Example7_2

Kevin Cummiskey

December 5, 2019

Review

Example 7.2 Predicting Real Estate Prices

The goal of this analysis is to predict house prices in Holland, Michigan. In general, what are some factors that predict home price?

First, we are going to partition our data in a training set (2/3) and test set (1/3). We will fit our model to the training set and assess its ability to predict prices in the test set. Why do we do this? What is this process called?

```
#Model Training Data
houses_train = read.table(file = "http://www.isi-stats.com/isi2/data/HomesDisc.csv",
                          header = T, sep = ",")
#Model Testing Data
houses_test = read.table(file = "http://www.isi-stats.com/isi2/data/HomesValid.csv",
                         header = T, sep = ",")
```

Data Analysis

There are a lot of variables here. How should we preceed with data analysis?

First, let's look at missingness.

```
missing = houses_train %>% summarise_all(funs(round(sum(is.na(.))/n(),2)))
```

```
## Warning: `is_lang()` is deprecated as of rlang 0.2.0.
## Please use `is_call()` instead.
## This warning is displayed once per session.
```

```
## Warning: `lang_modify()` is deprecated as of rlang 0.2.0.
## Please use `call_modify()` instead.
## This warning is displayed once per session.
```

```
## Warning: `mut_node_car()` is deprecated as of rlang 0.2.0.
## This warning is displayed once per session.
```

```
## Warning: The `printer` argument is deprecated as of rlang 0.3.0.
## This warning is displayed once per session.
```

```
kable(missing, caption = "Percent missing data by variable")
```

Table 1: Percent missing data by variable

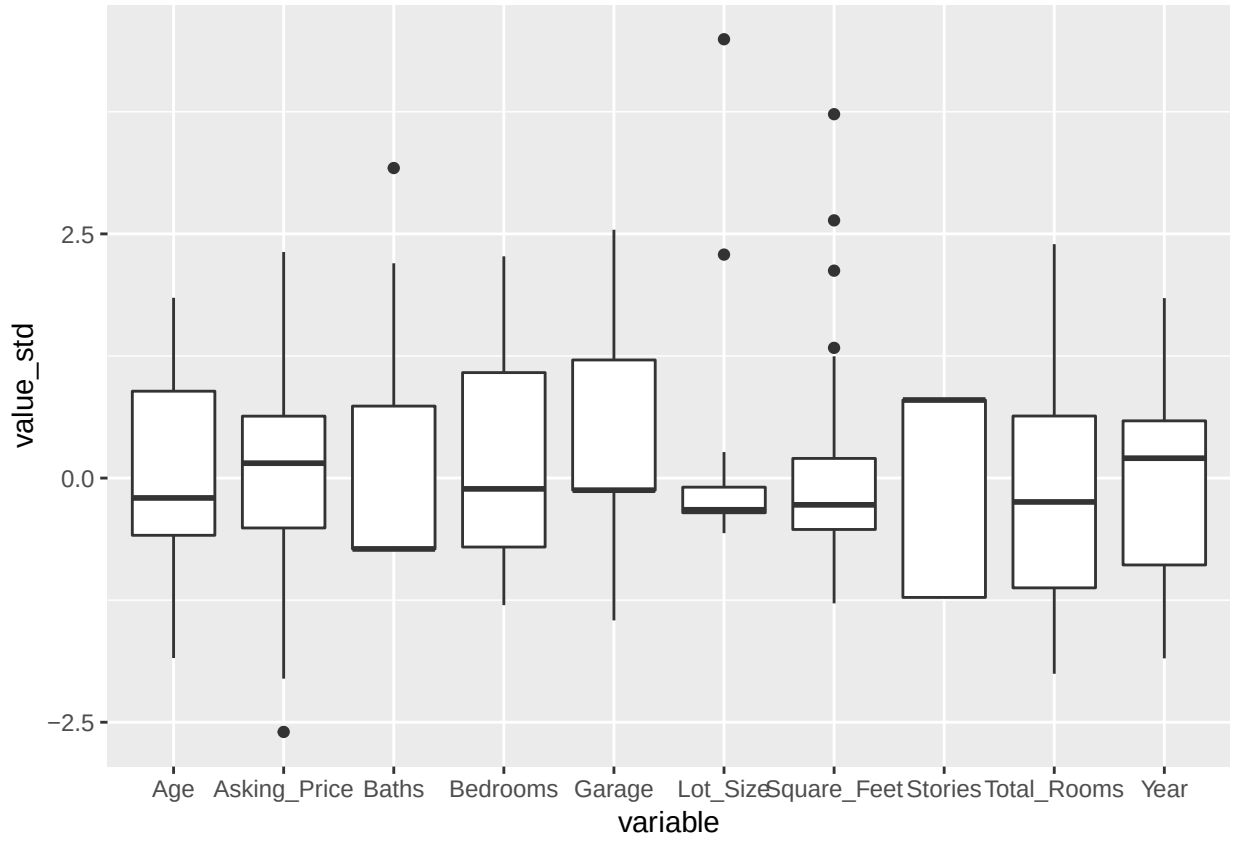
ID	Asking_Price	Bedrooms	Square_Feet	Baths	Garage	Stories	Year	Lot_Size	Age	Total_Rooms
0	0	0	0	0	0	0	0	0.33	0	0.16

What should we do about the missing data?

Next, let's check for outliers.

```
#reshape data
long = houses_train %>% gather("variable", "value", -c(ID))
#standardize variables
long = long %>% group_by(variable) %>% mutate(value_std = scale(value))
long %>% ggplot(aes(x = variable, y = value_std)) + geom_boxplot()
```

```
## Warning: Removed 21 rows containing non-finite values (stat_boxplot).
```



Do we see any outliers? What should we do with them?

Next, let's look at bivariate associations.

```
library(GGally)
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

```
houses_train %>% select(-ID) %>% ggpairs()
```

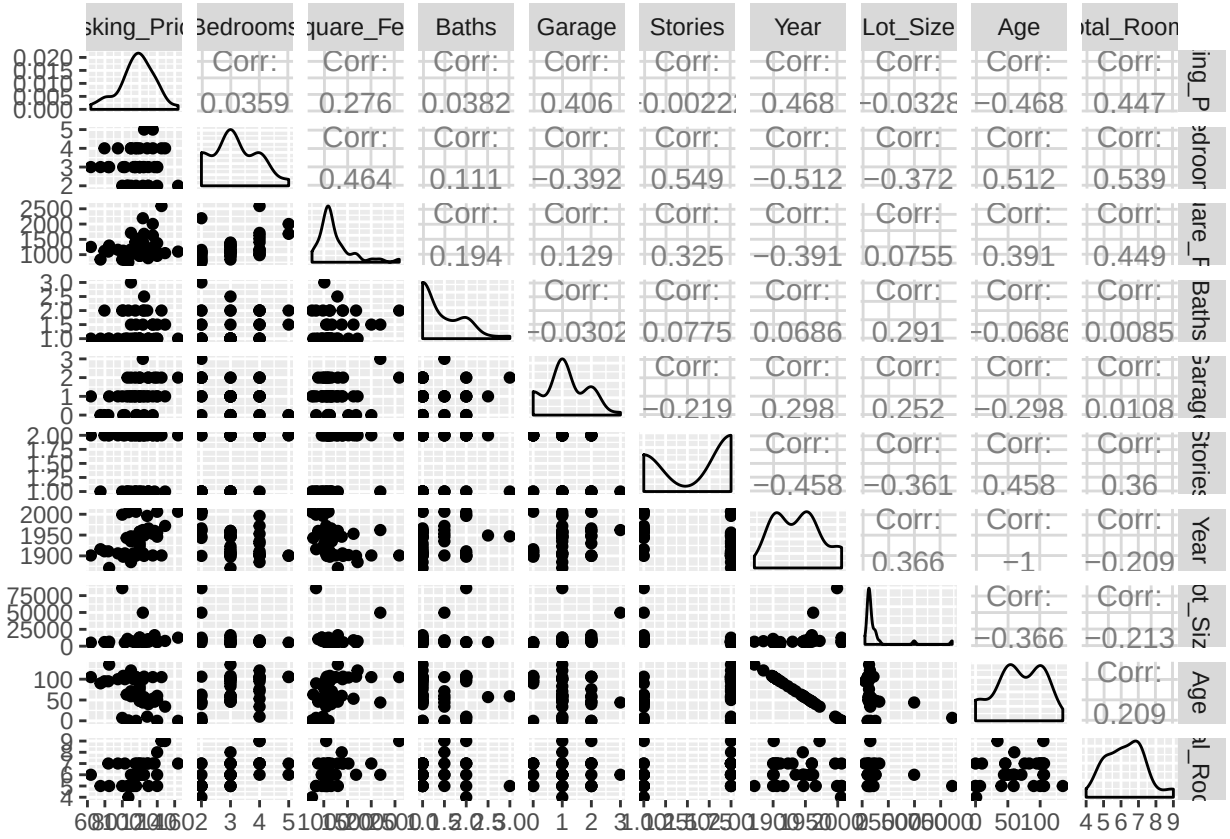
```
## Warning: `list_len()` is deprecated as of rlang 0.2.0.
## Please use `new_list()` instead.
## This warning is displayed once per session.
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 14 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 7 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 14 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 7 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 14 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 7 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 14 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 7 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 14 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 7 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 14 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 7 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 14 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 7 rows containing missing values

## Warning: Removed 14 rows containing missing values (geom_point).

## Warning: Removed 14 rows containing missing values (geom_point).

## Warning: Removed 14 rows containing missing values (geom_point).

## Warning: Removed 14 rows containing missing values (geom_point).

## Warning: Removed 14 rows containing missing values (geom_point).

## Warning: Removed 14 rows containing missing values (geom_point).

## Warning: Removed 14 rows containing missing values (geom_point).

## Warning: Removed 14 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 14 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 19 rows containing missing values

## Warning: Removed 14 rows containing missing values (geom_point).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 7 rows containing missing values
```

```
## Warning: Removed 7 rows containing missing values (geom_point).
## Warning: Removed 7 rows containing missing values (geom_point).
## Warning: Removed 7 rows containing missing values (geom_point).
## Warning: Removed 7 rows containing missing values (geom_point).
## Warning: Removed 7 rows containing missing values (geom_point).
## Warning: Removed 7 rows containing missing values (geom_point).
## Warning: Removed 7 rows containing missing values (geom_point).
## Warning: Removed 19 rows containing missing values (geom_point).
## Warning: Removed 7 rows containing missing values (geom_point).
## Warning: Removed 7 rows containing non-finite values (stat_density).
```



Which variables are most strongly associated with asking price? Are any of these associations nonlinear?

Are any of these variables associated with each other?

Variable Selection

So, which variables do we want to use?

```
model = lm(Asking_Price ~ Garage + Age + Bedrooms + log(Square_Feet) + Stories + Baths, data = houses_t
summary(model)
```

```
##
## Call:
## lm(formula = Asking_Price ~ Garage + Age + Bedrooms + log(Square_Feet) +
##     Stories + Baths, data = houses_train)
##
## 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 .
## log(Square_Feet)   32.79389   11.35140   2.889  0.00651 **
## Stories             4.20527    5.48441   0.767  0.44822
## Baths              -4.58556    4.41858  -1.038  0.30629
## ---
## 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
```

How well does our model predict on the training data?

Here are some common methods used to automate the process:

- Backwards elimination: put all variables in the model and then drop variables one-by-one using some criteria.
- Forwards elimination: enter variables one-by-one using some criteria.

- Best subsets: check all possible combinations for the best model.

These methods are referred to as stepwise regression.

Let's try one of these (backwards elimination) in R.

```
library(leaps)
models = regsubsets(Asking_Price ~ .,
                    data = houses_train %>% select(-ID),
                    method = "backward")
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
## Warning in rval$lopt[] <- rval$vorder[rval$lopt]: number of items to
## replace is not a multiple of replacement length
```

```
summary(models)
```

```
## Subset selection object
## Call: regsubsets.formula(Asking_Price ~ ., data = houses_train %>%
##     select(-ID), method = "backward")
## 9 Variables  (and intercept)
##             Forced in Forced out
## Bedrooms        FALSE      FALSE
## Square_Feet     FALSE      FALSE
## Baths           FALSE      FALSE
## Garage          FALSE      FALSE
## Stories         FALSE      FALSE
## Year            FALSE      FALSE
## Lot_Size        FALSE      FALSE
## Total_Rooms     FALSE      FALSE
## Age             FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: backward
##          Bedrooms Square_Feet Baths Garage Stories Year Lot_Size Age
## 1  ( 1 ) " "      " "         " "   " "    " "     "*"  " "      " "
## 2  ( 1 ) "*"      " "         " "   " "    " "     "*"  " "      " "
## 3  ( 1 ) "*"      " "         " "   "*"    " "     "*"  " "      " "
## 4  ( 1 ) "*"      " "         " "   "*"    " "     "*"  "*"      " "
## 5  ( 1 ) "*"      "*"         " "   "*"    " "     "*"  "*"      " "
## 6  ( 1 ) "*"      "*"         " "   "*"    " "     "*"  "*"      " "
## 7  ( 1 ) "*"      "*"         "*"   "*"    " "     "*"  "*"      " "
## 8  ( 1 ) "*"      "*"         "*"   "*"    "*"     "*"  "*"      " "
##          Total_Rooms
## 1  ( 1 ) " "
## 2  ( 1 ) " "
## 3  ( 1 ) " "
## 4  ( 1 ) " "
## 5  ( 1 ) " "
## 6  ( 1 ) "*"
## 7  ( 1 ) "*"
## 8  ( 1 ) "*"
```