Math 300 Lesson 39 Notes

Case Study

YOUR NAME HERE

July, 2022

Contents

Objectives 1
Reading 1
Lesson 1
Case Study 2
LC 11.1 (Objective 1) 7
Documenting software 7

Objectives

1. Using all the tools and ideas from the course, complete a study that uses the complete data analysis cycle.

Reading

Chapter 11 - 11.2

Lesson

Complete learning check 11.1.

- By completing the entire data analysis cycle, we will review many of the ideas from the course. This will help prepare us for the final.

Libraries

```
library(tidyverse)
library(broom)
library(infer)
library(moderndive)
library(skimr)
```

Case Study

First let's work through the case study in the reading.

The `house_prices` dataset consists of 21,613 houses and 21 variables describing the sale prices of homes sold between May 2014 and May 2015 in King County, Washington, US (for a full list and description of these variables, see the help file by running ?house_prices in the console). In this case study, we'll create a multiple regression model where:

The outcome variable y is the sale `price` of houses. Two explanatory variables: - A numerical explanatory variable x_1: house size `sqft_living` as measured in square feet of living space. Note that 1 square foot is about 0.09 square meters. - A categorical explanatory variable x_2: house `condition`, a categorical variable with five levels where `1` indicates "poor" and `5` indicates "excellent."

Exploratory

Let's get the data and explore them.

Recall the three common steps in an exploratory data analysis we introduced in Subsection 5.1.1:

- Looking at the raw data values.
- Computing summary statistics.
- Creating data visualizations.

```
glimpse(house_prices)
```

```
## Rows: 21,613
## Columns: 21
## $ id            <chr> "7129300520", "6414100192", "5631500400", "2487200875", ~
## $ date          <date> 2014-10-13, 2014-12-09, 2015-02-25, 2014-12-09, 2015-02~
## $ price         <dbl> 221900, 538000, 180000, 604000, 510000, 1225000, 257500,~
## $ bedrooms      <int> 3, 3, 2, 4, 3, 4, 3, 3, 3, 3, 3, 2, 3, 3, 5, 4, 3, 4, 2,~
## $ bathrooms     <dbl> 1.00, 2.25, 1.00, 3.00, 2.00, 4.50, 2.25, 1.50, 1.00, 2.~
## $ sqft_living   <int> 1180, 2570, 770, 1960, 1680, 5420, 1715, 1060, 1780, 189~
## $ sqft_lot      <int> 5650, 7242, 10000, 5000, 8080, 101930, 6819, 9711, 7470,~
## $ floors        <dbl> 1.0, 2.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0, 2.0, 1.0, 1~
## $ waterfront    <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ view          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0,~
## $ condition     <fct> 3, 3, 3, 5, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 3, 3, 3, 4, 4,~
## $ grade         <fct> 7, 7, 6, 7, 8, 11, 7, 7, 7, 7, 8, 7, 7, 7, 7, 9, 7, 7, 7~
## $ sqft_above    <int> 1180, 2170, 770, 1050, 1680, 3890, 1715, 1060, 1050, 189~
## $ sqft_basement <int> 0, 400, 0, 910, 0, 1530, 0, 0, 730, 0, 1700, 300, 0, 0, ~
## $ yr_built      <int> 1955, 1951, 1933, 1965, 1987, 2001, 1995, 1963, 1960, 20~
## $ yr_renovated  <int> 0, 1991, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ zipcode       <fct> 98178, 98125, 98028, 98136, 98074, 98053, 98003, 98198, ~
## $ lat           <dbl> 47.5112, 47.7210, 47.7379, 47.5208, 47.6168, 47.6561, 47~
## $ long          <dbl> -122.257, -122.319, -122.233, -122.393, -122.045, -122.0~
## $ sqft_living15 <int> 1340, 1690, 2720, 1360, 1800, 4760, 2238, 1650, 1780, 23~
## $ sqft_lot15    <int> 5650, 7639, 8062, 5000, 7503, 101930, 6819, 9711, 8113, ~
```

Now summary statistics.

```r
# This is to help with the skim() function
my_skim<-skim_with(numeric = sfl(hist = NULL))
```

```r
house_prices %>%
  select(price, sqft_living, condition) %>%
  my_skim() %>%
  print()
```

```
## -- Data Summary ------------------------
##                            Values
## Name                       Piped data
## Number of rows             21613
## Number of columns          3
## _______________________
## Column type frequency:
##   factor                   1
##   numeric                  2
## ________________________
## Group variables            None
##
## -- Variable type: factor -------------------------------------------------------
##   skim_variable n_missing complete_rate ordered n_unique
## 1 condition             0             1 FALSE          5
##   top_counts
## 1 3: 14031, 4: 5679, 5: 1701, 2: 172
##
## -- Variable type: numeric ------------------------------------------------------
##   skim_variable n_missing complete_rate    mean      sd    p0    p25    p50
## 1 price                 0             1 540088. 367127. 75000 321950 450000
## 2 sqft_living           0             1   2080.    918.   290   1427   1910
##      p75    p100
## 1 645000 7700000
## 2   2550   13540

## $factor
##
## -- Variable type: factor -------------------------------------------------------
##   skim_variable n_missing complete_rate ordered n_unique top_counts
## 1 condition             0             1 FALSE          5 3: 14031, 4: 5679, 5: ~
##
## $numeric
##
## -- Variable type: numeric ------------------------------------------------------
##   skim_variable n_missing complete_rate   mean     sd    p0    p25    p50    p75
## 1 price                 0             1 5.40e5 3.67e5 75000 321950 450000 645000
## 2 sqft_living           0             1 2.08e3 9.18e2   290   1427   1910   2550
## # ... with 1 more variable: p100 <dbl>
```

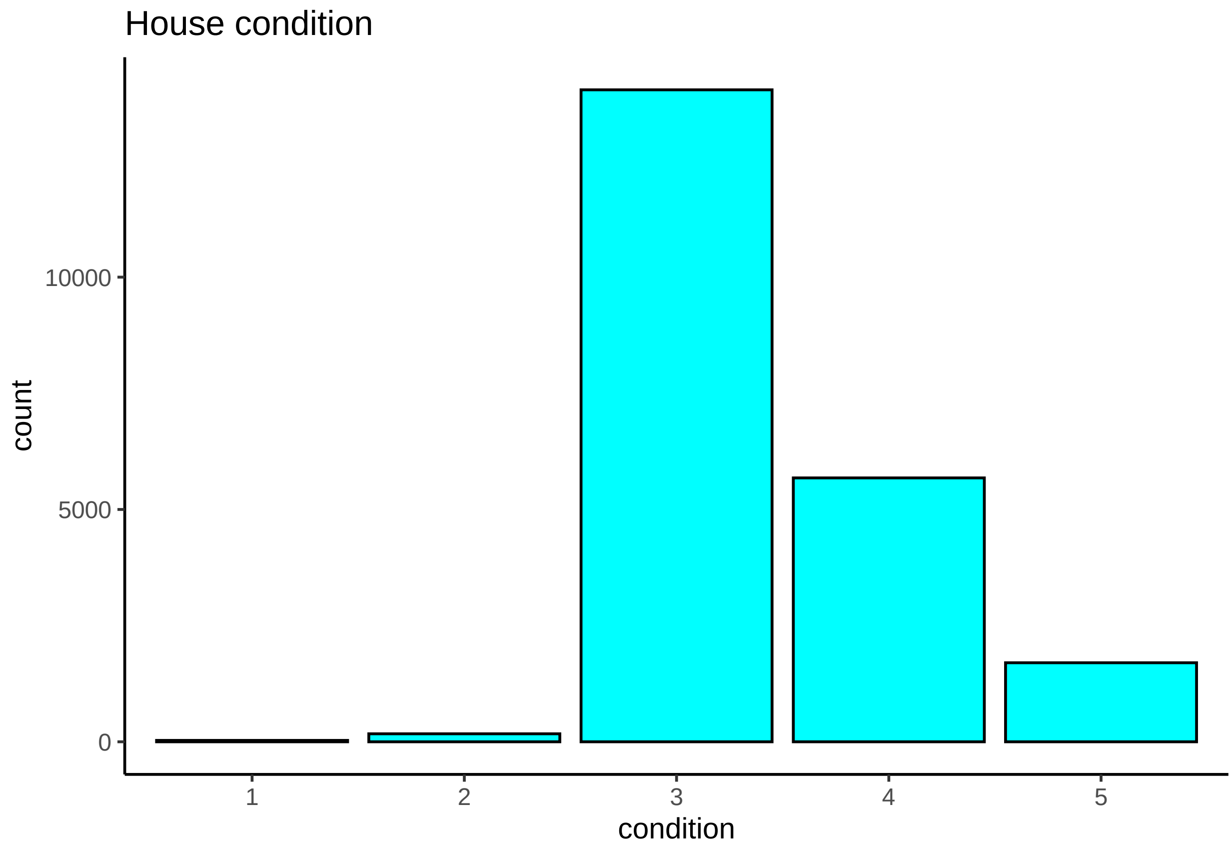
Let's now perform the last of the three common steps in an exploratory data analysis: creating data visualizations.

```r
# Complete the code
# Histogram of house price:
# ggplot(house_prices, aes(x = __________)) +
#   geom_histogram(color = "black", fill = "cyan") +
#   labs(x = "price (USD)", title = "House price") +
#   theme_classic()
```

```r
# Complete the code
# Histogram of sqft_living:
# ggplot(house_prices, aes(x = ____________)) +
#   geom_histogram(color = "black", fill = "cyan") +
#   labs(x = "living space (square feet)", title = "House size") +
#   theme_classic()
```

```r
# Complete the code
# Density plot of sqft_living:
# ggplot(house_prices, aes(x = ___________)) +
#   geom_density(color = "black", fill = "cyan") +
#   labs(x = "living space (square feet)", title = "House size") +
#   theme_classic()
```

```r
# Barplot of condition:
ggplot(house_prices, aes(x = condition)) +
  geom_bar(color = "black", fill = "cyan") +
  labs(x = "condition", title = "House condition") +
  theme_classic()
```



The distribution of the variables are skewed, so let's transform the variables. Let's create new log10 transformed versions of the right-skewed variable `price` and `sqft_living` using the `mutate()` function from Section 3.5, but we'll give the latter the name `log10_size`, which is shorter and easier to understand than the name `log10_sqft_living`.

```
house_prices_reduced <- house_prices %>%
  mutate(
    log10_price = log10(price),
    log10_size = log10(sqft_living)) %>%
  select(log10_price, log10_size, condition)
```

Let's plot the new variables.

```
# Complete the code
# Density plot of log10 sqft_living:
```

```
# Complete the code
# Density plot of log10 price:
```

These variables seem to be more symmetrical.

We are going to revise our multiple regression model to use our new variables:

The outcome variable y is the sale `log10_price` of houses. Two explanatory variables:
- A numerical explanatory variable x_1: house size `log10_size` as measured in log base 10 square feet of living space.
- A categorical explanatory variable x_2: house `condition`, a categorical variable with five levels where `1` indicates "poor" and `5` indicates "excellent."

Multivariate

Let's explore regression models using scatterplots.

```
# Complete the code
# Plot interaction model
# ggplot(house_prices_reduced,
#        aes(x = __________, y = ____________, col = _______________)) +
#   geom_point(alpha = 0.05) +
#   geom_smooth(method = "lm", se = FALSE) +
#   labs(y = "log10 price",
#        x = "log10 size",
#        title = "House prices in Seattle")
```

```
# Complete the code
# Plot parallel slopes model
# ggplot(house_prices_reduced,
#        aes(x = __________, y = _________, col = __________)) +
#   geom_point(alpha = 0.05) +
#   geom_parallel_slopes(se = FALSE) +
#   labs(y = "log10 price",
#        x = "log10 size",
#        title = "House prices in Seattle")
```

Let's create a faceted plot of the interaction model.

```
# Complete the code
# ggplot(house_prices_reduced,
#        aes(x = _________, y = ___________, col = ___________)) +
#   geom_point(alpha = 0.4) +
#   geom_smooth(method = "lm", se = FALSE) +
#   labs(y = "log10 price",
#        x = "log10 size",
#        title = "House prices in Seattle") +
#   facet_wrap(~ condition)
```

Regression model

Let's build the interaction model.

```
# Complete the code
# Fit regression model:
#price_interaction <- lm(___________ ~ ___________ * ____________,
#                         data = house_prices_reduced)
```

```
# Complete the code
# Get regression table:
#get_regression_table(_____________)
```

It is not clear that the interaction model is needed. At most, maybe for a house in condition 5. A machine learning class will help with making a better prediction model.

Predicting

Let's use the model to make predictions. Say you're a realtor and someone calls you asking you how much their home will sell for. They tell you that it's in `condition` = 5 and is sized 1900 square feet. Let's use the interaction model we fit to make predictions! We will use the `augment()` function.

```
# Complete the code
# price_interaction %>%
#   augment(newdata=tibble(condition="___________",log10_size=log10(_____________)))
```

So the predicted price is

```
10^5.724213
```

```
## [1] 529923.3
```

LC 11.1 (Objective 1)

(LC11.1) Repeat the regression modeling in Subsection 11.2.3 and the prediction making you just did on the house of condition 5 and size 1900 square feet in Subsection 11.2.4, but using the parallel slopes model you visualized in Figure 11.6.

```
# Fit regression model:
```

```
# Get regression table:
```

```
# Predict the price
```

Using the rounded numbers from the table:

Documenting software

- File creation date: 2022-07-05
- R version 4.1.3 (2022-03-10)
- `tidyverse` package version: 1.3.1
- `moderndive` package version: 0.5.4
- `skimr` package version: 2.1.4
- `broom` package version: 0.8.0