Assignment 5 Responses

2023-09-25

Assignment 5

Below are my answers to Assignment 5

Question 1: **This first problem will have you explore the AmesHousing data set in order to discover why we had to limit our use of data to homes built on or after 1950.**

Part A: **First, let's look at what we would have found if we hadn't looked only at homes built after 1950. Write some code to**

- load the `AmesHousing` data set into R and create the dataframe `ames`.
- create a new column called `Remodeled`, as we did in the lecture notes, that will be true if `Year_Built != Year_Remod_Add`
- sample 500 data points from the `ames` dataframe, using `set.seed(248)` to make sure we all get the same sample
- run an independent-sample t-test to test whether the means of the remodeled and nonremodeled groups are different.

```r
library(AmesHousing)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
ames <- make_ordinal_ames()


ames$Remodeled <- ames$Year_Built != ames$Year_Remod_Add


set.seed(248)

# names(ames)


samped <- sample_n(ames, size = 500)
remod <- filter(samped, Remodeled==TRUE) %>%
  select(Sale_Price)
```

```r
nonRemod <- filter(samped, Remodeled==FALSE) %>%
  select(Sale_Price)

t.test(remod, nonRemod)
```

```
##
##  Welch Two Sample t-test
##
## data:  remod and nonRemod
## t = -2.5661, df = 448.12, p-value = 0.01061
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -32613.90  -4324.58
## sample estimates:
## mean of x mean of y
##  172598.7  191067.9
```

Part B: **Compare the results of this t-test to those shown in the lecture notes. There should be on aspect that's markedly different.**

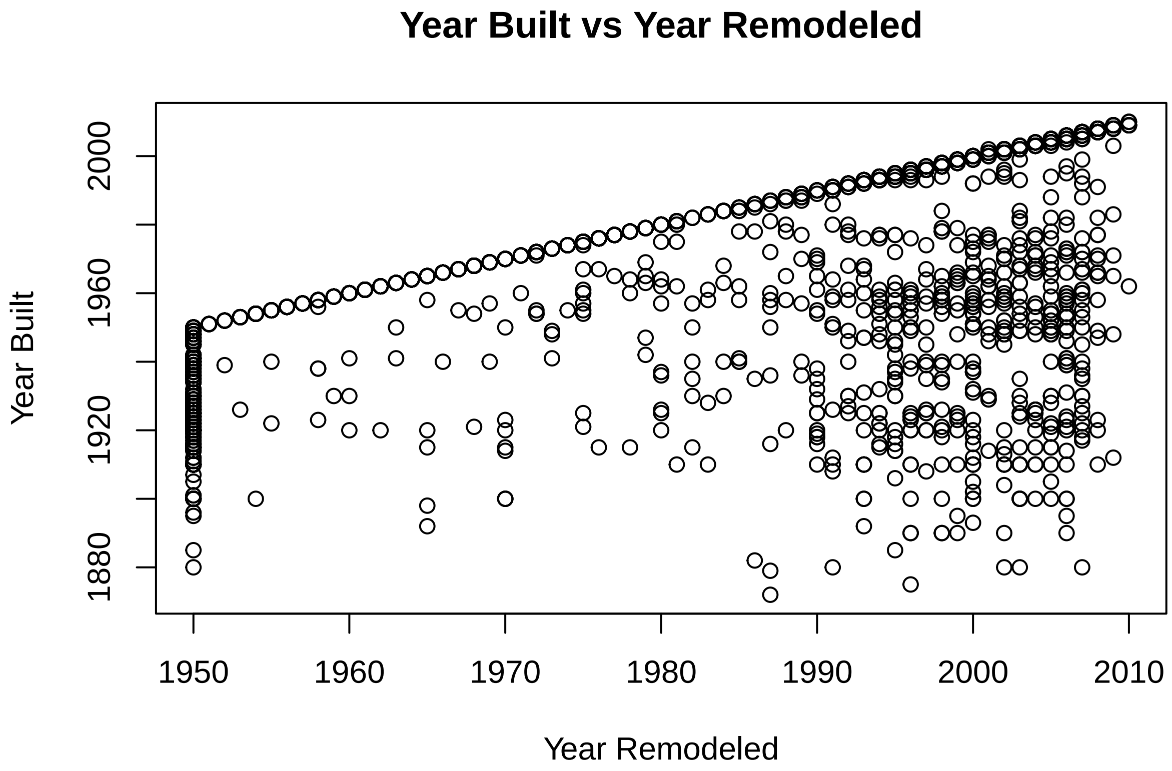
Response: The p-value for my response is about 100 times larger than the p-value given in the notes.

Part C: **What's going on? Make a plot that helps you explore the data. Then write a brief explanation of why the results differ, depending on whether homes built before 1950 are included in the data set.**

Response: **See code and response below**

```r
# pre <- filter(ames, Year_Built <= 1950)
# post <- filter(ames, Year_Built > 1950)

plot(ames$Year_Remod_Add, ames$Year_Built, xlab = "Year Remodeled", ylab = "Year Built", main = "Year Bu
```



As we can see, there was a mass remodeling of houses that were built before 1950 in the year 1950. This likely indicates that these houses weren't being remodeled in the colloquial sense of making upgrades, but were being updated to fit more modern standards or requirements. Hence this is skewing the data since many of these houses date back to the 19th century and may be due to a city-wide requirement rather than the owner's choice.

Question 2: **In this problem, we'll look at simulating the results of an independent-sample t-test in order to gain some insight about sample sizes.**

Part A: **Complete the code stub below to write code to generate two sets of random, normally distributed data, with the sample sizes, means and standard deviations as specified in the code. Then perform an independent-sample t-test using your two samples as the two groups to be compared. [Note: we're purposefully not going to use set.seed here.]**

Response: **See code below**

```
n1 <- 50
n2 <- 50

mu1 <- 15
mu2 <- 17

sigma1 <- 8
sigma2 <- 8

set_a <- rnorm(n1, mu1, sigma1)

set_b <- rnorm(n2, mu2, sigma2)
```

```r
# hist(set_a)
# hist(set_b)

t.test(set_a, set_b, "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  set_a and set_b
## t = -2.3408, df = 83.592, p-value = 0.01081
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf -1.002914
## sample estimates:
## mean of x mean of y
##  13.22125  16.68646
```

Part B: **Suppose your simulated data represents the results of a trial of a new blood pressure medication. Group 1 represents the decrease in blood pressure for a group taking the current "best in class" medication, while Group 2 represents the decrease in blood pressure for a group taking a new competitor's drug. Interpret the results of the test you did in a way that communicates accurately, but without unnecessary complexity. Make sure that you refer to the specifics of this real-world scenario. [Assume $\alpha = .05$ so that you can get a specific Yes/No answer to the test.]**

Response: **See below**

There is moderately strong evidence (p-value of 0.01) to suggest that the medication in Group 2 lowers blood pressure more than the medication presented to Group 1. The null hypothesis is that there is no significant difference in the two groups, i.e. that there is no significant difference between Group 1 medication and Group 2 medication for lowering blood pressure. The alternative hypothesis is that the medication in Group 2 lowers blood pressure more than Group 1. If the two groups **truly** lowered blood pressure by the same amount, there would be about a 1 percent chance to see a sampled data set like the one here. Since we established that we reject the null hypothesis if p < 0.05, we reject the null hypothesis that both medications lower blood pressure the same amount.

Question 3: **This question picks up where the last left off. If you run your code from Question 2 several times (clickvthe green arrow on the code chunk several times), you should notice that sometimes you get a p-valuevthat would reject the null hypothesis, and sometimes you don't. Furthermore, remember that therevare two ways that a hypothesis test might be in error:**

- Type I Error: Rejecting H_0 when H_0 is true (type I error: α)
- Type II Error: Failing to reject H_0 when H_0 is really false. (type II error: β)

Part A: **Run your simulation and t-test many times and calculate the percentage of those runs where you fail to reject the null hypothesis.**

Response: **See code and response below**

```r
# Set a vector of zeros.  We will reject the null hypothesis if the p-value is
# < 0.05 and set the for loop value to 1
c = numeric(20)
for (x in 1:20){
n1 <- 50
n2 <- 50

mu1 <- 15
mu2 <- 17

sigma1 <- 8
sigma2 <- 8

set_a <- rnorm(n1, mu1, sigma1)

set_b <- rnorm(n2, mu2, sigma2)
# hist(set_a)
# hist(set_b)

res <- t.test(set_a, set_b, "less")
## 1 if hypothesis is rejected, 0 otherwise
c[x] <- res$p.value < 0.05
}

# We reject the null
reject_null <- (sum(c) / length(c))

# We fail to reject the null
fail_reject <- 1 - reject_null

fail_reject
```

```
## [1] 0.55
```

Under the assumption that the new drug lowers blood pressure by 2 mmHg more than its competitor, I estimate the probability of failing to detect that difference is 0.55.

Part B: **Suppose you find this probability of Type II error to be too high in the scenario outlined above. One way to decrease it would be to raise your sample size. By experimenting with changing n1 and n2, find the minimum sample sizes necessary to reduce your probability of Type II error to 0.05. [You might try $n = 100$, $n = 200$, etc. as a start. Note: The power of a test is defined to be $1 - \beta$ so we're really finding the sample size that would give a power of 95% for this test.]**

Response: **See code and response below**

```
# Make a loop for the first part, where we repeat the t-test 20 times
# (with n being our sample size)
loop <- function(n){
c = numeric(20)
for (x in 1:20){
  n1 <- n
  n2 <- n

  mu1 <- 15
  mu2 <- 17

  sigma1 <- 8
  sigma2 <- 8

  set_a <- rnorm(n1, mu1, sigma1)

  set_b <- rnorm(n2, mu2, sigma2)
```

```r
  res <- t.test(set_a, set_b, "less")

  c[x] <- res$p.value < 0.05
}
return(c)
}

# Create a list of sample sizes, increasing by 10
samp_size <- seq(from = 100, to = 1000, by = 10)

# We haven't found the appropriate sample size, so set it to FALSE
found <- FALSE
# Loop through our sample list and terminate loop when 1 - beta = 0.95.
for (idx in 1:length(samp_size)){
  # Get vector of p-value < 0.05
  c <- loop(samp_size[idx])
  # Proportion of reject null hypothesis to trials
  reject <- sum(c) / length(c)

  print(paste0("Current 1 - beta: ", reject))
  print(paste0("Current sample size: ", samp_size[idx]))
  # if the proportion of reject is greater than or equal to 0.95, set found
  # to the sample size and terminate the loop
  if(reject >= 0.95){found <- samp_size[idx]

  break}
}
```

```
## [1] "Current 1 - beta: 0.6"
## [1] "Current sample size: 100"
## [1] "Current 1 - beta: 0.7"
## [1] "Current sample size: 110"
## [1] "Current 1 - beta: 0.55"
## [1] "Current sample size: 120"
## [1] "Current 1 - beta: 0.7"
## [1] "Current sample size: 130"
## [1] "Current 1 - beta: 0.75"
## [1] "Current sample size: 140"
## [1] "Current 1 - beta: 0.85"
## [1] "Current sample size: 150"
## [1] "Current 1 - beta: 0.7"
## [1] "Current sample size: 160"
## [1] "Current 1 - beta: 0.75"
## [1] "Current sample size: 170"
## [1] "Current 1 - beta: 0.65"
## [1] "Current sample size: 180"
## [1] "Current 1 - beta: 0.8"
## [1] "Current sample size: 190"
## [1] "Current 1 - beta: 0.8"
## [1] "Current sample size: 200"
## [1] "Current 1 - beta: 0.9"
## [1] "Current sample size: 210"
## [1] "Current 1 - beta: 0.7"
## [1] "Current sample size: 220"
## [1] "Current 1 - beta: 0.75"
## [1] "Current sample size: 230"
## [1] "Current 1 - beta: 0.9"
## [1] "Current sample size: 240"
## [1] "Current 1 - beta: 0.9"
## [1] "Current sample size: 250"
## [1] "Current 1 - beta: 1"
## [1] "Current sample size: 260"
```

```r
# Print sample size
print(found)
```

```
## [1] 260
```

```r
# Print 1 - beta

print(reject)
```

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

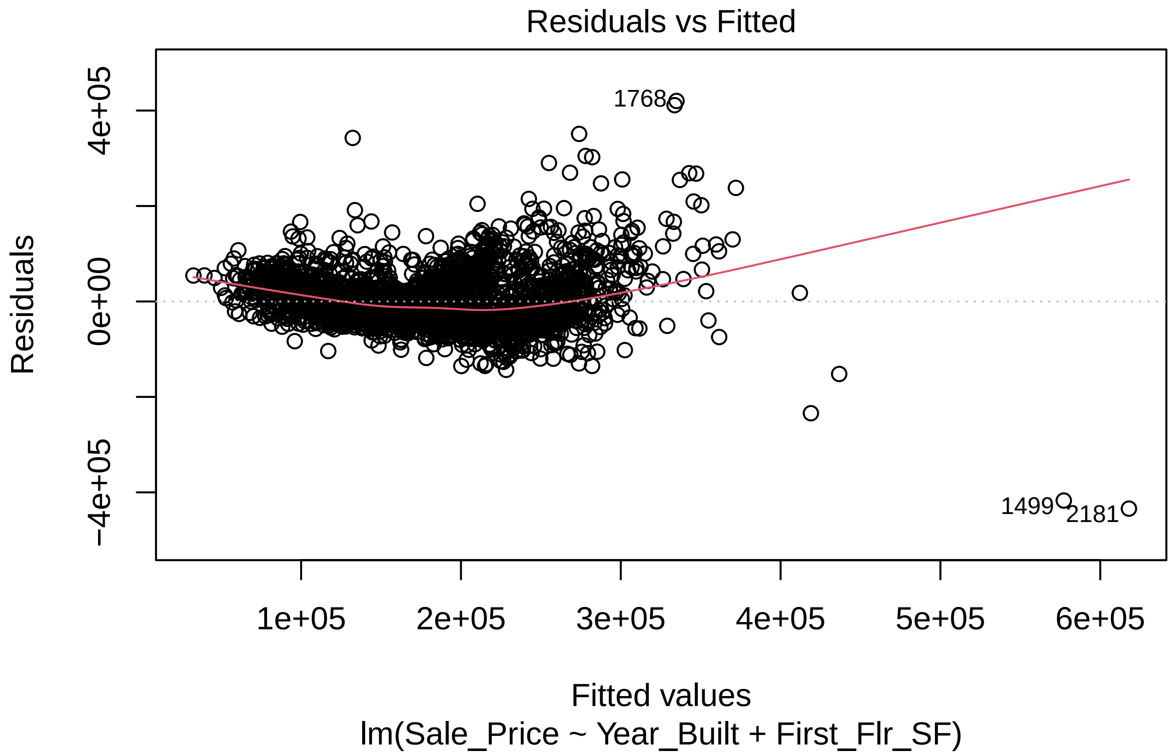
Question 4: **4. We saw that taking the log transform of Sale_Price was one way to try to get control of the variance of residuals in our simple linear regression. Another way to try to improve the model's fit and the model assumptions is to add further important variables to the model. Often bad fit is the result of confounding from lurking variables. In this problem, start with the first multiple linear regression model we looked at in the lecture. It's named fit.original in the code chunk below. Then do the following:**

Part A: Write code to plot residuals vs. fit for `fit.original` (try to find the right plot option to print out only that graph).

Response: **See code below**

```r
fit.original <- lm(Sale_Price ~ Year_Built + First_Flr_SF, data=ames)

plot(fit.original, which = 1)
```



Part B: **Write code to output the R^2 adjusted for `fit.original`**

Response: **See code below**

```
sumar <- summary(fit.original)
print(sumar$adj.r.squared)
```

```
## [1] 0.5339375
```

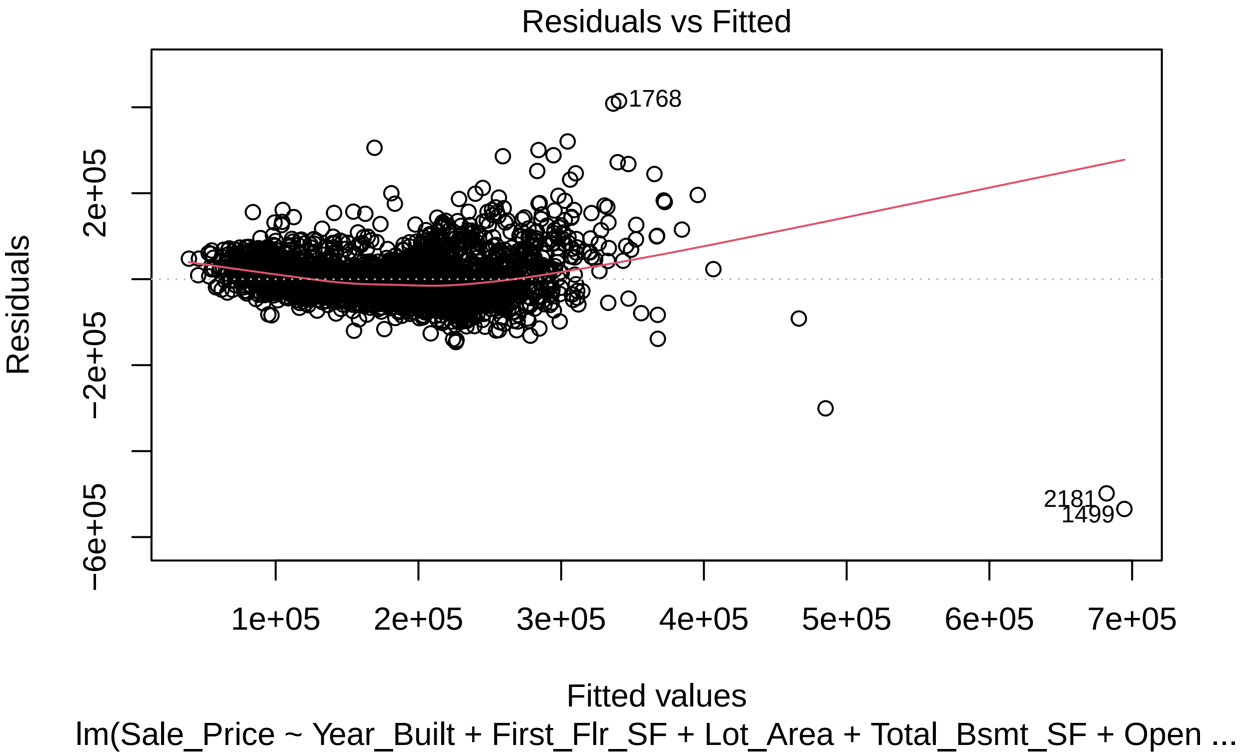
Part C: **Think about which variables which variables might also be highly related to Sale_Price. Then create a new model using those variables that has a higher R2-adjusted and seems to satisfy the linearity and constant variance conditions better. Output the plot of fit vs. residuals and the adjusted r-squared for this new model as well. There's no single right answer here—for now you're just playing around to see what you'll find.**

Response: **See code and response below**

```
fit.new <- lm(Sale_Price ~ Year_Built + First_Flr_SF +
                Lot_Area + Total_Bsmt_SF + Open_Porch_SF, data = ames)
summary(fit.new)$adj.r.squared
```

```
## [1] 0.5698008
```

```
plot(fit.new, which = 1)
```



Residuals vs Fitted
Residuals
2e+05
−2e+05
−6e+05
1768
2181
1499
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
Fitted values
lm(Sale_Price ~ Year_Built + First_Flr_SF + Lot_Area + Total_Bsmt_SF + Open ...