Homework 10

Ra-Zakee M.

Due Tuesday 11/23/2021

```
library(modelr)
options(na.action = na.warn)
library(tidyverse)
```

Question 1 (12 points)

Consider the attached file "CA-Vac-JanFeb.csv", which contains data about the number of vaccine doses administered in California each day in January and February 2021 (the "day" column simply numbers these days in order, starting with 1 for January 1). This data was pulled from https://github.com/datadesk/california-coronavirus-data/blob/master/cdph-vaccination-state-totals.csv. We'll try to model this data in a few different ways.

a. **Import this data. Suppose you first try a linear model where new_doses_administered = -15000 + 6000 * day. Plot the data as well as this line.**

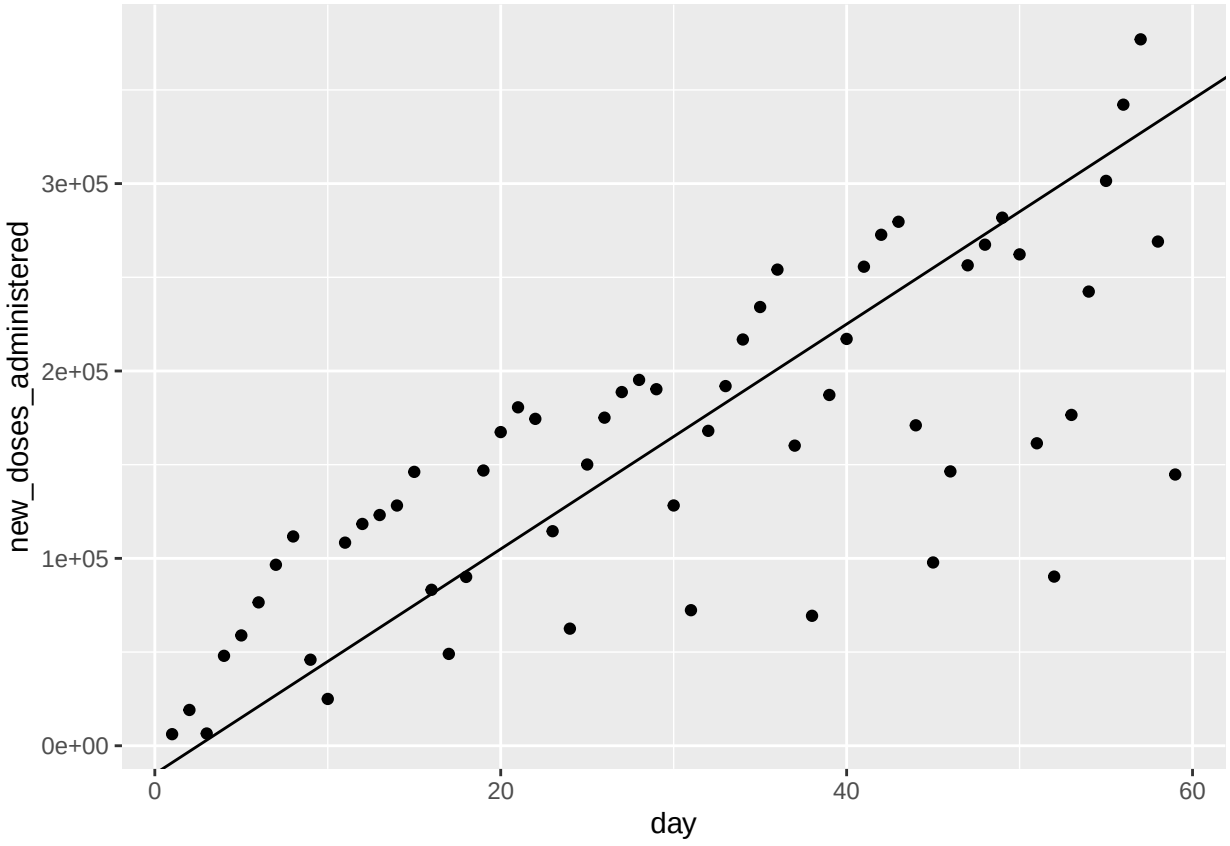
```
jan_feb <- read_csv("CA-Vac-Jan-Feb.csv")
```

```
## Rows: 59 Columns: 2
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (2): day, new_doses_administered
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jan_feb %>%
  ggplot(aes(x = day, y = new_doses_administered)) +
  geom_point() + geom_abline(slope = 6000, intercept = -15000)
```

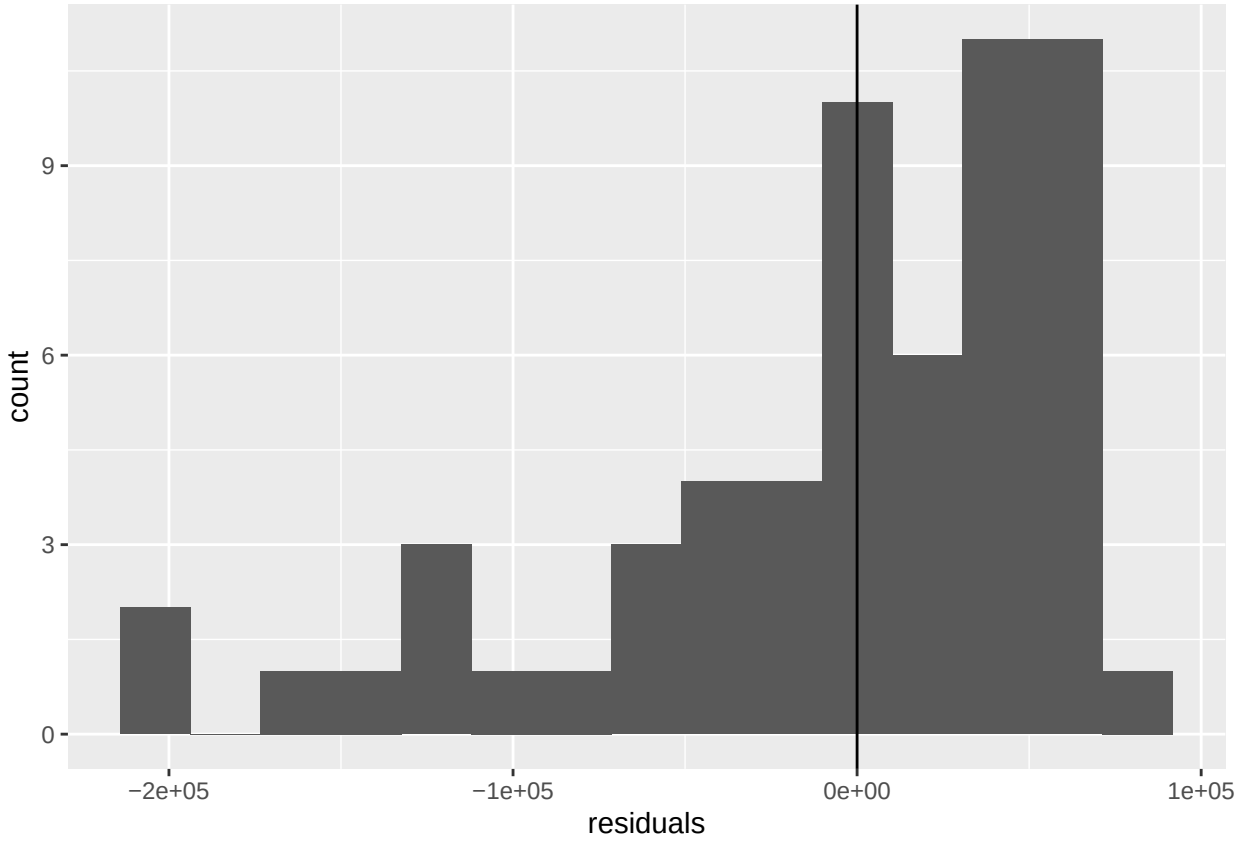


b. **Calculate the residuals for this model (do not use the add_residuals or gather_residuals functions).**

```
jan_feb_withresids <-jan_feb %>% mutate(prediction = -15000 + 6000*day) %>% mutate(residuals = new_doses
```

c. **Make a density plot or histogram of your residuals. Note that ggplots require a tibble or data frame, not a vector. Do these residuals appear equally likely to be positive or negative?**

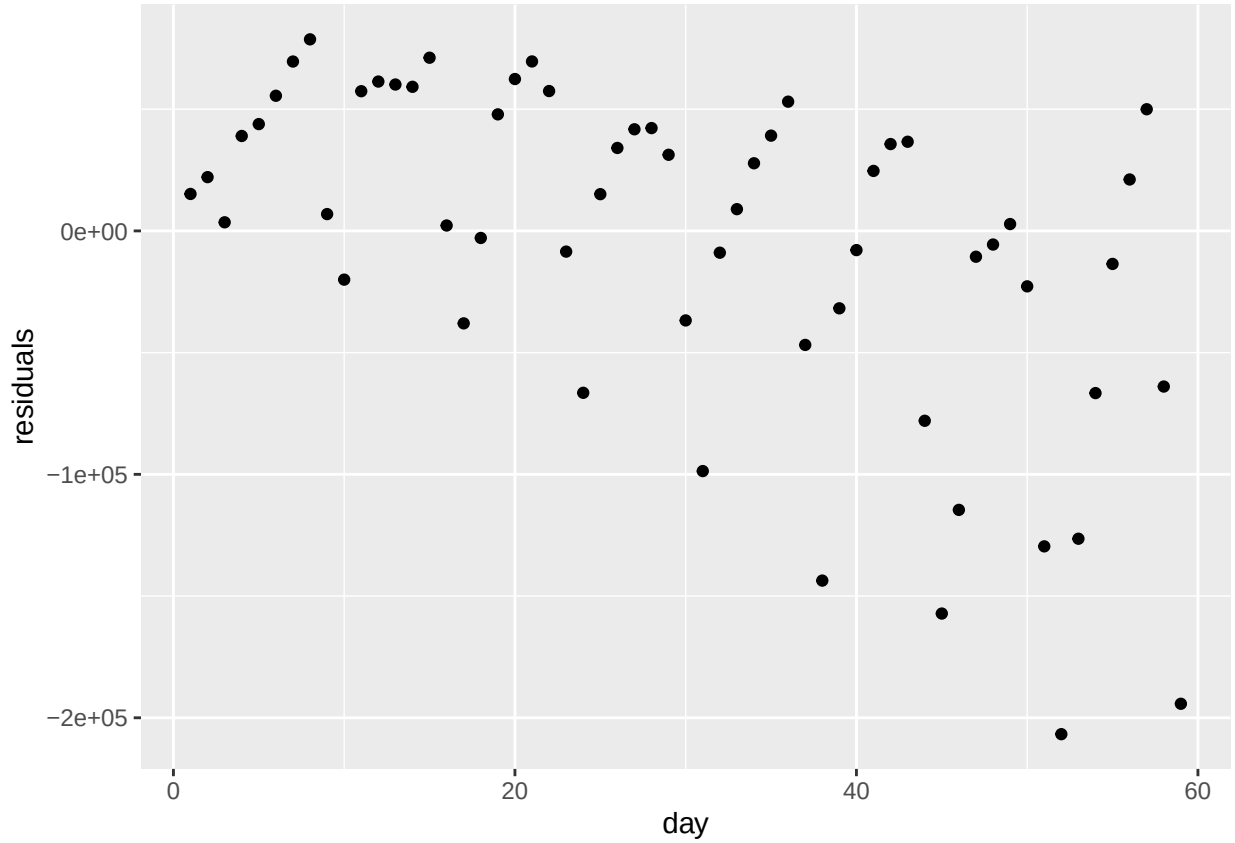
```
jan_feb_withresids %>% ggplot(aes(x = residuals)) + geom_histogram( bins = 15) + geom_vline(xintercept =
```



These residuals appear more likly to be positive.

d. **Make a scatterplot of day vs. residuals. Based on this scatterplot, does this linear model appear to be a good one?**

```
jan_feb_withresids %>% ggplot(aes(x= day, y = residuals)) + geom_point()
```



No it doesn't because the residuals dont appear to remain near zero and spread to greater negative extremes as the days progress.

Question 2 (15 points)

This question also uses the data in "CA-Vac-Jan-Feb.csv". Rather than minimizing the sum of the squares of the residuals, or minimizing the sum of the absolute values of the residuals, in this question you'll use a different optimization method: making the largest residual as small as possible. We'll specifically consider the absolute values of the residuals, since we're really concerned about minimizing the largest distance from a data point to the model line, and aren't concerned with whether that point is above or below the model line.

a. **For the linear model new_doses_administered = -15000 + 6000 * day, calculate the largest (in absolute value) of the residual values.**

```
jan_feb_withresids_abs <-jan_feb %>% mutate(prediction = -15000 + 6000*day) %>% mutate(residuals = abs(r

jan_feb_withresids_abs %>% summarise(max(residuals))
```

```
## # A tibble: 1 x 1
##   `max(residuals)`
##              <dbl>
## 1           206733
```

b. **Write a function that has a single parameter, a length 2 vector containing slope and intercept values. This function should return the maximum absolute value of a residual for the linear model with slope and intercept given in the length 2 vector. For example, when you pass this function c(-15000, 6000), it should return the same result as in part (a).**

```
max_res_func <- function(vect_param) {


jan_feb_withresids_abs_func <-jan_feb %>% mutate(prediction = vect_param[1] + vect_param[2]*day) %>% mu


  return(jan_feb_withresids_abs_func %>% summarise(max(residuals)))
}
```

```
max_res_func(c(-15000,6000))
```

```
## # A tibble: 1 x 1
##   `max(residuals)`
##              <dbl>
## 1           206733
```

c. **Using your function from (b), try a few different slope an intercept values to try to find a linear model where the maximum absolute value of the residuals is smaller than for the model given by c(-15000, 6000).**

```
max_res_func(c(-39000,5000))
```

```
## # A tibble: 1 x 1
##   `max(residuals)`
##              <dbl>
## 1           130983
```

```
max_res_func(c(-61000,5000))
```

```
## # A tibble: 1 x 1
##   `max(residuals)`
##              <dbl>
## 1           152983
```

```
max_res_func(c(-61000,5400))
```

```
## # A tibble: 1 x 1
##   `max(residuals)`
##              <dbl>
## 1           130183
```

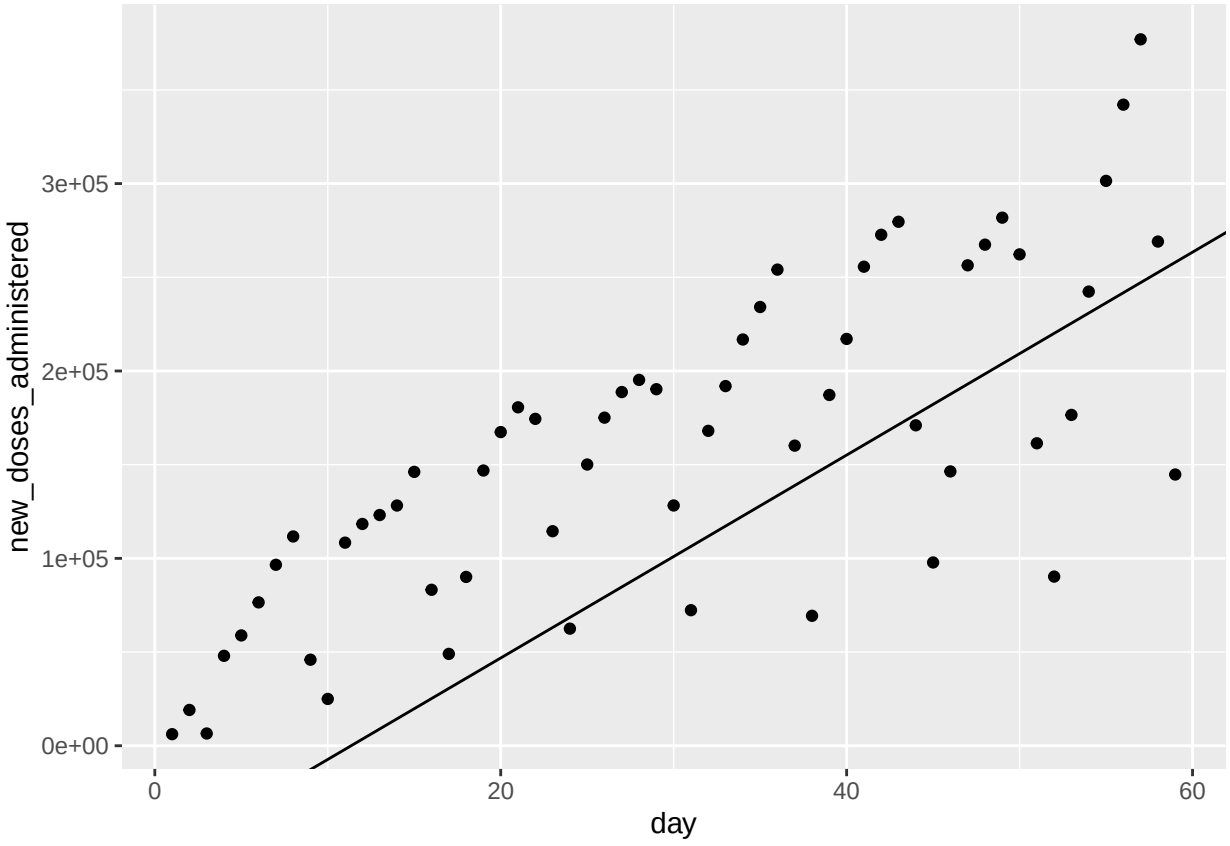
d. **Find the values for slope and intercept that minimize the maximum absolute value of the residuals.**

```
optimal_coeffs <- optim( c(-39000,5000), ## a guess that is near where you think the optimal inputs wil
       max_res_func, # function whose output you are trying to minimize
  )
optimal_coeffs
```

```
## $par
## [1] -61440.227   5414.041
##
## $value
## [1] 129822.9
##
## $counts
## function gradient
##      175       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

e. **Plot the line given by the slope and intercept you found in the previous part on top of a scatterplot of the data. Does this line appear to be a good model? Does assessing models according to the size of the largest residual make sense as a method?**

```
jan_feb %>%
  ggplot(aes(x = day, y = new_doses_administered)) +
  geom_point() + geom_abline(slope = 5414.041, intercept = -61440.227)
```



Question 3 (14 points)

This question continues using the data in "CA-Vac-Jan-Feb.csv"

a. (6 points) **Find the linear model that minimizes the sum of the absolute values of the residuals. Plot the line for this model on top of a scatterplot of the data points.**

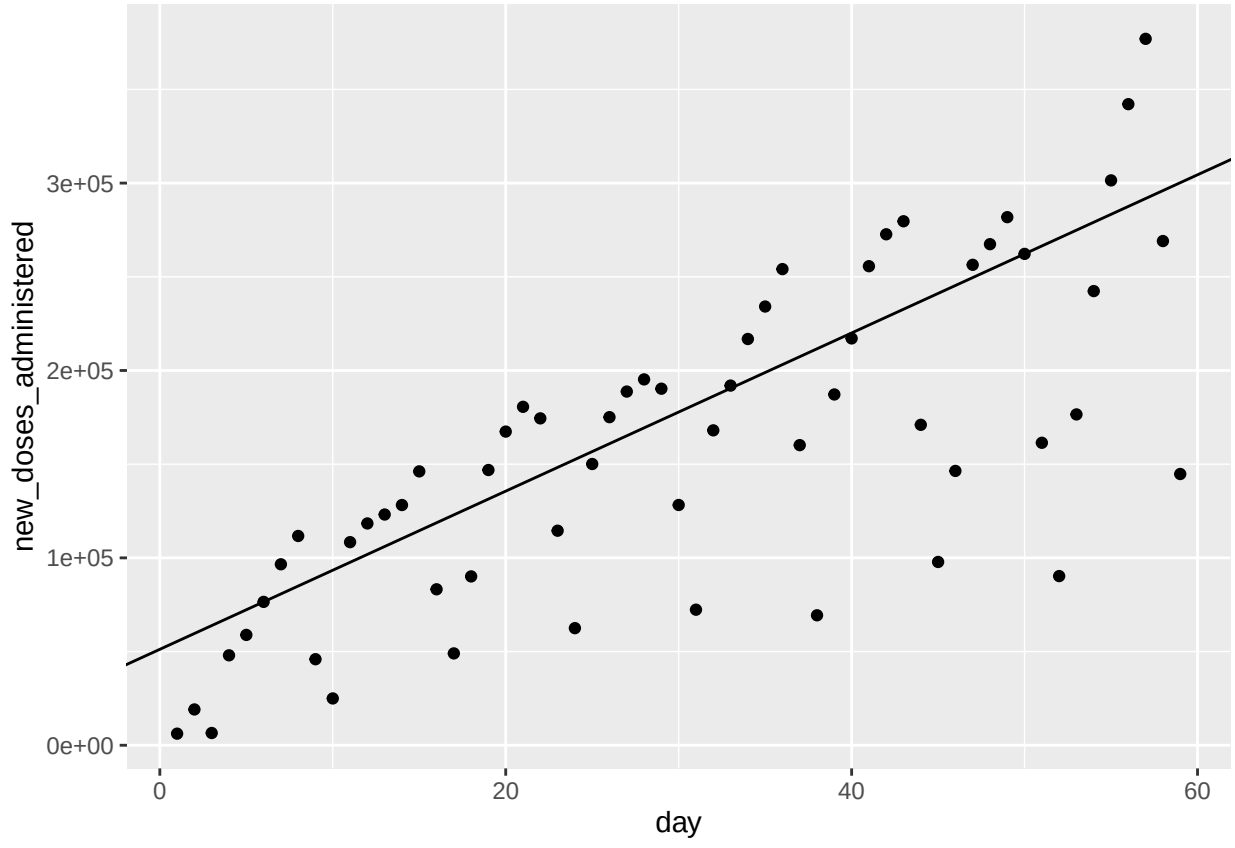
```
max_res_func_3a <- function(vect_param) {


jan_feb_withresids_abs_func <-jan_feb %>% mutate(prediction = vect_param[1] + vect_param[2]*day) %>% mu


  return(jan_feb_withresids_abs_func %>% summarise(sum(residuals)))
}
```

```
optimal_coeffs_a <- optim( c(-61440.227,5414.041), ## a guess that is near where you think the optimal
       max_res_func_3a, # function whose output you are trying to minimize
  )
optimal_coeffs_a
```

```
## $par
```

```
## [1] 51192.736  4220.544
##
## $value
## [1] 2648372
##
## $counts
## function gradient
##      149       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

```
jan_feb %>%
  ggplot(aes(x = day, y = new_doses_administered)) +
  geom_point() + geom_abline(slope = 4220.544, intercept = 51192.736)
```



b. (5 points) **Find the linear model that minimizes the sum of the squares of the residuals. Plot the line for this model on top of a scatterplot of the data points.**

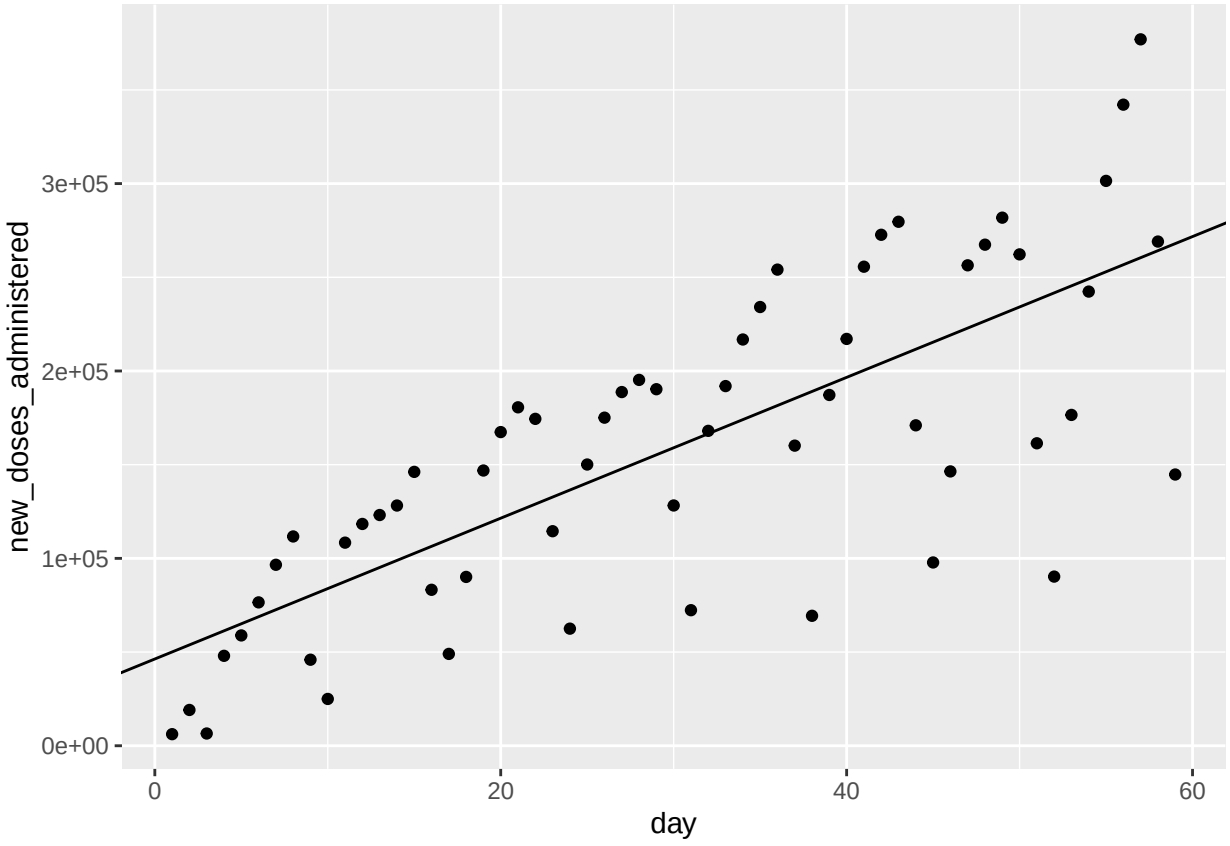
```
max_res_func_3b <- function(vect_param) {

```

```
jan_feb_withresids_abs_func <-jan_feb %>% mutate(prediction = vect_param[1] + vect_param[2]*day) %>% mu


  return(jan_feb_withresids_abs_func %>% summarise(sum(residuals^2)))
}
```

```
optimal_coeffs_b <- optim( c(51192.736,4220.544), ## a guess that is near where you think the optimal i
       max_res_func_3b, # function whose output you are trying to minimize
  )
optimal_coeffs_b
```

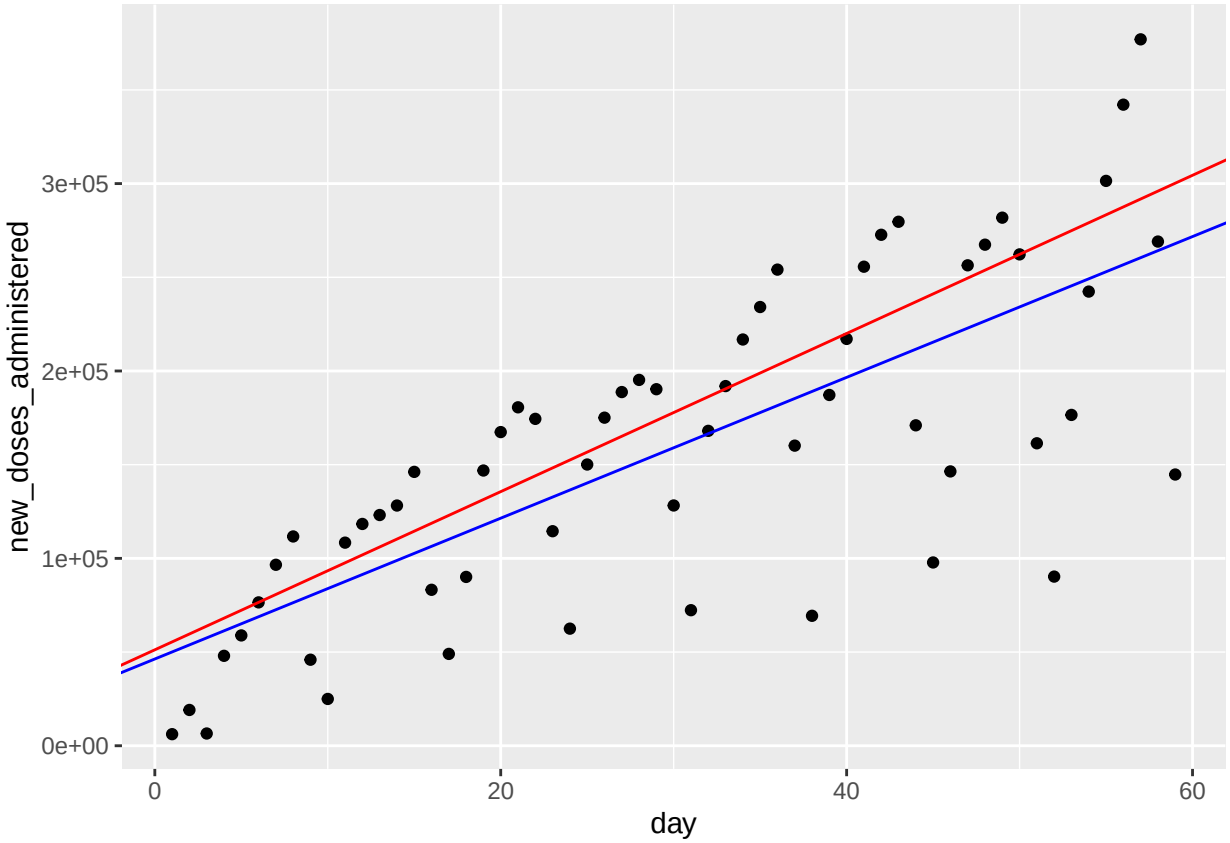
```
## $par
## [1] 46299.577  3757.205
##
## $value
## [1] 188972241430
##
## $counts
## function gradient
##       63       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

```
jan_feb %>%
  ggplot(aes(x = day, y = new_doses_administered)) +
  geom_point() + geom_abline(slope = 3757.205, intercept = 46299.577 )
```



c. (3 points) **Plot both of your model lines from the previous two parts in the same plot using different colors, and describe the similarities/differences between them.**

```
jan_feb %>%
  ggplot(aes(x = day, y = new_doses_administered)) +
  geom_point() +
  geom_abline(slope = 3757.205, intercept = 46299.577, col = "blue") +
  geom_abline(slope = 4220.544, intercept = 51192.736, col = "red")
```



sum of residuals (red) is steeper than sum of residuals squared (blue). Sum of residuals also has a higher intercept than sum of residuals squared. Both lines have a positive slope though.

Question 4 (6 points)

For each part below, explain why the two commands have the same output or why they have different outputs.

```
ca_vac_day_of_week <- read_csv("CA-Vac-Day-of-week.csv")
```

```
## Rows: 478 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): day_of_week
## dbl (2): day, new_doses_administered

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

a.

```
data_grid(ca_vac_day_of_week, day)
```

```
## # A tibble: 478 x 1
##      day
##    <dbl>
##  1     1
##  2     2
##  3     3
##  4     4
##  5     5
##  6     6
##  7     7
##  8     8
##  9     9
## 10    10
## # ... with 468 more rows
```

```
select(ca_vac_day_of_week, day)
```

```
## # A tibble: 478 x 1
##      day
##    <dbl>
##  1     1
##  2     2
##  3     3
##  4     4
##  5     5
##  6     6
##  7     7
##  8     8
##  9     9
## 10    10
## # ... with 468 more rows
```

day alone is a primary key in this data set so every every row has a unique value for day there are 478 days, so select isolates that column giving us the day numbers in sequential order. data grid passes day into expand which finds all combinations of day which is just 1 to 478 since no other factor is passed into expand.

b.

```
data_grid(ca_vac_day_of_week, day_of_week)
```

```
## # A tibble: 7 x 1
##   day_of_week
##   <chr>
## 1 Friday
## 2 Monday
## 3 Saturday
## 4 Sunday
## 5 Thursday
## 6 Tuesday
## 7 Wednesday
```

```
select(ca_vac_day_of_week, day_of_week)
```

```
## # A tibble: 478 x 1
##    day_of_week
##    <chr>
##  1 Monday
##  2 Tuesday
##  3 Wednesday
##  4 Thursday
##  5 Friday
##  6 Saturday
##  7 Sunday
##  8 Monday
##  9 Tuesday
## 10 Wednesday
## # ... with 468 more rows
```

data grid passes day into expand which finds all combinations of day which is just the 7 days of the week since no other factor is passed into expand. select isolat the column without looking at distinct values meaning all 478 rows for day_of_week will be returned.

c.

```
data_grid(ca_vac_day_of_week, day_of_week)
```

```
## # A tibble: 7 x 1
##   day_of_week
##   <chr>
## 1 Friday
## 2 Monday
## 3 Saturday
## 4 Sunday
## 5 Thursday
## 6 Tuesday
## 7 Wednesday
```

```
ca_vac_day_of_week %>% group_by(day_of_week) %>% summarize()
```

```
## # A tibble: 7 x 1
##   day_of_week
##   <chr>
## 1 Friday
## 2 Monday
## 3 Saturday
## 4 Sunday
## 5 Thursday
## 6 Tuesday
## 7 Wednesday
```

data grid passes day into expand which finds all combinations of day which is just the 7 days of the week since no other factor is passed into expand. The second code block groups the rows by the distinct days of

the week and then summarise tells us how those groups were definied again returning to us the 7 days of the week.

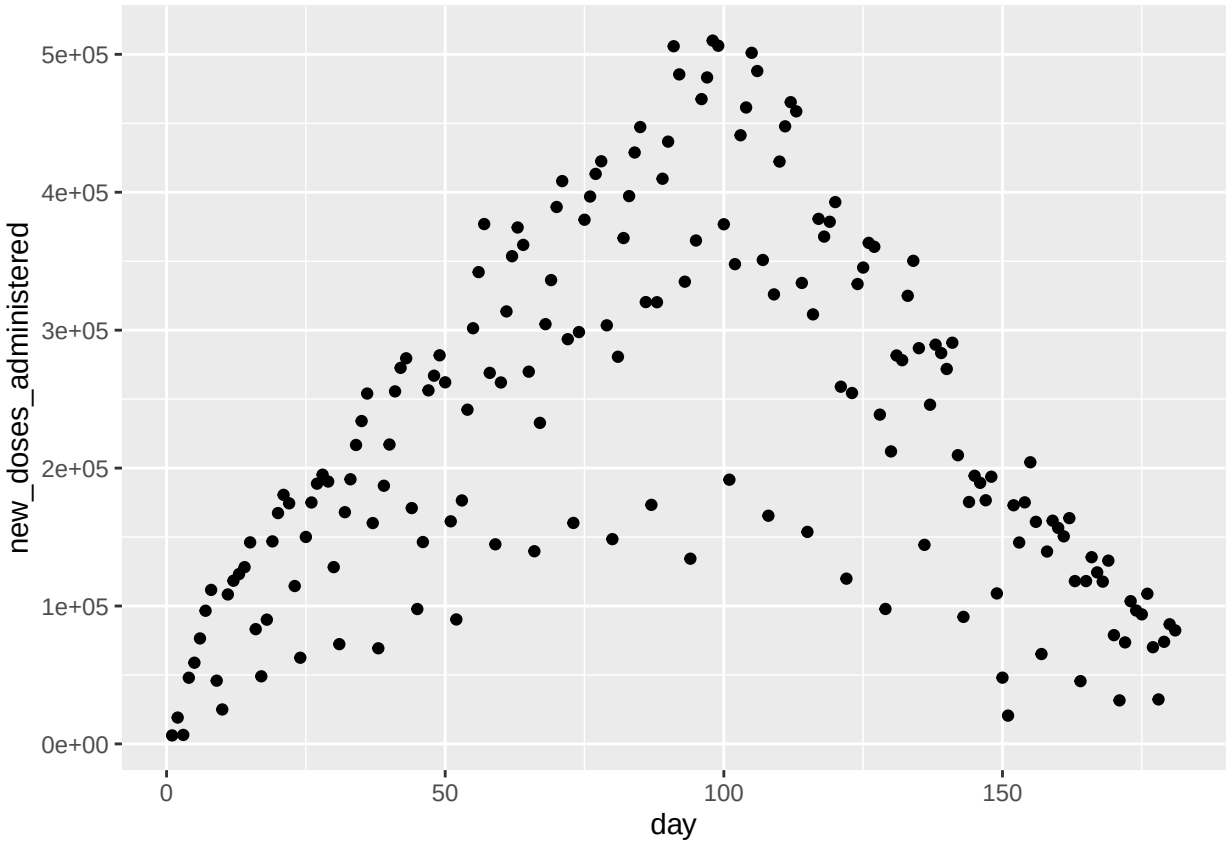
Question 5 (15 points)

Consider the data in "Ca-Vac-Jan-Jun.csv"

a. **Import the data and make a scatterplot of it. Explain why quadratic models make a better family of of models to consider for this data than linear models.**

```
read_csv("Ca-Vac-Jan-Jun.csv") %>% ggplot(aes(x = day, y = new_doses_administered)) + geom_point()
```

```
## Rows: 181 Columns: 2

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (2): day, new_doses_administered

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```



A quadratic model would be a better family of models to choose from because of the way the data curves around 100th day. The data increases at a decreasing rate then decreases at an increasing rate around 100.

b. **Make an additional column in your data set for the square of the day. Use the lm function to find the best quadratic model for new_doses_administered in terms of day and day^2. What are the intercept, the coefficient for day, and the coefficient from day^2 in this model?**

```
with_square <- read_csv("Ca-Vac-Jan-Jun.csv") %>% mutate(day_square = day^2)
```

```
## Rows: 181 Columns: 2
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (2): day, new_doses_administered
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
lm(new_doses_administered ~ day_square+ day,with_square)
```

```
##
## Call:
## lm(formula = new_doses_administered ~ day_square + day, data = with_square)
##
## Coefficients:
## (Intercept)   day_square          day
##   -13259.68       -42.03      7753.83
```

c. **Add both the predictions of this model and the residuals of this model onto your data set.**

```
with_square %>% mutate(predict = -13259.68 -42.03*day_square+7753.83*day) %>% mutate(residuals = new_do
```

```
## # A tibble: 181 x 5
##      day new_doses_administered day_square predict residuals
##    <dbl>                  <dbl>      <dbl>   <dbl>     <dbl>
##  1     1                   6173          1  -5548.    11721.
##  2     2                  19096          4   2080.    17016.
##  3     3                   6531          9   9624.    -3093.
##  4     4                  47983         16  17083.    30900.
##  5     5                  58871         25  24459.    34412.
##  6     6                  76493         36  31750.    44743.
##  7     7                  96559         49  38958.    57601.
##  8     8                 111665         64  46081.    65584.
##  9     9                  45862         81  53120.    -7258.
## 10    10                  24978        100  60076.   -35098.
## # ... with 171 more rows
```

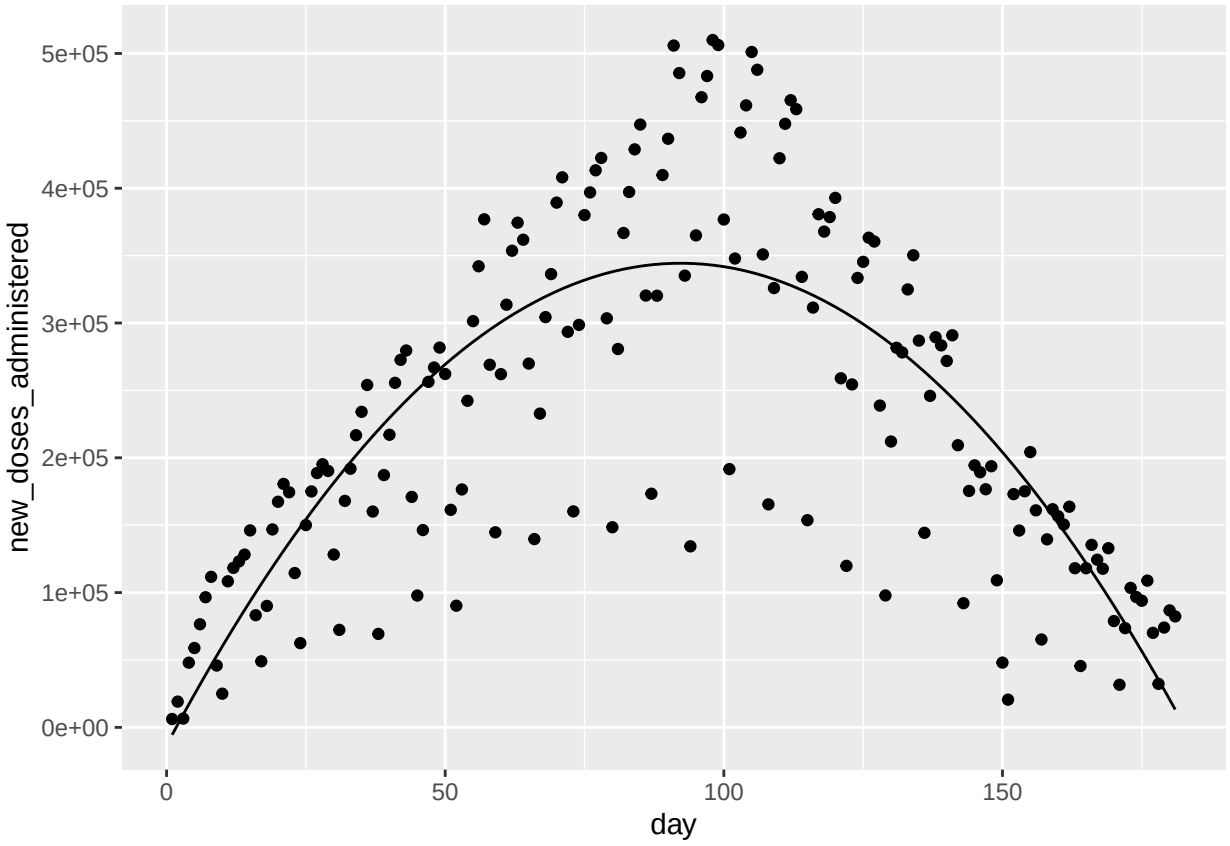
d. **Draw your quadratic model on top of your scatterplot of the data points. Reflect on whether this is a good model for this data or not.**

```
read_csv("Ca-Vac-Jan-Jun.csv") %>% ggplot(aes(x = day, y = new_doses_administered)) + geom_point()+ sta
```

```
## Rows: 181 Columns: 2
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (2): day, new_doses_administered
```

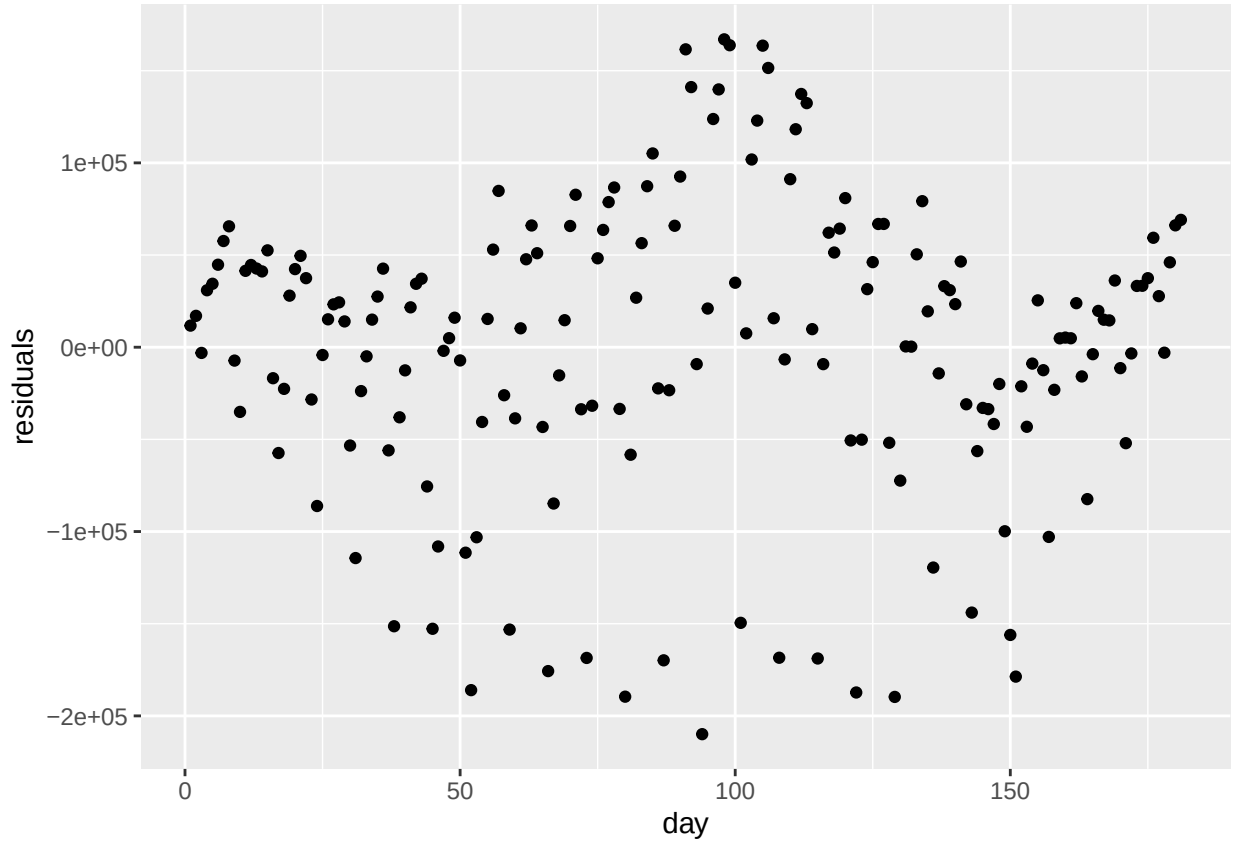
```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```



it seems like this might be a good model considering how the data is layed out to begin with. It seems the model might miss a substantial amount of points regardless of where the vertex is because of how variant the data is near the 100th day.

e. **Make a scatterplot of day vs. the residuals of your quadratic model. Reflect on what this tells you about your model.**

```
with_square %>% mutate(predict = -13259.68 -42.03*day_square+7753.83*day) %>% mutate(residuals = new_do
```



This plot tells us that that the residual values are much les extreme at the first and last few days, but the middle period is where the greatest extremes are present which means the model is most in acurate during the middle period but actuare around the edges.

Question 6 (14 points)

a. (4 points) **Consider the following tibble. Add a column called "Day" (hint: use the row_number() function). Pivot so that this data is tidy, with a column for vaccine_type and a column for num_doses. Make sure your vaccine_type column only says "jj", "moderna", or "pfizer" (this is a really good review of concepts from earlier this semester in preparation for your final project!)**

```
oct_vaccine <- read_csv("https://raw.githubusercontent.com/datadesk/california-coronavirus-data/master/
  select(date, new_pfizer_doses, new_moderna_doses, new_jj_doses) %>%
  filter(str_detect(date, "2021-10"))
```

```
## Rows: 486 Columns: 18

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (17): doses_administered, new_doses_administered, pfizer_doses, new_pfi...
## date  (1): date
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
oct_alt <- oct_vaccine %>%
  mutate(Day = row_number()) %>%
  rename( jj=new_jj_doses, moderna = new_moderna_doses, pfizer = new_pfizer_doses) %>%
  pivot_longer(cols = !c(date, Day), names_to = "New_Doses_Type") %>%
  rename(number_of_doses = value)
```

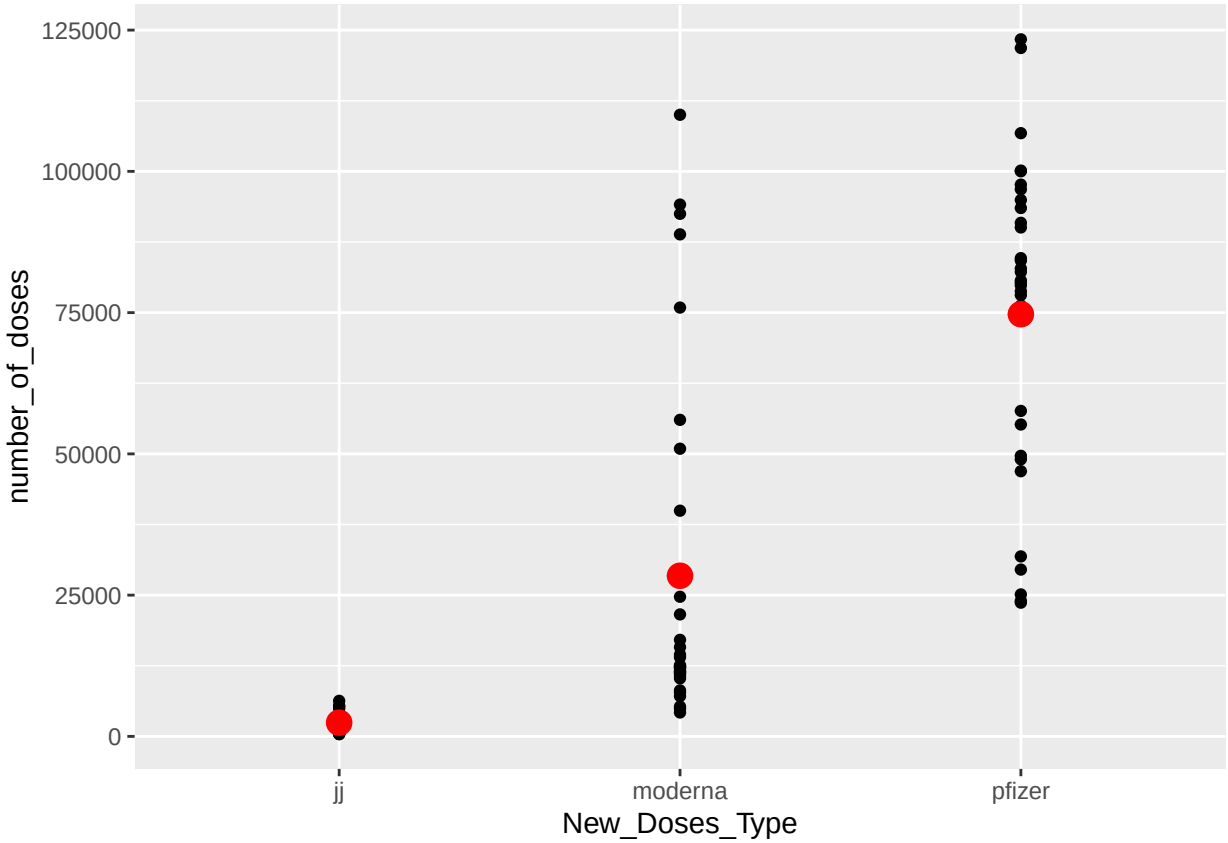
b. (4 points) **Make a linear model that attempts to predict the number of daily doses given the vaccine type. What does your model predict as the daily doses for jj, moderna, and pfizer vaccines?**

```
mod2 <- lm(number_of_doses ~ New_Doses_Type, data = oct_alt)

grid <- oct_alt %>% data_grid(New_Doses_Type) %>% add_predictions(mod2)
grid
```

```
## # A tibble: 3 x 2
##   New_Doses_Type   pred
##   <chr>           <dbl>
## 1 jj              2432.
## 2 moderna        28428.
## 3 pfizer         74722.
```

c. (4 points) **Plot your model from the previous part, with vaccine_type on the x-axis and num_doses on the y-axis. Include your original data points along with the model predictions.**

```
ggplot(oct_alt, aes(x = New_Doses_Type)) +
  geom_point(aes(y = number_of_doses)) +
  geom_point(data = grid, aes(y = pred), color = "red", size = 4)
```



d. (2 points) **Compute the average daily number of moderna doses given in October, and do the same for jj and pfizer as well. Compare them to the predictions of your model.**

```
oct_alt %>% group_by(New_Doses_Type)%>% summarise( average_daily_number = mean(number_of_doses))
```

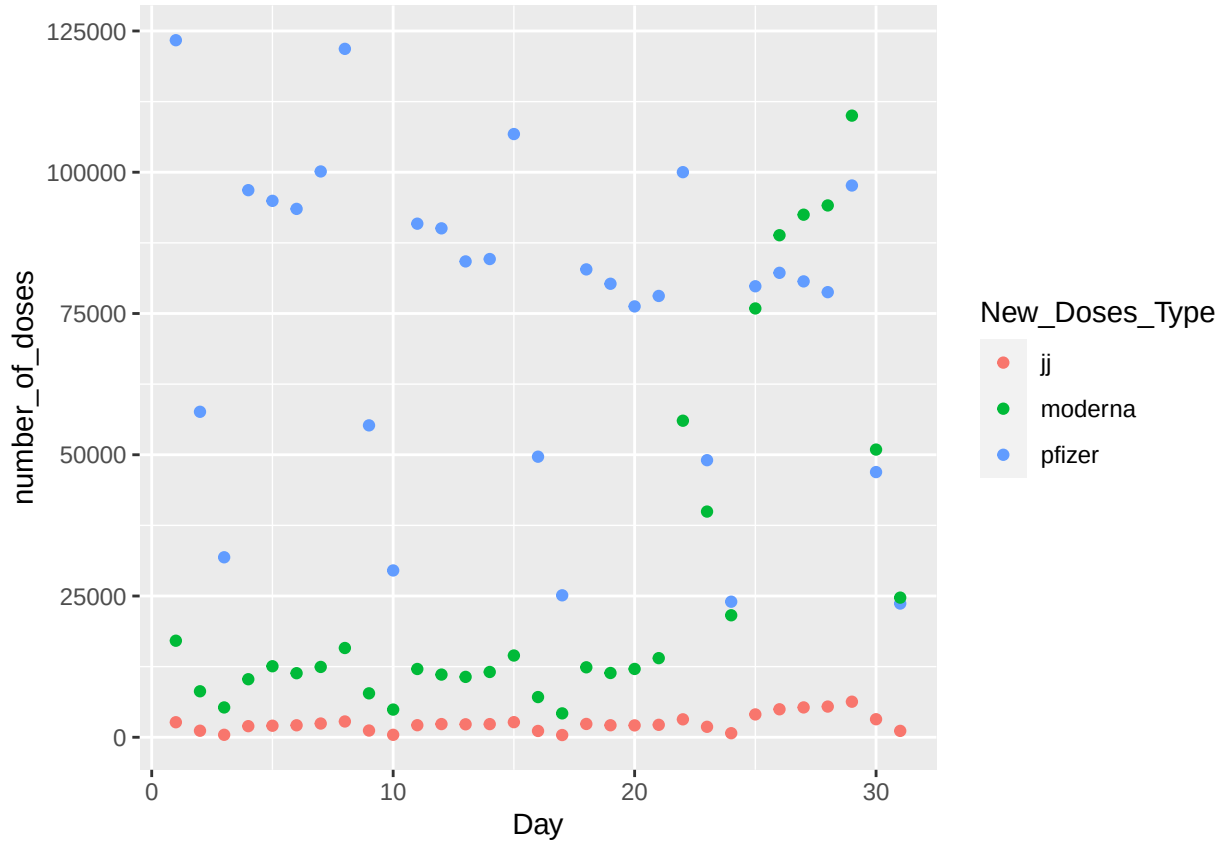
```
## # A tibble: 3 x 2
##   New_Doses_Type average_daily_number
##   <chr>                         <dbl>
## 1 jj                            2432.
## 2 moderna                      28428.
## 3 pfizer                       74722.
```

Question 7 (12 points)

In this question, use the pivoted table you created in part (a) of the previous question. We'll create a model for the daily number of vaccine doses administered using two predictors: day and vaccine type.

a. **Before creating the model, make a plot that shows the relationship between day, vaccine_type, and num_doses. What do you observe?**

```
oct_alt %>% ggplot(aes(x = Day, y = number_of_doses)) + geom_point(aes(color = New_Doses_Type))
```



We can observe how day impacts number of doses and how across time type of vaccine affects number of doses. JJ is always the lowest number of doses, moderna generally comes in second and pfizer usually has the most, as the days increase pfizer and moderna swap places and number of vaccine doses falls generally.

b. **Create a linear model for the number of daily vaccine doses given in October in terms of both day of the month and vaccine type. Be sure you allow for interactions between day and vaccine type. How many coefficients does your model have?**

```
mod3 <- lm(number_of_doses ~ New_Doses_Type + Day, data = oct_alt)
mod3
```

```
##
## Call:
## lm(formula = number_of_doses ~ New_Doses_Type + Day, data = oct_alt)
##
## Coefficients:
##           (Intercept)  New_Doses_Typemoderna   New_Doses_Typepfizer
##               -6205.4               25996.2               72289.9
##                   Day
##                 539.8
```

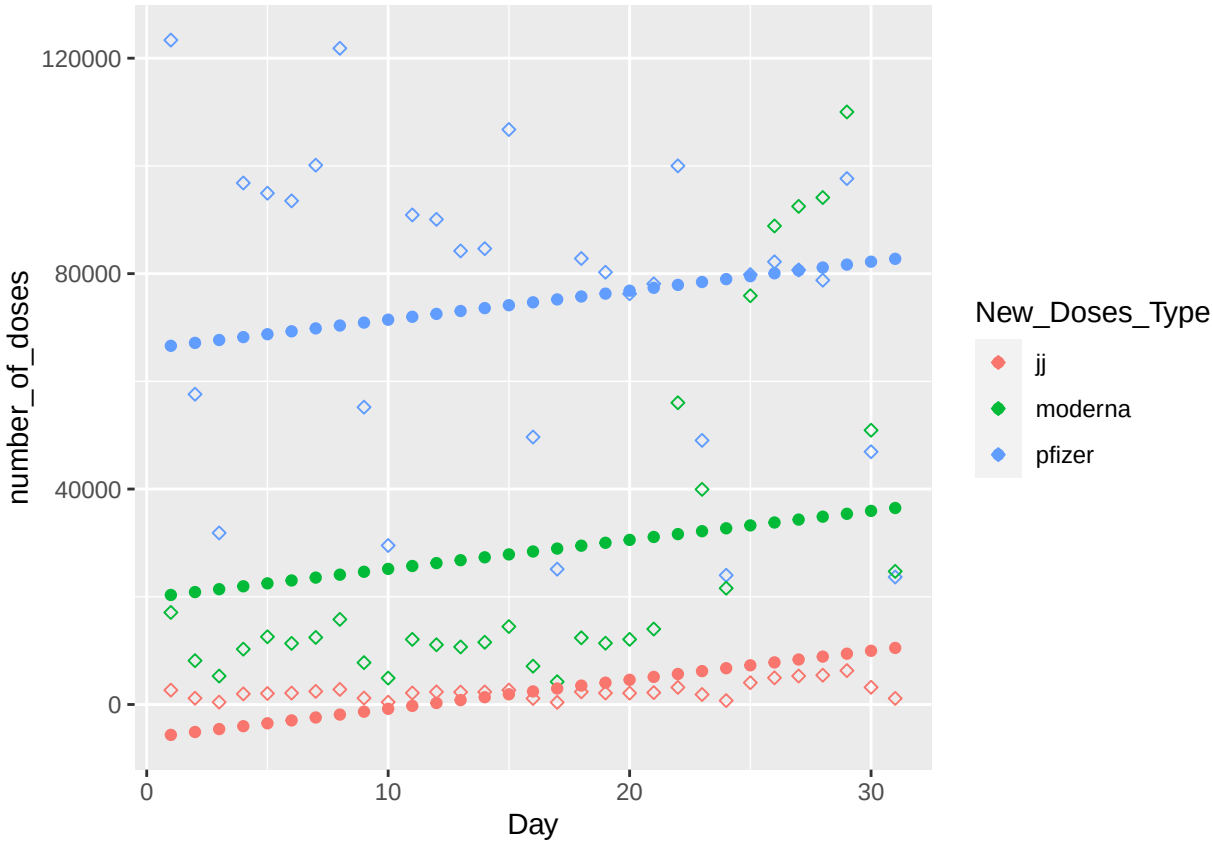
my model has 4 coefficients

c. **Make a tibble containing the predictions for each day and vaccine_type. Plot these predictions (and your original data points).**

```
oct_alt %>% mutate(predict = -6184.4  + 25977.4*(New_Doses_Type == "moderna") + 72245.1*(New_Doses_Type
```

```
## # A tibble: 93 x 5
##    date         Day New_Doses_Type number_of_doses predict
##    <date>     <int> <chr>                    <dbl>   <dbl>
##  1 2021-10-01     1 pfizer                  123365  66599.
##  2 2021-10-01     1 moderna                  17088  20331.
##  3 2021-10-01     1 jj                        2661  -5646
##  4 2021-10-02     2 pfizer                   57605  67138.
##  5 2021-10-02     2 moderna                   8138  20870.
##  6 2021-10-02     2 jj                        1161  -5108.
##  7 2021-10-03     3 pfizer                   31853  67676.
##  8 2021-10-03     3 moderna                   5277  21408.
##  9 2021-10-03     3 jj                         435  -4569.
## 10 2021-10-04     4 pfizer                   96829  68214.
## # ... with 83 more rows
```

```
oct_alt %>% mutate(predict = -6184.4  + 25977.4*(New_Doses_Type == "moderna") + 72245.1*(New_Doses_Type
```



d. **Plot the residuals of your previous model; your plot should have three facets, one for each vaccine_type. What do you observe?**

no credit

Question 8 (12 points)

Sometimes a linear or quadratic model isn't good enough for a particular data set, such as understanding the relationship between price and carat in the diamonds data set. Let's try a polynomial model instead. That is, if y is price and x is carat, we want a model with an equation that looks like:

$$y = c_1 * x^{c_2}$$

(in our previous models, c_2, the exponent of x, was always 1 or 2; this model lets that exponent vary).

By taking the logarithm of both sides, we get the equation:

$$\log(y) = \log(c_0) + c_1 \log(x)$$

This means there is a *linear* relationship between $\log(y)$ and $\log(x)$ whenever y and x have a polynomial relationship.

Let's see if there is a linear relationship between $\log(price)$ and $\log(carat)$ in the diamonds data set.

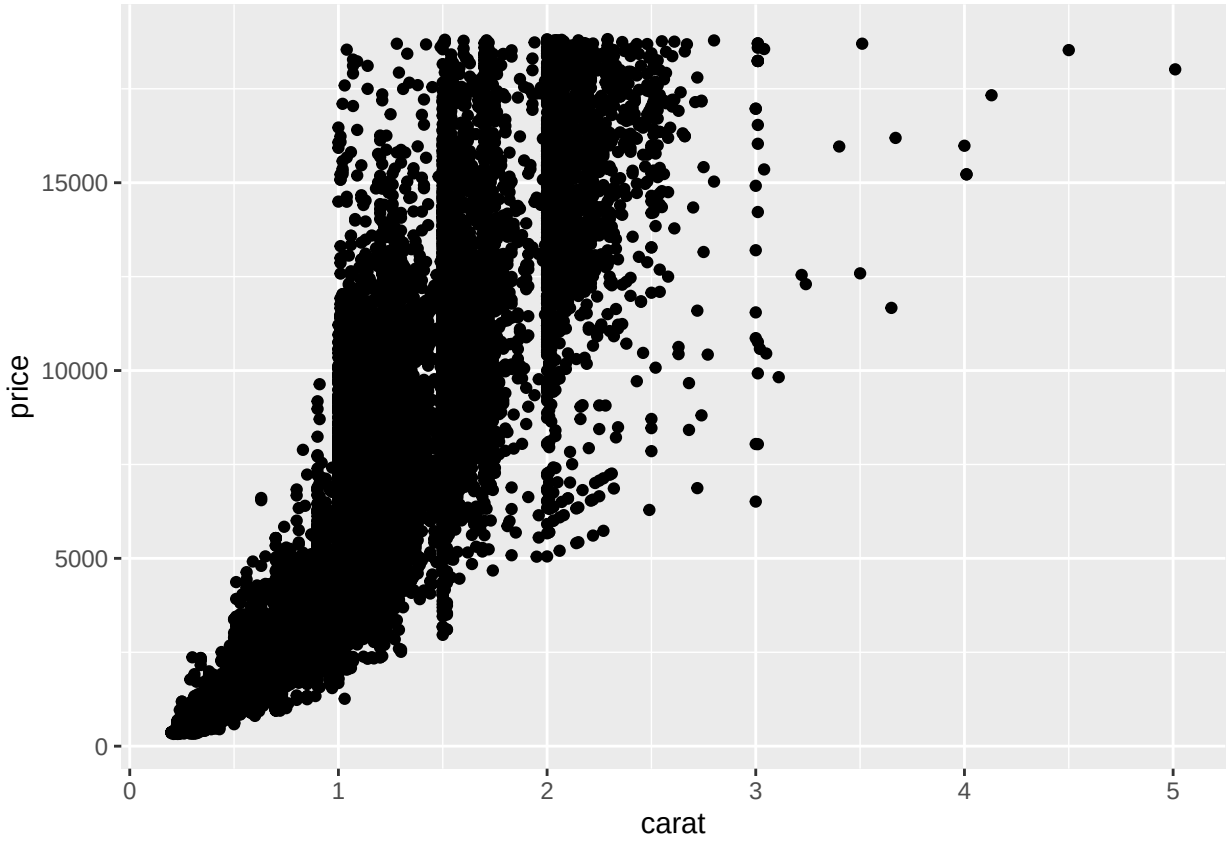
a. **Add columns for the logarithm of price and the logarithm of carat to the diamonds data set. The R function for the (base 2) logarithm is log2().**

```
diamonds %>% mutate(log_two_price = log2(price), log_two_carat = log2(carat))
```

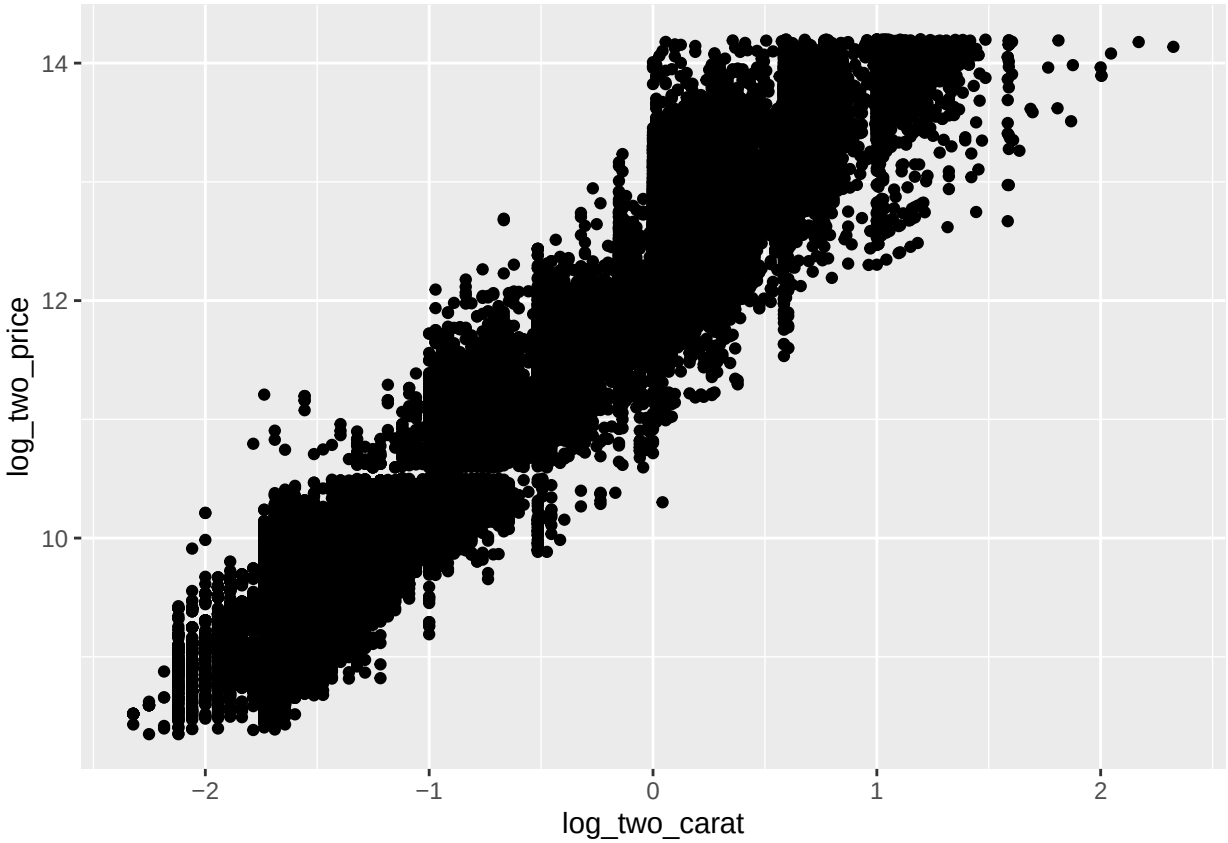
```
## # A tibble: 53,940 x 12
##    carat cut       color clarity depth table price     x     y     z log_two_price
##    <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>         <dbl>
##  1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43          8.35
##  2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31          8.35
##  3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31          8.35
##  4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63          8.38
##  5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75          8.39
##  6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48          8.39
##  7  0.24 Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47          8.39
##  8  0.26 Very Good H     SI1      61.9    55   337  4.07  4.11  2.53          8.40
##  9  0.22 Fair      E     VS2      65.1    61   337  3.87  3.78  2.49          8.40
## 10  0.23 Very Good H     VS1      59.4    61   338  4     4.05  2.39          8.40
## # ... with 53,930 more rows, and 1 more variable: log_two_carat <dbl>
```

b. **Make a scatterplot of log(carat) vs. log(price). Does this data look more linear than the original data?**

```
diamonds %>% mutate(log_two_price = log2(price), log_two_carat = log2(carat)) %>% ggplot(aes(x = carat,
```




```
diamonds %>% mutate(log_two_price = log2(price), log_two_carat = log2(carat)) %>% ggplot(aes(x = log_tw
```



Yes its more linear than the original

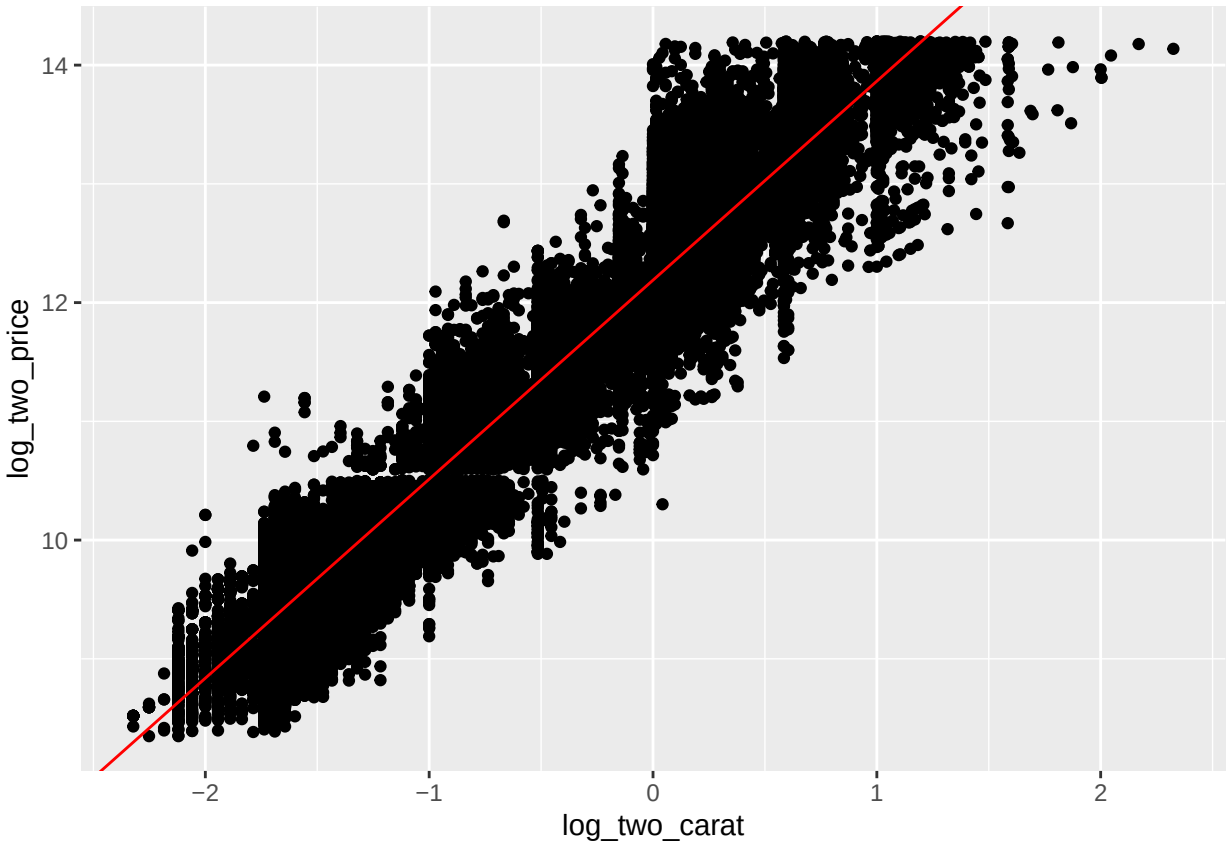
c. **Make a linear model to predict log(price) using log(carat). What are the coefficients of this model?**

```
log_di <- diamonds %>%
  mutate(log_two_price = log2(price), log_two_carat = log2(carat))

  lm(log_two_price ~ log_two_carat, data = log_di)
```

```
##
## Call:
## lm(formula = log_two_price ~ log_two_carat, data = log_di)
##
## Coefficients:
##   (Intercept)  log_two_carat
##        12.189          1.676
```

d. **Draw your linear model from the previous part on top of your scatterplot from part (b).**

```
diamonds %>% mutate(log_two_price = log2(price), log_two_carat = log2(carat)) %>% ggplot(aes(x = log_two
```


14
12
10
log_two_price
−2
−1
0
1
2
log_two_carat