Homework 9

The purpose of this homework is to practice fitting logistic regression models, to learn how to join data frames, and to learn how to create maps. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through Gradescope by 11pm on Sunday November 26th.

As always, if you need help with the homework, please attend the TA office hours which are listed on Canvas and/or ask questions on Ed Discussions. Also, if you have completed the homework, please help others out by answering questions on Ed Discussions, which will count toward your class participation grade.

Part 1: Logistic regression

As we have discussed in class, logistic regression can be used to estimate the probability that a response variable `y` belongs to one of two possible categories. In these exercises, you will learn how to use logistic regression by fitting models that can give the probability that a car is new or used based on other predictors, including the car's price.

Part 1.1 (6 points): The code below creates a data frame called `toyota_data` that has sales information on 500 randomly selected new and used Toyota cars. It also changes the order of the levels on the categorical variable `new_or_used_bought` which will make the plotting and modeling work better in these exercises.

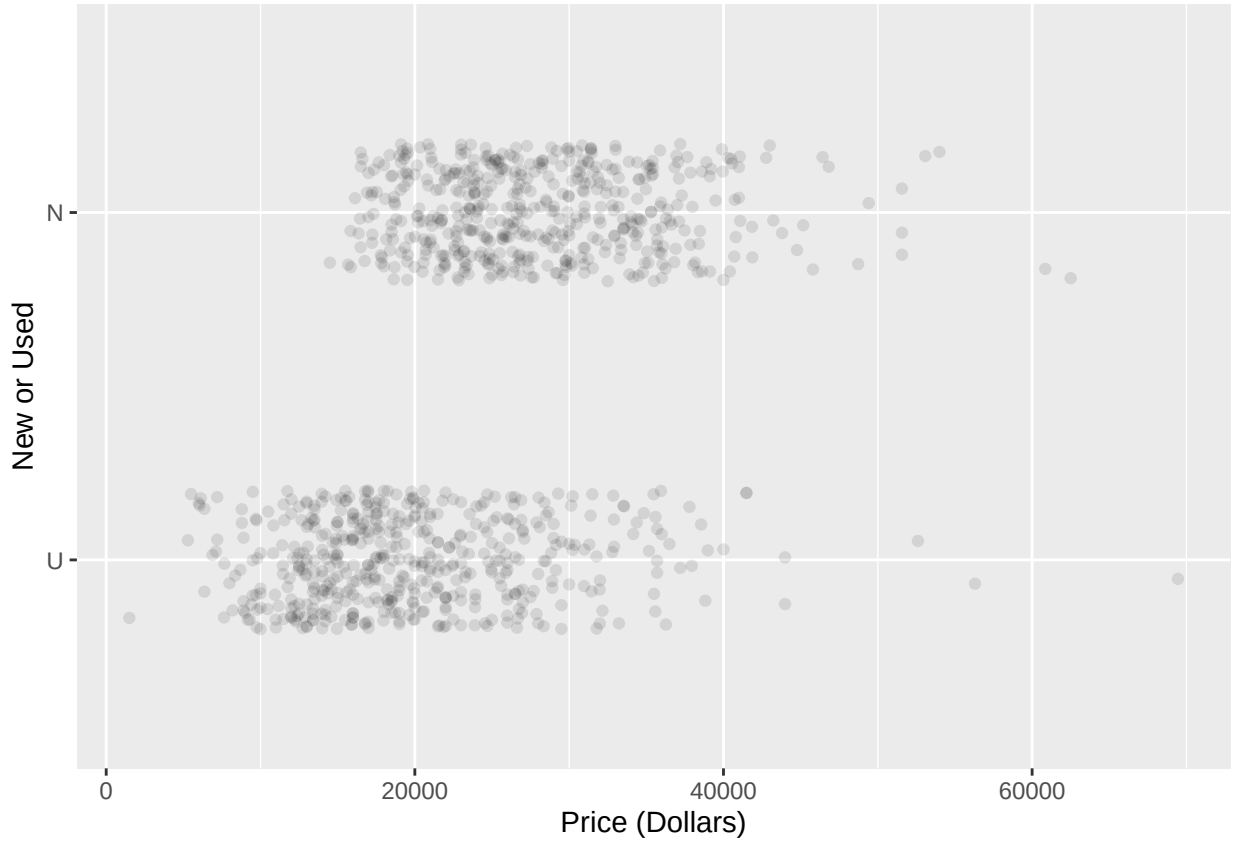
To start, create a visualization of the data using ggplot where price is on the x-axis and whether the car is new or used is on the y-axis. Use a `geom_jitter()` to plot points that have less overlap, and adjust the amount of vertical jitters so that there is a clear separation between the new and used cars (using `position_jitter()` inside the `geom_jitter()` could be useful). Also set the alpha transparency appropriately to also help deal with over-plotting.

```
load("car_transactions.rda")

# get toyota cars and change the order of the levels of the new_or_used_bought variable
toyota_data <- filter(car_transactions, make_bought == "Toyota") |>
  mutate(new_or_used_bought =
           factor(as.character(new_or_used_bought,
                               levels = c("U", "N")))) |>
  mutate(new_or_used_bought = relevel(new_or_used_bought, "U")) |>
  na.omit() |>
  group_by(new_or_used_bought) |>
  slice_sample(n = 500)
```

```r
# visualize the data

toyota_data %>% ggplot(aes(price_bought, new_or_used_bought)) +
  geom_jitter(alpha = .1, position = position_jitter(height = .2)) +
  xlab("Price (Dollars)") +
  ylab("New or Used")
```



Part 1.2 (8 points): Now fit a logistic model to predict the probability a car is new based on the price that was paid for the car using the `glm()` function. Then extract the offset and slope coefficients and save them to objects called `b0` and `b1`. Based on these coefficients, calculate what the predicted log-odds, odds and probability are that a car is new if it costs $15,000. Print these values out, and in the answer section, report what these values are.

A note on R and logistic models: `new_or_used_bought` is a factor variable where "U" is the lowest level and "N" is the highest level. This means we can plug the variable into the `glm` function without further transformation, since R will interpret this dichotomous factor variable as composed of the 1s and 0s that a logistic regression expects.

```r
# build the logistic regression function
lr_fit <- glm(new_or_used_bought ~ price_bought,
              data = toyota_data, family = "binomial")
```

```r
# extract the coefficients
b0 <- coefficients(lr_fit)[1]
b1 <- coefficients(lr_fit)[2]

# log odds
(log_odds <- b0 + (b1 * 15000))
```

```
## (Intercept)
##   -1.247285
```

```r
# odds
(odds <- exp( b0 + b1 * 15000))
```

```
## (Intercept)
##   0.2872838
```

```r
# probability
(prob <- (exp(b0 + b1 * 15000)) / (1 + exp(b0 + b1 * 15000)))
```

```
## (Intercept)
##   0.2231705
```

Answer:

The predicted values that a car is new if it cost $15,000 are:

1. log-odds = -1.384158
2. odds = 0.2505348
3. probability = 0.2003421

Part 1.3 (8 points):

Now using the `b0` and `b1` coefficients calculated in part 1.2, write a function `get_prob_new()` that takes a price a car was sold for as an input argument and returns the probability the car was new using the equation:

$$\text{P(new|price)} = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 \cdot x_{price})}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 \cdot x_{price})}$$

Then use the `get_prob_new()` function to predict the probability that a car that sold for $10,000 was a new car and report this probability. Comment on whether you would expect a predicted probability this high based on the visualization of the data you created in part 1.1.

```r
get_prob_new <- function(price){
  prob <- exp(b0 + b1 * price) / (1 + exp(b0+ b1 * price))
  return(prob)
}

get_prob_new(1e4)
```

```
## (Intercept)
##   0.1232236
```

Answer:

$P(new|price = 15000) = 0.1052264$

Since the visualization of the data in 1.1 suggests that no new cars were sold for $10000, I would not expect a predicted probability this high.

Part 1.4 (8 points): Next plot the data again as you did in part 1.1, but this time you will add a red line showing the logistic regression function for predicting the probability a car is new based on its price. In order to plot the logistic regression line at the appropriate height on the figure, create a function called `plot_prob_new(price)` that returns the value of `get_prob_new(price)` plus 1.

Adding 1 is necessary because R understands the "U" level of our factor variable to be 1 numerically and "N" to be 2, but `get_prob_new` returns probabilities between 0 and 1.

Then, use the `stat_function()` function in the ggplot2 package, in combination with the `plot_prob_new()` function, to create a visualization where you create the scatter-plot visualization you created in part 1.1. Make sure that you also overlay the logistic regression line in red on the plot. Use `xlim` to toggle the dimensions of the x-axis appropiate, such that the plot centers the points and shows all aspects of the plotted function line.

```r
# to plot this function we add 1 to it
plot_prob_new <- function(price) {
get_prob_new(price) + 1
}

# plot the logistic regression function
toyota_data |>
ggplot(aes(x = price_bought, y = new_or_used_bought)) +
geom_jitter(alpha = .1, position = position_jitter(height = .2)) +
xlab("Price (Dollars)") +
ylab ("New or Used") +
stat_function(fun = plot_prob_new, color = "red") +
xlim(0, 70000)
```

Part 1.5 (10 points): Let's now fit a multiple logistic regression to the data in order to get the probability a car is new given the price of the car (`price_bought`) and the number of miles driven (`mileage_bought`). From the model that you fit, extract the regression coefficient estimates and intercept, price_bought and mileage_bought coefficients and save them to objects called `b0`, `b1` and `b2`. Then write a function called `get_prob_new2()` that uses the coefficients to predict the probability a car is new using the equation:

$$\text{P(new|price, mileage)} = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 \cdot price + \hat{\beta}_2 \cdot mileage)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 \cdot price + \hat{\beta}_2 \cdot mileage)}$$

Finally, use the `get_prob_new2()` function to predict the probability that a car that sold for 10,000 and had 500 miles was a new car, and the probability that a car that sold for 10,000 and had 5000 miles was a new car. Report these probabilities in the answer section.

```
# fit the model
lr_fit2 <- glm(new_or_used_bought ~ price_bought + mileage_bought, data = toyota_data, family = "binomia

# extract the regression coefficients
b0 <- coefficients(lr_fit2)[1]
b1 <- coefficients(lr_fit2)[2]
b2 <- coefficients(lr_fit2)[3]

# build a function
get_prob_new2 <- function(price, mileage) {
```

```
(exp(b0 + (b1 * price) + (b2 * mileage))) / (1 + exp(b0 + (b1 * price) + (b2 * mileage)))
}

# predict probability
get_prob_new2(10000, 500)
```

```
## (Intercept)
##   0.9308157
```

```
get_prob_new2(10000, 5000)
```

```
## (Intercept)
##  0.01364563
```

Answer:

For a car that sold for $10,000 and had 500 miles, the predicted probability that the car is new is: 0.8563813

For a car that sold for $10,000 and had 5000 miles, the predicted probability that the car is new is: 0.002199367

Part 1.6 (5 points):

This is a "challenge problem" that you should try to figure out without getting help from the TAs.

We can also fit logistic regression models with categorical predictors. The code below creates a data set that selects 500 random new and used Toyotas and BMWs. Let's use this data to build a model that predicts whether a car is new or used based on the cars price and whether it is a Toyota and BMW. To start, use ggplot to visualize the data where: `price_bought` is mapped to the x-axis, `new_or_used_bought` is mapped to the y-axis, `make_bought` is mapped to color. Also use the `geom_jitter()` glyph using an appropriate amount of jitter, and as always, label your axes.

Then, fit a logistic regression model to predict whether a car is new or used based on the price when the car was bought and whether the car was a Toyota or BMW. Finally, print out odds ratio for predicting whether a car is new or used if it is a Toyota relative to a BMW. Report in the answer section what the odds ratio is and how to interpret what it means.

```
set.seed(230)

# get Toyota cars and change the order of the levels of the new_or_used_bought variable
toyota_bmw_data <- filter(car_transactions, make_bought %in% c("Toyota", "BMW")) |>
  mutate(new_or_used_bought = factor(as.character(new_or_used_bought), levels = c("U", "N"))) |>
  mutate(new_or_used_bought = relevel(new_or_used_bought, "U")) |>
  na.omit() |>
  group_by(new_or_used_bought, make_bought) |>
  slice_sample(n = 500)
```

Answer:

Part 2: Practice building regression models

To help consolidate what you have learned this semester, and to prepare for the final project, let's practice building another regression model trying to predict the price of a condominium in New Haven. In particular, you will go through the full model building process of choosing, fitting, assessing, and using a model.

Motivation: In May 2022, I was interested in buying a condominium in New Haven. One condominium I was interested in was located at 869 Orange St, Unit 5E. The asking price for the condominium was $475,000. However, usually buyers don't put in offers that are exactly at the asking price, but instead put in offers above or below. The advantage of putting in a lower offer is that one could save money, but there is a risk someone else could put in a higher offer that would be taken instead of your offer.

In order to help get a sense of how much money I should offer, I scraped data on all the house prices in New Haven from the New Haven online assessment website (https://gis.vgsi.com/newhavenct/). The goal of this exercise is to use this data to create visualizations and regression models in order to come up a offer that you think will be very likely to be accepted, but that will not be way more than the property is really worth (i.e., more than you could resell the property for in the future).

When building regression models below, the response variable you will try to predict is `sale_price` which is the price that different condominium in New Haven sold for. Explanatory variables that might be useful to include in your model include:

1. `zone`: Contains information about the neighborhood where condominiums are located.
2. `year_built`: The year the condominium was built.
3. `building_area`, `living_area` and `size_acres`: Information about the size of the property.
4. `appraisal`: How much tax collectors assess that the property was worth.
5. `sale_date`: The year the property was sold.

You can also use other variables that in the data that are not listed above. For more information about different variables, see the online assessment website https://gis.vgsi.com/newhavenct/ (I don't know much more about these variables than the information that is listed on this site).

Finally, note that there is no one "correct answers" to the exercises below. Rather, just use visualizations and modeling to come up with "useful" models that can given insight into the question of interest and we can discuss more about the models everyone has developed in class.

Part 2.1 (5 points): Before building regression models, it is almost always useful to apply transformations of the data. For example, it could be useful to filter the data to use only subsets of the data that are most relevant. It could also be useful to apply transformations to variables in the dataset. For example, the price a property sold for is going to be heavily influenced on how long ago the property sold, so converting `sale_date` into a variable such as `year_old_sold` which indicates how many years ago the property was sold (and hence the year the is from `sale_price`) could also be useful.

In the R chunk below, please transform the data in anyway you think would be useful for subsequent analyses and save your transformed data to a data frame called `property_data` (3-5 transformations/filtering operations should be fine, but there is flexibility here to do what you think is best). Then in the answer section please**briefly** describe the rationale behind any data you removed (i.e., rationale behind any data filtered out of the dataset).

Hints: the `lubridate` library contains many useful functions for dealing with dates. For example, the `year()` function can take a date and extract the year from it, etc.

```r
library(dplyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
load("condos_may_2022.rda")

property_data <- condos %>%
  select(zone, year_built, building_area, appraisal, sale_date, sale_price,
         street_number) %>%
  na.omit %>%
  mutate(year_old_sold = 2022 - year(sale_date)) %>%
  mutate(years_old = 2022 - year_built) %>%
  mutate(log_sale = log10(sale_price)) %>%
  filter(sale_price > 1)

# size_acres = 0 for all but 3 data points
condos %>% filter(size_acres > 0)
```

```
## # A tibble: 3 x 32
##   street    street_~1 neigh~2 zone  zone_2 use_c~3 build~4 longi~5 latit~6 url
##   <chr>     <chr>     <chr>   <chr> <chr>  <chr>     <dbl>   <dbl>   <dbl> <chr>
## 1 ORANGE ST 637 ORAN~ 1020    1200  RM2    Condom~       1   -72.9    41.3 http~
## 2 ORANGE ST 637 ORAN~ 1020    1200  RM2    Condom~       1   -72.9    41.3 http~
## 3 ORANGE ST 637 ORAN~ 1020    1200  RM2    Condom~       1   -72.9    41.3 http~
## # ... with 22 more variables: account_number <chr>, pid <dbl>,
## #   book_and_page <chr>, address <chr>, year_built <dbl>, building_area <dbl>,
## #   living_area <dbl>, building_pct_good <dbl>, size_acres <dbl>,
## #   frontage <dbl>, depth <dbl>, alt_land_appr <chr>, valuation_year <int>,
## #   assessment <dbl>, assessed_val <dbl>, appraisal <dbl>, appraised_val <dbl>,
## #   sale_date <date>, sale_price <dbl>, replacement_cost <dbl>,
## #   less_deprecation <dbl>, replacement_cost_less_depreciation <dbl>, and ...
```

```r
# no living area data for most data points
# no difference between living and building area
condos %>% select(living_area) %>% na.omit
```

```
## # A tibble: 1,505 x 1
##    living_area
##          <dbl>
##  1         947
##  2         967
##  3         780
##  4         870
##  5         905
##  6         980
```

```
##  7         675
##  8         595
##  9         610
## 10         680
## # ... with 1,495 more rows
```

```
area_diff <- condos %>% select(building_area, living_area) %>%
  mutate(area_difference = building_area - living_area) %>% na.omit
mean(area_diff$area_difference)
```

```
## [1] 0
```

Answer:

size_acres was filtered out because it is nonzero for only three properties.

living_area was filtered out because there is no data for most properties, and where there is data, living_area is equal to building_area.

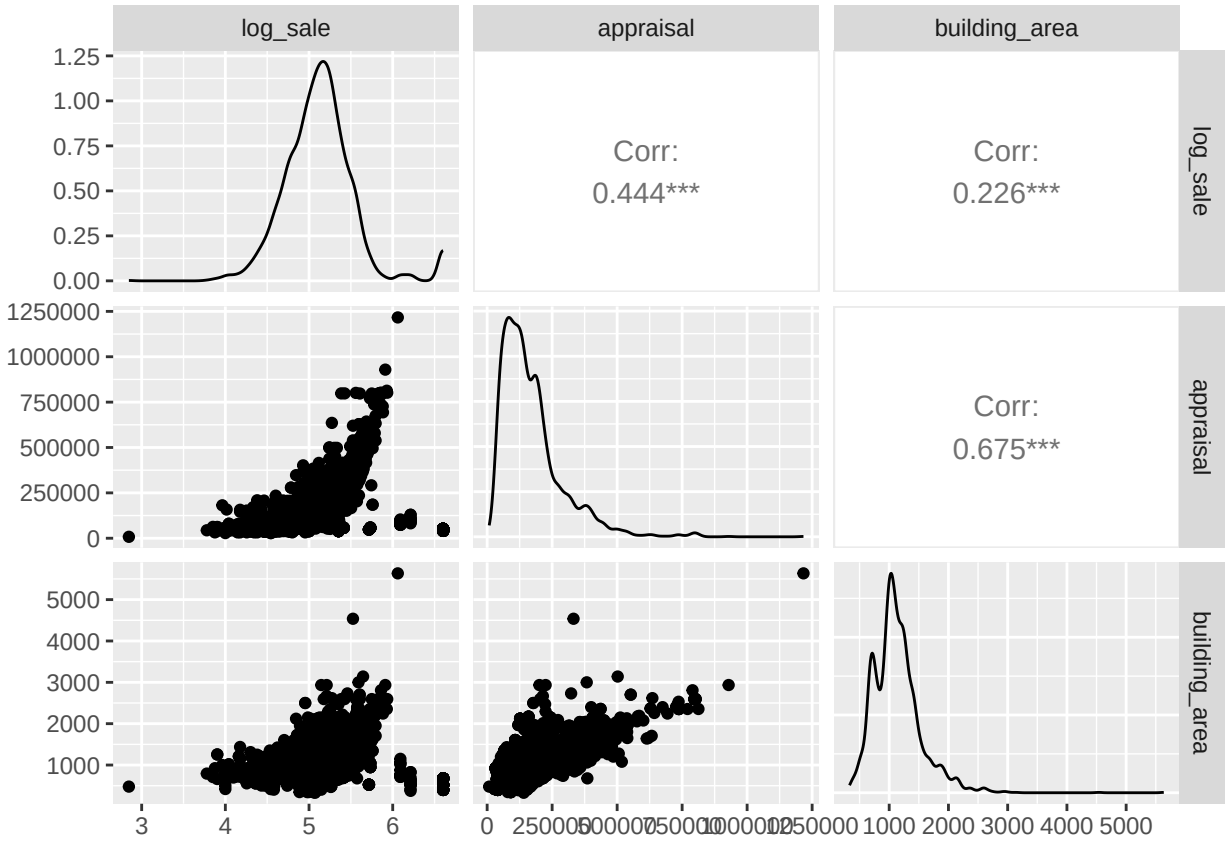
Properties that sold for $1 or less were filtered out because it is unlikely that the owner of a property will sell for so little unless the buyer is a friend or family member.

Part 2.2 (5 points): Before modeling the data, one should almost always visualize the data. Please go ahead to create 2 visualizations that give insight into the housing market and/or into how to begin to model the data. In the answer section briefly describe what your plot(s) are showing and how they given insight into the question of interest.

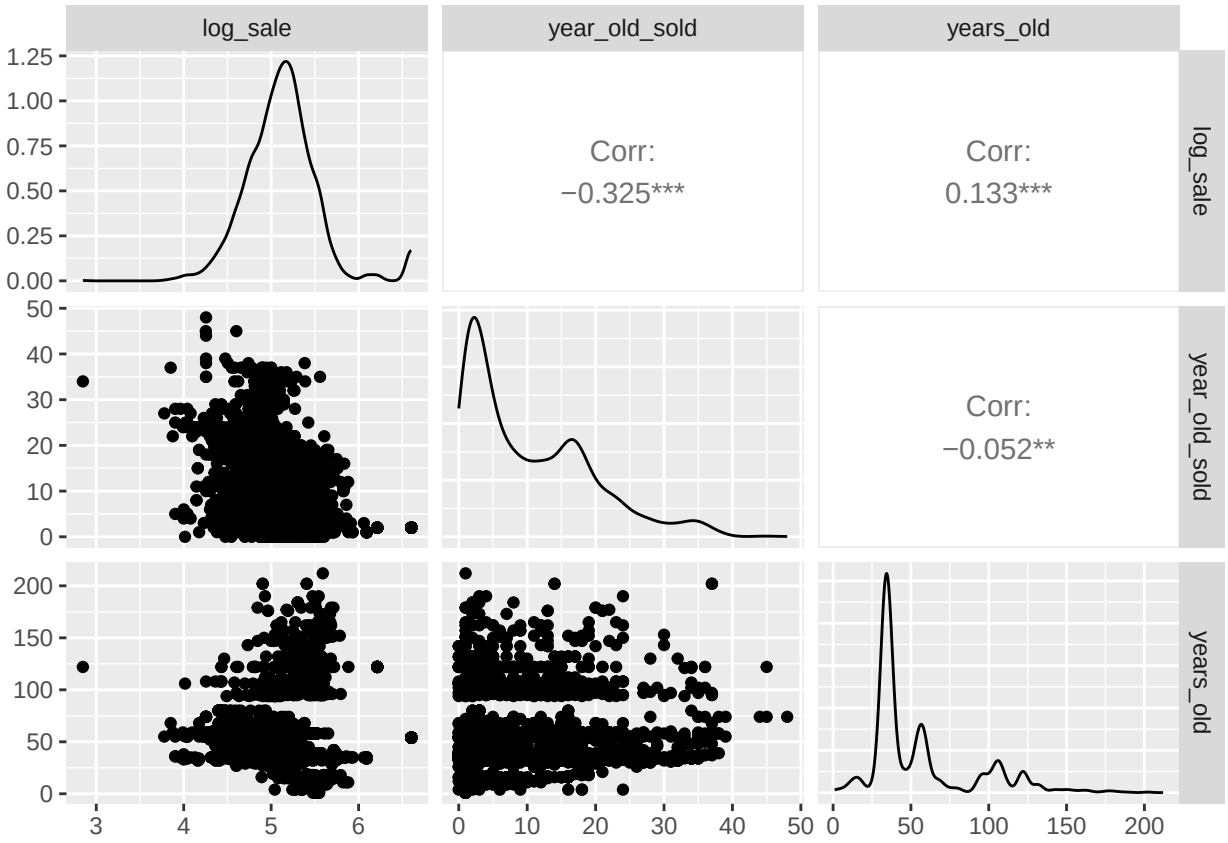
```
library(ggplot2)
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.2.3
```

```
ggpairs(select(property_data, log_sale, appraisal, building_area))
```




```
ggpairs(select(property_data, log_sale, year_old_sold, years_old))
```



Answer:

The graphs indicate that there is a strong correlation between sale price and appraisal value, building area, years since the property was sold, and years since the property was built. This gives insight into which variables to include in the model.

Part 2.3 (5 points): Now please go ahead and create a regression model for predicting the sales price of condos (i.e., the `sale_price` variable) from other explanatory variables. You can include however many terms in the model that you think leads to a good model (our model ended up using 5 variables and an interaction term, but you might be able to come up with a better model so please do what you think is best).

Also, in the R chunk below, print out a few relevant statistics about the model, and show appropriate diagnostic plots that give an indication of whether you model is fitting the data well (again, there is not one "right answer" here but just use a few of the methods we have discussed).

In the answer section, describe whether you think your model does a reasonable job fitting the data.

Note, in the process of creating your regression model you might fit several intermediate models, but only show your final model in the code chunk below.

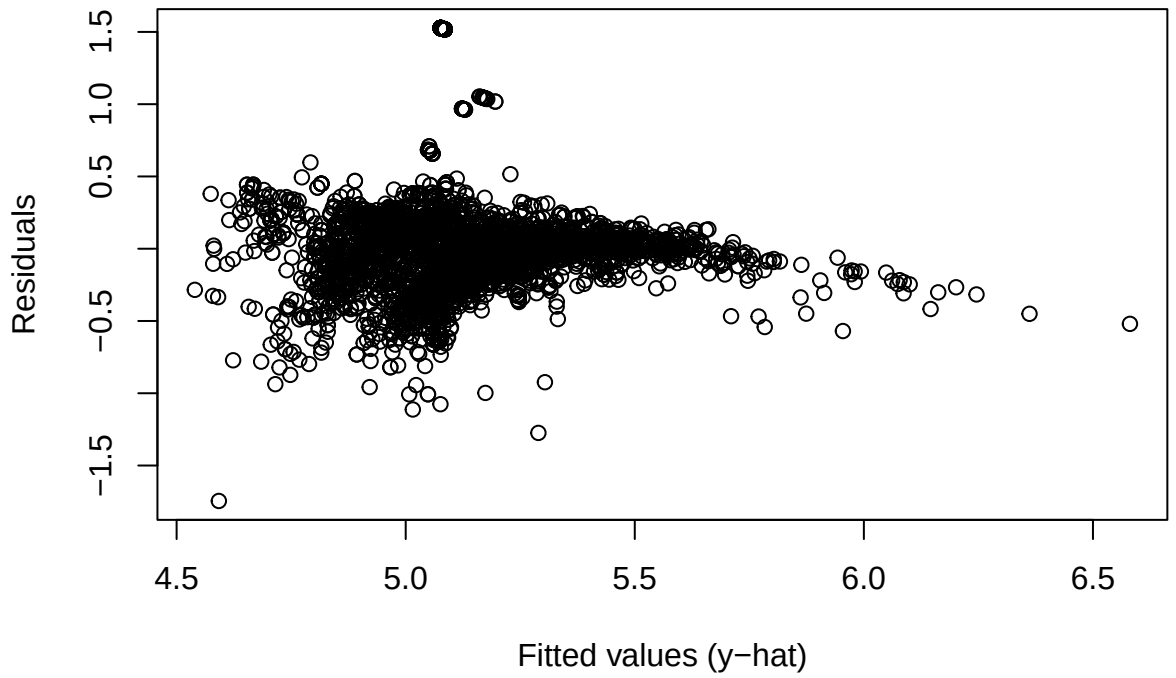
```
lm_property <- lm(log_sale ~ appraisal + building_area + year_old_sold,
                  data = property_data)

summary(lm_property)
```

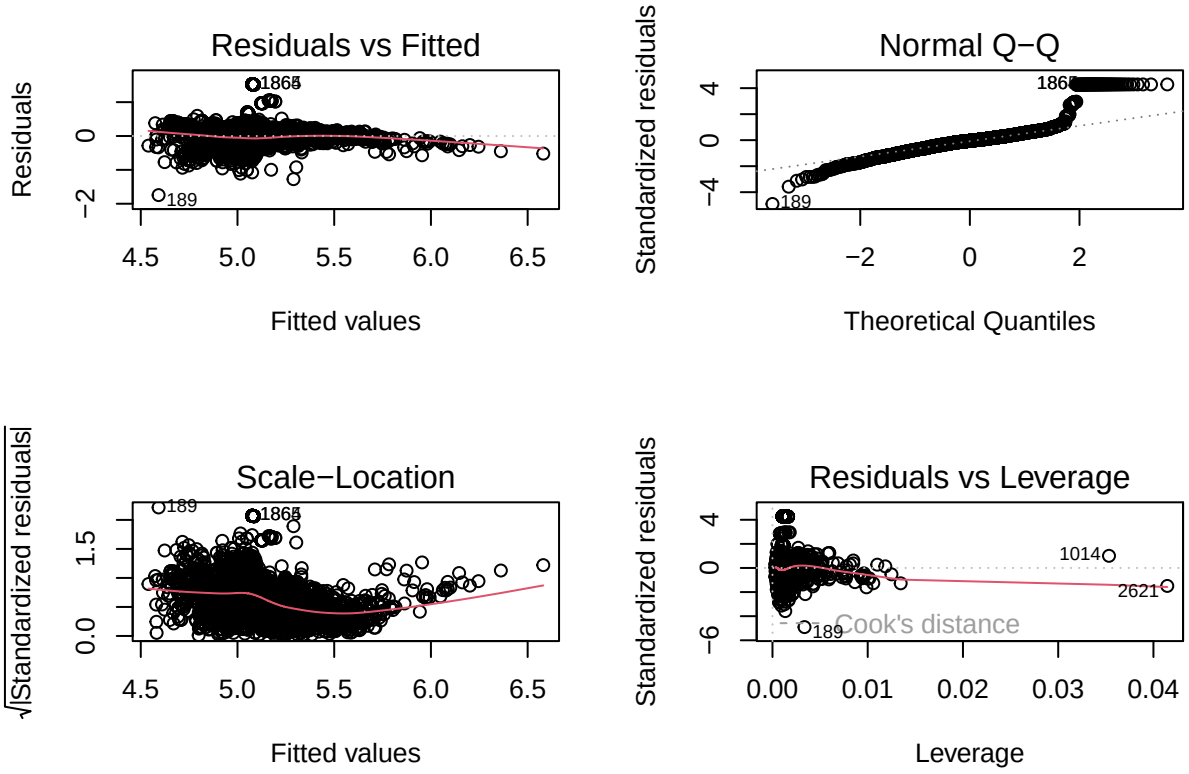
```
##
```

```
## Call:
## lm(formula = log_sale ~ appraisal + building_area + year_old_sold,
##     data = property_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.74521 -0.17179 -0.01722  0.11336  1.52666
##
## Coefficients:
##                      Estimate     Std. Error t value             Pr(>|t|)
## (Intercept)    5.08290345172  0.02051012091 247.824 < 0.0000000000000002 ***
## appraisal      0.00000175388  0.00000007333  23.916 < 0.0000000000000002 ***
## building_area -0.00010577124  0.00002153240  -4.912          0.000000946 ***
## year_old_sold -0.01335391722  0.00070632143 -18.906 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3564 on 3161 degrees of freedom
## Multiple R-squared:  0.2876, Adjusted R-squared:  0.2869
## F-statistic: 425.4 on 3 and 3161 DF,  p-value: < 0.00000000000000022
```

```
plot(lm_property$fitted.values, lm_property$residuals,
     xlab = "Fitted values (y-hat)",
     ylab = "Residuals")
```



```r
par(mfrow = c(2,2))
plot(lm_property)
```




```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.3
```

```
## Loading required package: carData
```

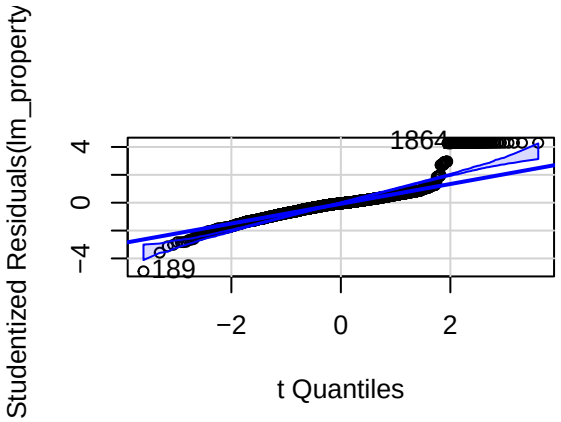
```
## Warning: package 'carData' was built under R version 4.2.3
```

```
##
## Attaching package: 'car'
```

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

```r
qqPlot(lm_property)
```

```
## [1]  189 1864
```



Answer

The diagnostic plots indicate that the model may be slightly heteroscedastic and somewhat non-normal, and that the data contains high-leverage points.

The summary statistics indicate that the appraisal, building_area, and year_old_sold variables are highly significant, but based on the R^2 statistic, they account for only 28.76% of the total sum of squares of sale prices.

Based on this, I think my model does at best a somewhat reasonable job of fitting the data.

Part 2.4 (3 points): Now that you've done your analyses, you are ready to make an offer on the condominium!

In the R chunk below, please use the model you created in part 2.3 to make a prediction for what the price is for the 896 Orange ST 5E condo. Hint: the code below extracts a data frame with just this relevant condo, and you can use the `predict()` using the `newdata` argument to get a prediction from your model.

Then in the answer section below please do the following:

1. Write down the predicted price for the property at 869 Orange St Unit 5E and whether you think this is a reasonable prediction or if you think it is too high or too low.
2. Write down what you think the best offer would be to put in for trying to buy this condominium and in 1-3 sentences explain how you came to this number. Remember, you definitely want your offer accepted, but you do not want to over pay so that you are "underwater" on the condominium and you would lose a lot of money if you had to sell it.

3. Enter these values in the homework 9 reflection and we will examine the distribution of offers the class comes up with. When you enter your numbers, please make sure to enter them a single numbers without any commas. For example if you think the fair price is the asking price of $475,000 then enter the number 475000 into the homework 3 reflection.

```
# get a data frame with just the 869 ORANGE ST #5E condo
orange_st_condo <- filter(property_data, street_number == "869 ORANGE ST #5E")
10^(predict(lm_property, newdata = orange_st_condo))
```

```
##        1
## 367719.5
```

Answer:

1. The predicted price is $367719.50. I think this is too low.
2. I think the best offer would be $410000. The best offer would be slightly above the appraisal value, which was $406700 in this case. Then the seller is receiving slightly more than the property is worth and the buyer can justify selling it for a similar price in the future if necessary.

Part 3: Thoughts on your final project (3 points)

Describe what you are thinking of doing for your final project and where you will get your data from. If you are not sure yet what you are going to do for your final project, that is fine and you can just say that. However, on homework 10 you will need to load the data you will use for your final project data to demonstrate that you have found relevant data, so please start preparing for this now. I encourage you to email TA's and ULA's, or to talk to me after class for guidance.

Answer

I haven't chosen a data set yet. I've been looking through the government data challenges linked on the Canvas site, but many are a bit advanced for me or outside the scope of this project.

Reflection (3 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 9