# LSN 28 - Predicting Real Estate Prices (7.2)

MA376 - LTC J.K.Starling

### Intro

Making predictive models of real estate data is a common technique among realtors and appraisers. We have investigated real estate data at different times earlier in this book. However, in those cases we only focused on two or three variables at a time in order to illustrate key concepts related to multi-variable thinking like interactions and correlations between explanatory variables. Now, we will investigate another real estate dataset to explore practical considerations when building a statistical model with the potential for many more than two or three explanatory variables.

One of the authors (from your book) used to own a home on the south side of Holland, Michigan. He was curious about how well statistical models would align with realtor and tax assessor valuations of his home. He decided to try to predict his home's value using readily available characteristics of homes for sale in the area.

```r
homes <- read.csv("https://raw.githubusercontent.com/jkstarling/MA376/main/HomesFull.csv")
# homes <-read.csv("HomesFull.csv")
homes <- homes %>% select(-ID)
glimpse(homes)
```

```
## Rows: 65
## Columns: 10
## $ Asking_Price <dbl> 163.900, 154.900, 148.999, 144.900, 144.900, 141.900, 139~
## $ Bedrooms     <int> 2, 3, 4, 3, 4, 3, 2, 3, 3, 3, 4, 5, 5, 3, 2, 2, 3, 5, 3, ~
## $ Square_Feet  <int> 1100, 1500, 1050, 1336, 2590, 1410, 960, 1378, 1080, 1296~
## $ Baths        <dbl> 1.0, 1.5, 1.5, 1.5, 2.0, 1.5, 1.0, 1.5, 1.0, 1.0, 1.5, 1.~
## $ Garage       <int> 2, 2, 1, 1, 2, 2, 2, 1, 1, 2, 1, 0, 2, 1, 2, 2, 1, 0, 1, ~
## $ Stories      <int> 2, 1, 1, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 1, 1, 2, 2, 1, ~
## $ Year         <int> 2006, 1988, 1972, 1921, 1901, 1966, 2006, 1946, 1956, 196~
## $ Lot_Size     <int> 12600, NA, 7788, NA, NA, 11220, 7864, NA, 5808, 16335, 87~
## $ Age          <int> 0, 18, 34, 85, 105, 40, 0, 60, 50, 46, 53, 105, 95, 46, 4~
## $ Total_Rooms  <int> 7, 6, 9, 6, 9, 8, 5, 8, 6, 7, NA, 7, NA, NA, 5, 7, 7, 6, ~
```

1) Using your notes from last section (7.1), what might we want to check for with this dataset?

Missing data.

2) Create summary statistics for the mean price and standard deviation for those homes that are a) missing `Lot_Size` and b) missing `Total_Rooms`. Perform a two-sample t-test. Interpret what you see.

```
lot.mis <- homes %>% filter(is.na(Lot_Size))
lot.cmp <- homes %>% filter(!is.na(Lot_Size))
bed.mis <- homes %>% filter(is.na(Total_Rooms))
bed.cmp <- homes %>% filter(!is.na(Total_Rooms))
```

```
homes %>% group_by(is.na(Lot_Size)) %>% summarise(meanPrice = mean(Asking_Price), sd = sd(Asking_Price)
```

```
## # A tibble: 2 x 4
##   ‘is.na(Lot_Size)‘ meanPrice    sd     n
##   <lgl>                 <dbl> <dbl> <int>
## 1 FALSE                  117.  19.4    43
## 2 TRUE                   118.  19.0    22
```

```
lot.ttest <- t.test(lot.mis$Asking_Price, lot.cmp$Asking_Price)
lot.ttest$p.value
```

```
## [1] 0.8158262
```

```
homes %>% group_by(is.na(Total_Rooms)) %>% summarise(meanPrice = mean(Asking_Price), sd = sd(Asking_Pri
```

```
## # A tibble: 2 x 4
##   ‘is.na(Total_Rooms)‘ meanPrice    sd     n
##   <lgl>                    <dbl> <dbl> <int>
## 1 FALSE                     118.  19.3    56
## 2 TRUE                      113.  18.6     9
```

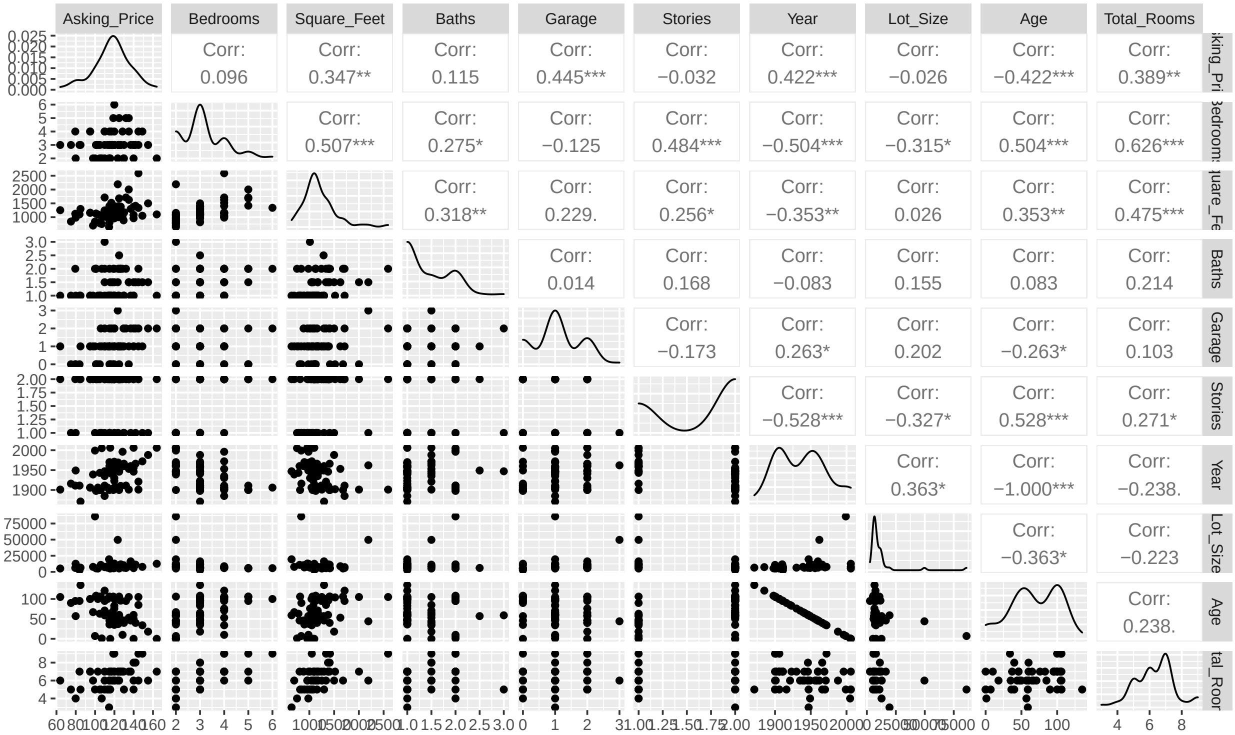
```
room.ttest <- t.test(bed.mis$Asking_Price, bed.cmp$Asking_Price)
room.ttest$p.value
```

```
## [1] 0.4853271
```

\textcolor{blue}{From these tables we see that homes that are missing the lot size variable tend to be a bit more expensive than homes that are not missing lot size, but this is not a statistically significant difference. Similarly, comparing the asking price of homes that are missing the total rooms to those that are not missing this information, we see that homes that have a missing value for this variable are about $5000 less expensive on average, but this is not a statistically significant difference. Thus, it appears that neither of these indicator variables explains much variation in the asking prices. Therefore, if these homes are dropped from the analysis, we do not believe this will impact the distribution of asking prices, but will reduce our sample size.}

3) Create a pairs plot of all variables (except ID). What can we conclude based on these figures? Which variables are most strongly associated with asking price? Which explanatory variables are most strongly associated with each other?

```
homes %>% ggpairs(homes)
```



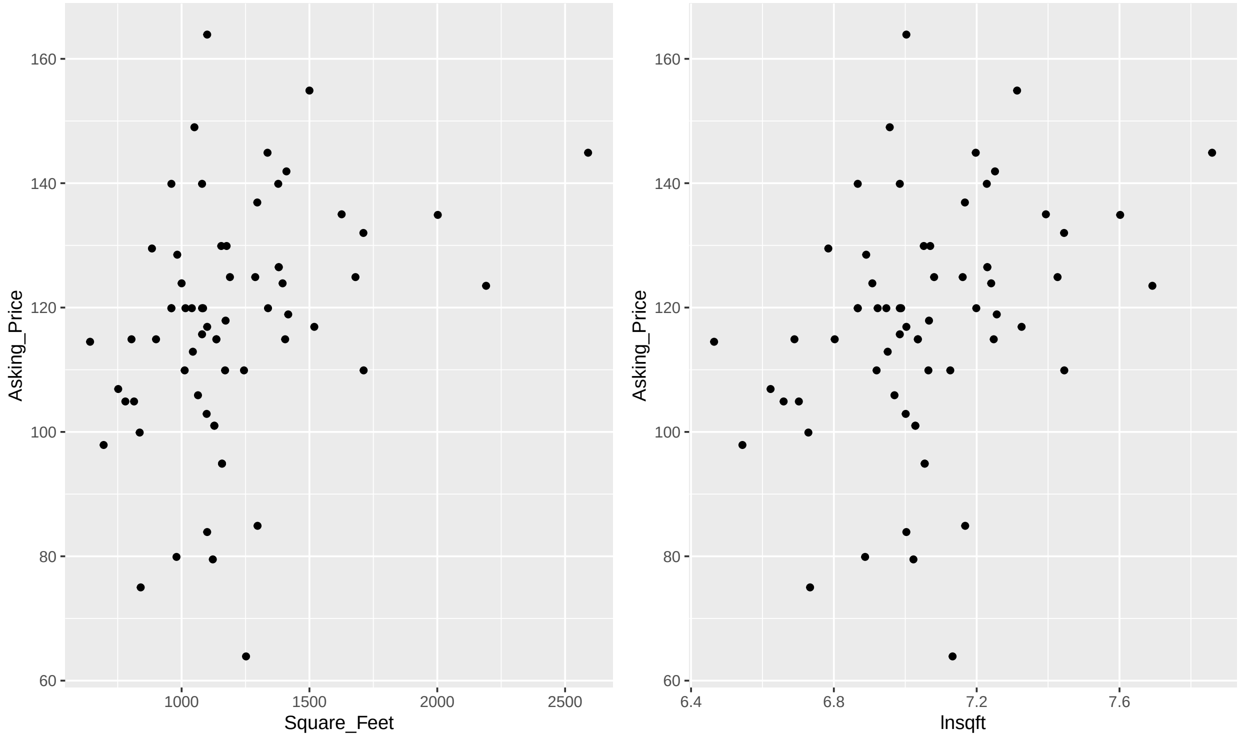
The variables square feet (r = 0.35), garages (r = 0.44), year (r = 0.42), age (r = -0.42), and rooms (r = 0.39) have the strongest associations with the asking price, but maybe we don't need all of these variables in the model, especially if some of them explain the same variation in asking price.

\textcolor{blue}{In particular, the explanatory variables age and year are perfectly correlated with each other (r = -1.00) as age = 2006 - year built; clearly, we don't need both of these variables in the model. We also notice that beds and rooms are pretty strongly associated with each other (r = 0.63), as are square footage and rooms (r = 0.48). Because of these associations, we can feel comfortable dropping rooms from consideration in the model, as its association with price should be reflected in square feet and bedrooms. This is advantageous because eliminating rooms from consideration means that we do not lose the 14% of homes that are missing data for this variable. We also notice that lot size, the other variable with missing values, doesn't appear to have much association with asking price (r= -0.03), so we will also drop it from further consideration. If we had decided lot size was important, we could try to recover the missing information through additional real estate databases of government websites, but this doesn't appear to be worth our time.}

4) Would it be appropriate to transform Square Footage?

```
library(cowplot)
homes$lnsqft <- log(homes$Square_Feet)
p1 <- homes %>% ggplot(aes(x = Square_Feet, y = Asking_Price)) + geom_point()
p2 <- homes %>% ggplot(aes(x = lnsqft, y = Asking_Price)) + geom_point()

plot_grid(p1,p2)
```



Yes. We can take the log of squar footage to get a more linear fit between log(square footage) vs Asking price.

5) What are the most promising variables after the above analysis? How does **model simplicity** play into our decision?

number of garages, age of the home, number of bedrooms, number of bathrooms, number of stories, and ln(sqft). Model simplicity is an important consideration in choosing a model. If two models explain a similar amount of variation in the response variable, then choose the simpler model (e.g., fewer explanatory variables).

6) What is model overfitting? How do we avoid overfitting? Is there a way to improve this method?

Overfitting means that we develop a model which fits the observed data well, but if we used the same model on another set of similar data (e.g., the next year's real estate listings), the model wouldn't necessarily fit that data as well. One way to evaluate whether overfitting is a problem is cross-validation. One simple cross-validation technique is to build the model on a random subset of the data (we call this the discovery sample) and then see how that model performs with the remaining data (we call this the validation sample). A common choice of discovery/validation sample fractions are 2/3 of the observations are in the discovery data sample and the remaining 1/3 of the observations are in the validation sample, although this split can vary depending on the context and the total sample size.

Our book mentions cross-validation; one way to improve this method is to use k-fold cross-validation. This takes multiple train/test set splits to evaluate how the model does on the various subsets (test sets) of data.

7) For this dataset, we randomly selected 2/3 of the data (43 homes) to serve as the discovery sample (HomesDisc). We will subsequently validate the model on the remaining 22 homes (HomesValid).

```
homes.dsc <- read.csv("https://raw.githubusercontent.com/jkstarling/MA376/main/HomesDisc.csv")
homes.val <- read.csv("https://raw.githubusercontent.com/jkstarling/MA376/main/HomesValid.csv")
# homes.dsc <- read.csv("HomesDisc.csv")
# homes.val <- read.csv("HomesValid.csv")

homes.dsc$lnsqft <- log(homes.dsc$Square_Feet)
homes.val$lnsqft <- log(homes.val$Square_Feet)
```

Creating a linear regression model with the variables identified in 5) we get the following results. Explain what you see.

```
home.lm <- homes.dsc %>% lm(Asking_Price ~ Garage + Age + Bedrooms + Stories + Baths + lnsqft, data = .
summary(home.lm)
```

```
##
## Call:
## lm(formula = Asking_Price ~ Garage + Age + Bedrooms + Stories +
##     Baths + lnsqft, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.873  -8.409   3.694   9.931  19.440
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -115.53947   71.40307  -1.618  0.11436
## Garage         7.24828    3.51579   2.062  0.04652 *
## Age           -0.42712    0.07687  -5.556 2.72e-06 ***
## Bedrooms       6.72835    3.82276   1.760  0.08689 .
## Stories        4.20527    5.48441   0.767  0.44822
## Baths         -4.58556    4.41858  -1.038  0.30629
## lnsqft        32.79389   11.35140   2.889  0.00651 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 14.16 on 36 degrees of freedom
## Multiple R-squared:  0.5853, Adjusted R-squared:  0.5161
## F-statistic: 8.467 on 6 and 36 DF,  p-value: 9.244e-06
```

\textcolor{blue}{This model explains 58.5% of the variation in asking prices for the discovery sample, and the typical prediction error is about $14,200. As expected, ln(sqft) and garage are both positively associated with asking price, and age is negatively associated.}

8) Because these are the only variables with small p-values, it's tempting to conclude that ln(sqft), age, and garage are the only important variables in predicting house price. Can we remove bedrooms, stories, and baths from the model and say that's "good enough"? What is a downside of doing this?

It's important to remember that the association of each variable with asking price is adjusted for the other variables in the model. For example, number of baths is negatively associated with price, for homes with the same size, age, and number of garage stalls and so on. It's also important to remember that, because the explanatory variables are associated with each other, removing even one variable from this model could change the statistical significance of the remaining variables.

9) So how do we decide the correct number of variables and which variables to use?

- Backward elimination:
  putting all variables into the model and then dropping variables one-by-one using some guideline (e.g., p-value greater than some value; change in R2 smaller than some value).

- Forward entry: entering variables one-by-one into the model using some criterion (e.g., p-value less than some value; change in $R^2$ larger than some value) to see whether the next variable added contributes significantly more explanation of variation in the response.

- Best-subsets: having the computer check all possible combinations of variables that could be used and reporting the best model for each value of p, the number of explanatory variables in the model using some rule (e.g., smallest SSError, largest $R^2$

10) Using the backward elimination method, use the model created in 7) and remove those variables that are not statistically significant. What is your final model?

We are left with Garage, Age, Bedrooms, lnsqft. We would have removed Bedrooms if we have removed all stat. insignificant variables from the first model.

11) Changing gears to conduct the forward entry method. First thing to do is to compare the $R^2$ and p-values for the model with only one variable. Which variable do we select?

```
myvars <- c("Bedrooms","Baths","Garage",
            "Stories","Age","lnsqft" )

for (myvar in myvars){
  mm <- paste0("Asking_Price ~ ", noquote(myvar))
  my.lm <- homes.dsc %>% lm(mm, data = .)
  my.sum <- summary(my.lm)
  print(paste(myvar, ". r-sq: ", round(my.sum$r.squared,4), ". p-val: ", round(my.sum$coefficients[-1,-
}
```

```
## [1] "Bedrooms . r-sq:  0.0013 . p-val:  0.819"
## [1] "Baths . r-sq:  0.0015 . p-val:  0.808"
## [1] "Garage . r-sq:  0.1652 . p-val:  0.0068"
## [1] "Stories . r-sq:  0 . p-val:  0.9887"
## [1] "Age . r-sq:  0.2193 . p-val:  0.0015"
## [1] "lnsqft . r-sq:  0.0717 . p-val:  0.0826"
```

We select the variable AGE since it has the best r-sq value and lowest p-value.

12) Continue the forward selection process until there are no statistically significant variables to add.

We end up adding Age and lnsqft only if we only include those with statistically significant p-values.

13) Use the `homes.val` dataset to forward selection and backward selection models you created above. How well does each model perform? We can calculate the Mean Square Prediction Error (MSPE) for each model and compare it to the naive model. Which model does the best?

```
fs.preds <- predict(home.lm3, homes.val)
bs.preds <- predict(home.lm2, homes.val)
full.preds <- predict(home.lm, homes.val)

sum((fs.preds-homes.val$Asking_Price)^2)
```

```
## [1] 3814.467
```

```
sum((bs.preds-homes.val$Asking_Price)^2)
```

```
## [1] 4269.146
```

```
sum((full.preds-homes.val$Asking_Price)^2)
```

```
## [1] 4101.462
```

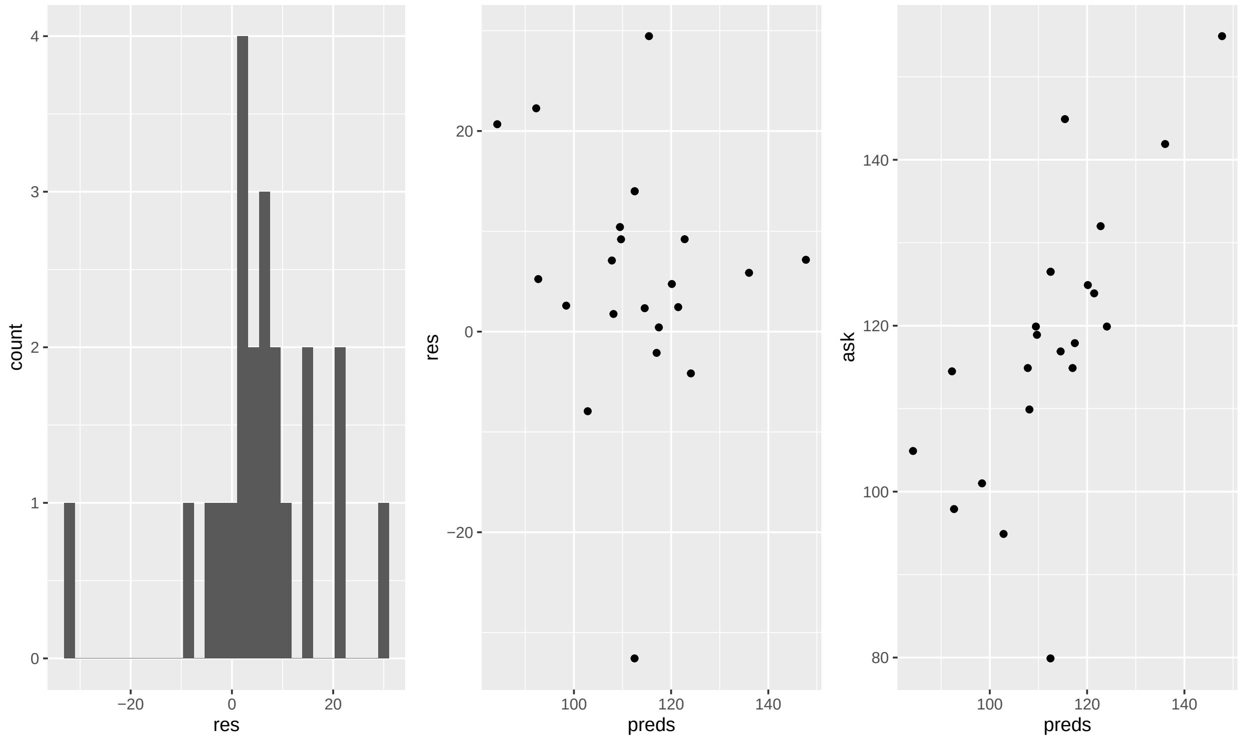
```
#naive model (single-mean model)
naive.pred <- mean(homes.dsc$Asking_Price)
sum((naive.pred - homes.val$Asking_Price)^2)
```

```
## [1] 6081.271
```

The forward selection has the lowest MSPE compared to all three models.

14) Bonus: recreate the figures on p. 535 (Figure 7.2.14)

```
pred <- data.frame(preds = predict(home.lm3, homes.val))
pred$ask <- homes.val$Asking_Price
pred$res <- pred$ask - pred$preds
p1 <- pred %>% ggplot(aes(x=res)) + geom_histogram()
p2 <- pred %>% ggplot(aes(x=preds, y=res)) + geom_point()
p3 <- pred %>% ggplot(aes(x=preds, y=ask)) + geom_point()
plot_grid(p1,p2,p3, ncol = 3)
```



...

...

...

...