted[order(actual_predicted$Actual),]
```

Find the absolute residual and then take mean of that

```
ggplot(actual_predicted,aes(x=1:nrow(lm_model),color=Series)) +
  geom_line(data=actual_predicted, aes(x=1:nrow(lm_model),
                                       y=Actual, color="Actual")) +
  geom_line(data=actual_predicted, aes(x=1:nrow(lm_model),
                                       y=Predicted, color="Predicted",
                                       alpha=0.3)) +
  xlab("Number of Houses") + ylab("House Sale Price")
```



The plot shows that the model closely follows the actual prices. There are a few outliers on Actual values which the model is not able to predict, and that's fine as this model is not influenced by these small outliers.

Thank you!