Homework_5

Emmenta Janneh

2024-02-17

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     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
```

```
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.3.2
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.2
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.3.2
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.3.2
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(latex2exp)
```

```
## Warning: package 'latex2exp' was built under R version 4.3.2
```

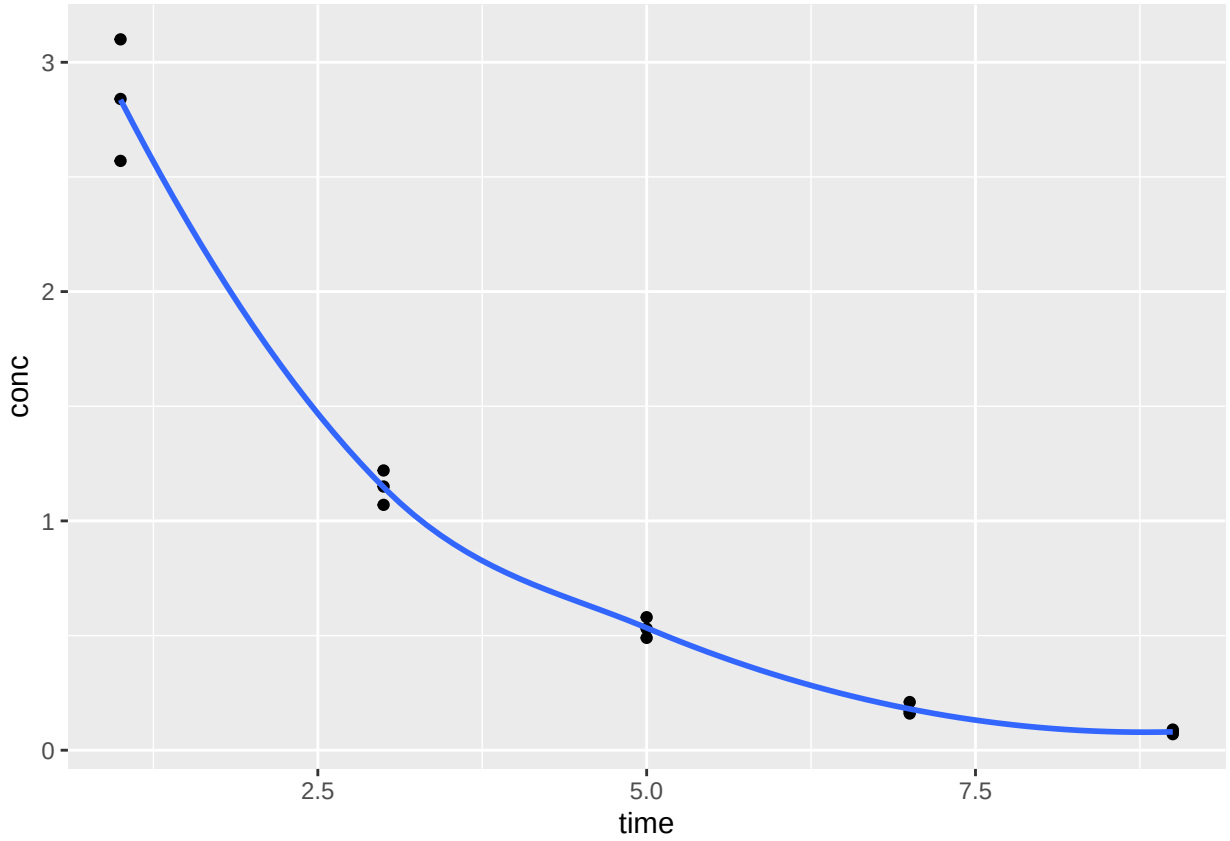
```
sol <- tribble(~conc, ~time,
               0.07, 9,
               0.09, 9,
               0.08, 9,
               0.16, 7,
               0.17, 7,
               0.21, 7,
               0.49, 5,
               0.58, 5,
               0.53, 5,
               1.22, 3,
               1.15, 3,
               1.07, 3,
               2.84, 1,
               2.57, 1,
               3.10, 1
)
```

Solution Concentration

1

```
ggplot(sol, aes(x = time, y = conc)) +
  geom_point() +
  geom_smooth(se = FALSE)
```

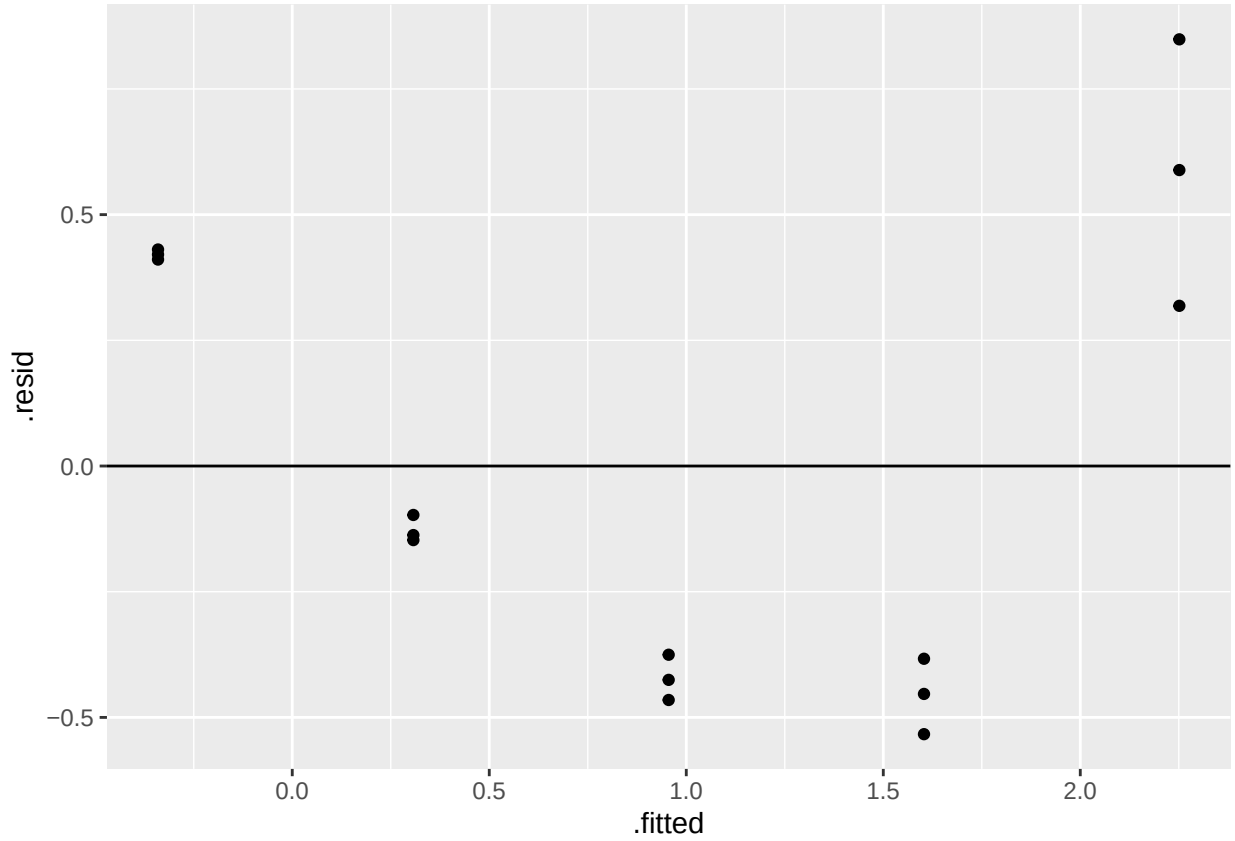
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



The relationship between concentration and time is curved, and the association is negative. There are no outliers in the data points, but we have a violation of constant variance.

Lets explore the residuals in a linear regression.

```
lm_sol <- lm(conc~time, data = sol)
a_sol <- augment(lm_sol)
ggplot(a_sol, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0)
```



From the graphs above, we observe the following:

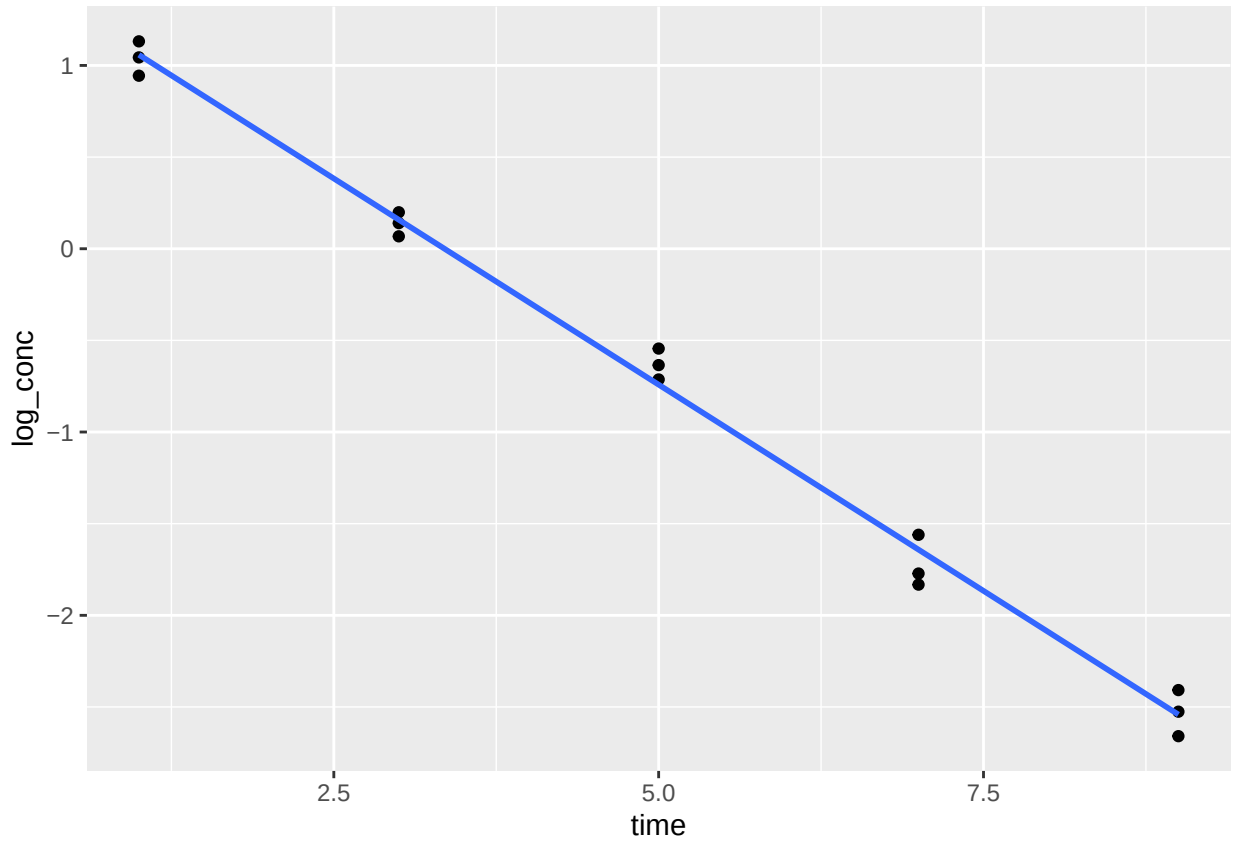
- Curved relation between time and concentration
- Monotone
- Variance look like it increase as y increases

2

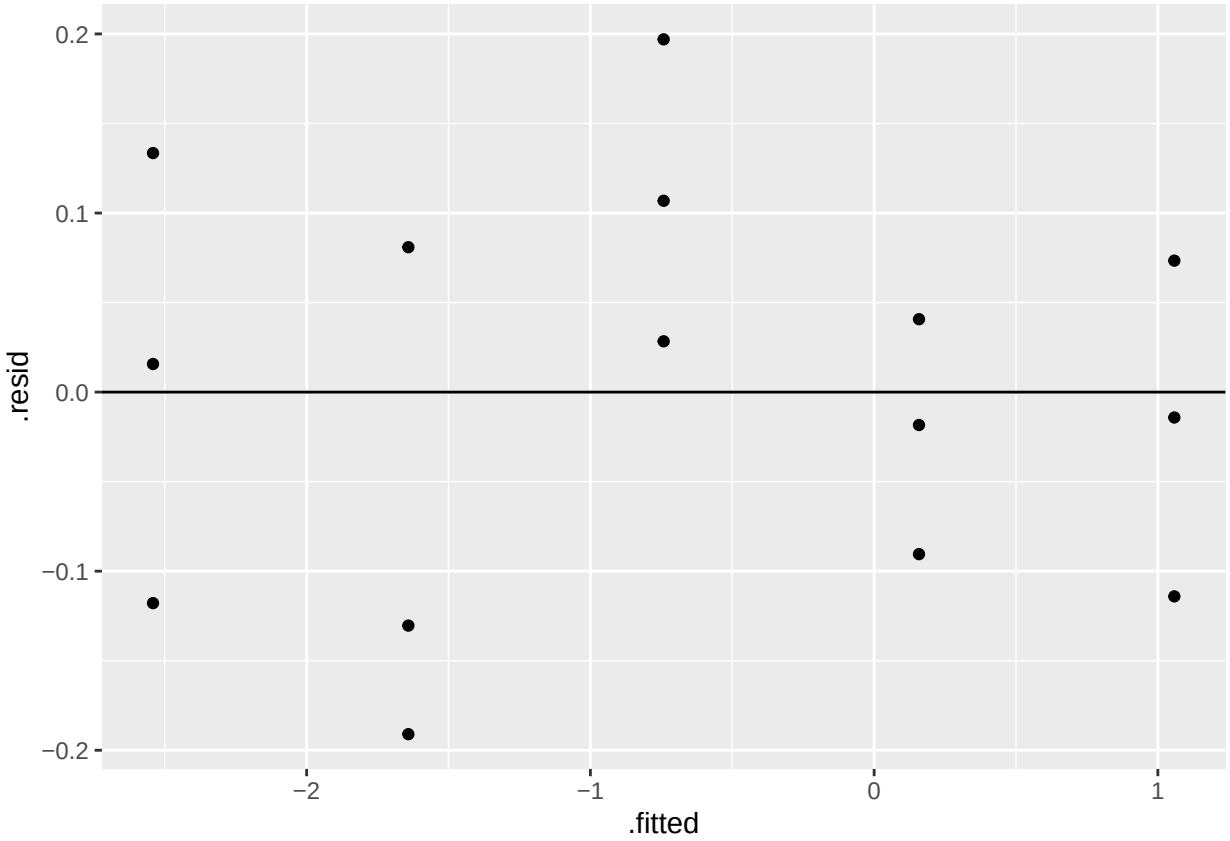
To resolve the violations in our analysis, lets transform the concentration to log.

```
sol <- mutate(sol, log_conc = log(conc))
ggplot(sol, aes(x = time, y = log_conc)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
new_lmsol <- lm(log_conc ~ time, sol)
new_asol <- augment(new_lmsol)
ggplot(new_asol, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0)
```




```
tidy(new_lmsol, conf.int = TRUE)
```

```
## # A tibble: 2 x 7
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)    1.51     0.0603      25.0 2.22e-12    1.38      1.64
## 2 time          -0.450    0.0105     -42.9 2.19e-15   -0.473    -0.427
```

With the transformation, we came up with a model `log(y) = 1.51 - 0.45x + noise`. With this, we would conclude that we have a very strong evidence of a linear association between concentration and time ($\boldsymbol{p < 0.001, n = 15}$). Solutions with 1 more hour are estimated to have 36.2% worse concentrations, on average (with confidence interval of 34.8% lower and 37.7% lower).

Real Estate Data

We are using the estate data from `https://dcgerard.github.io/stat_415_615/data/estate.csv` for this analysis.

```
estate <- read_csv("https://dcgerard.github.io/stat_415_615/data/estate.csv")
```

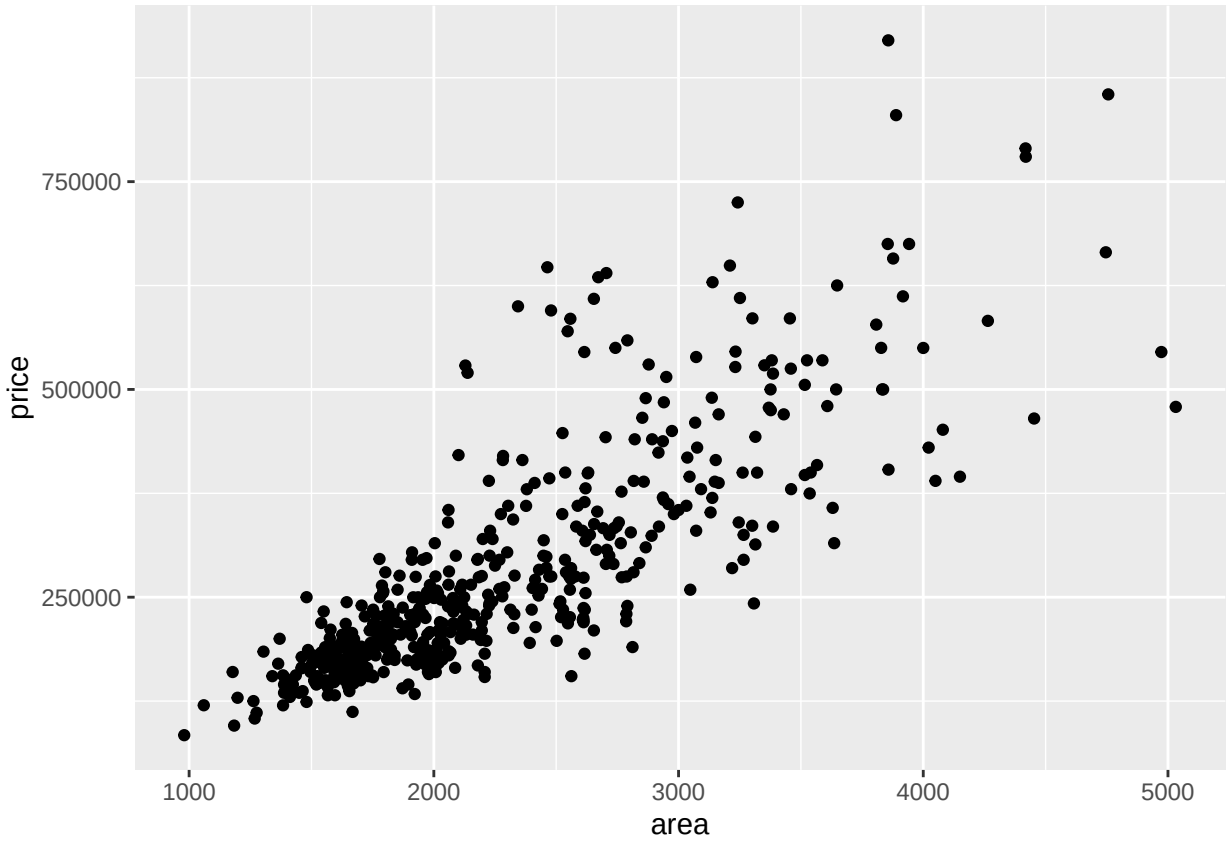
```
## Rows: 522 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## chr (4): ac, pool, quality, highway
## dbl (9): id, price, area, bed, bath, garage, year, style, lot
##
## 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.
```

1

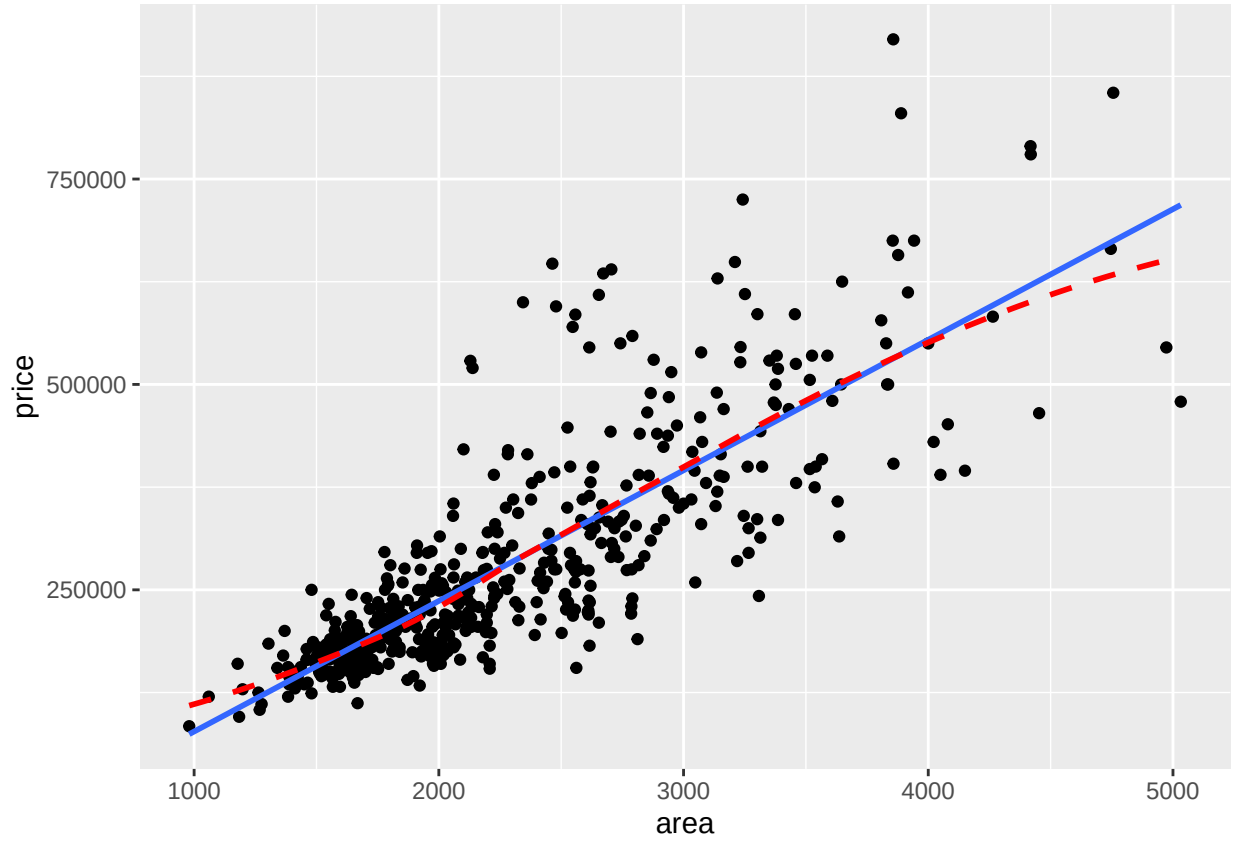
The two variables are - `price`: sales price for houses in US Dollars. - `area`: size of houses finished in square feet.

```
ggplot(estate, aes(x = area, y = price)) +
  geom_point()
```



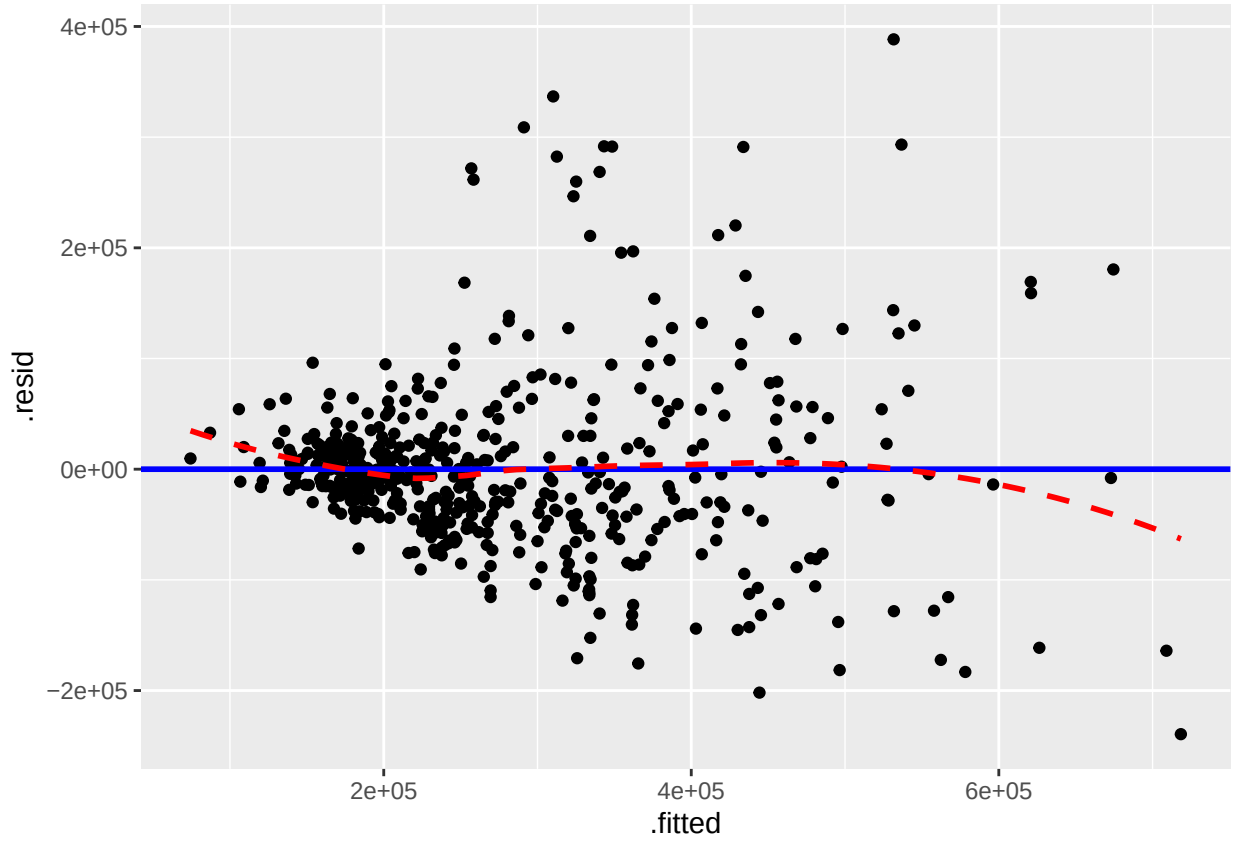

```
ggplot(estate, aes(x = area, y = price)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE) +
  geom_smooth(se = FALSE, color = "red", linetype = 2)
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
## ‘geom_smooth()‘ using method = ’loess’ and formula = ’y ~ x’
```



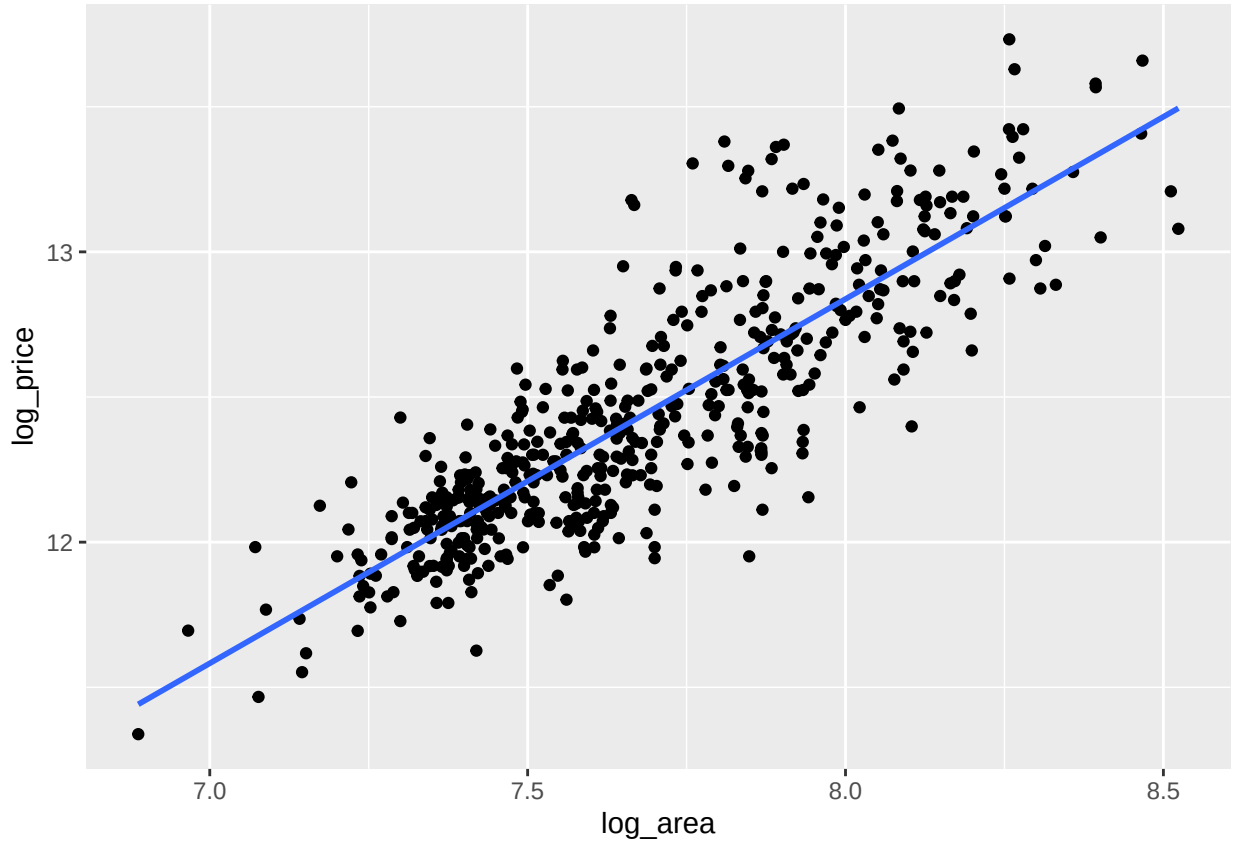

```
lm_estate <- lm(price ~ area, estate)
a_estate <- augment(lm_estate)
ggplot(a_estate, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "blue", linewidth = 1) +
  geom_smooth(se = FALSE, color = "red", linetype = 2)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
new_estate <- mutate(estate, log_price = log(price), log_area = log(area))
ggplot(new_estate, aes(x = log_area, y = log_price)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE)
```

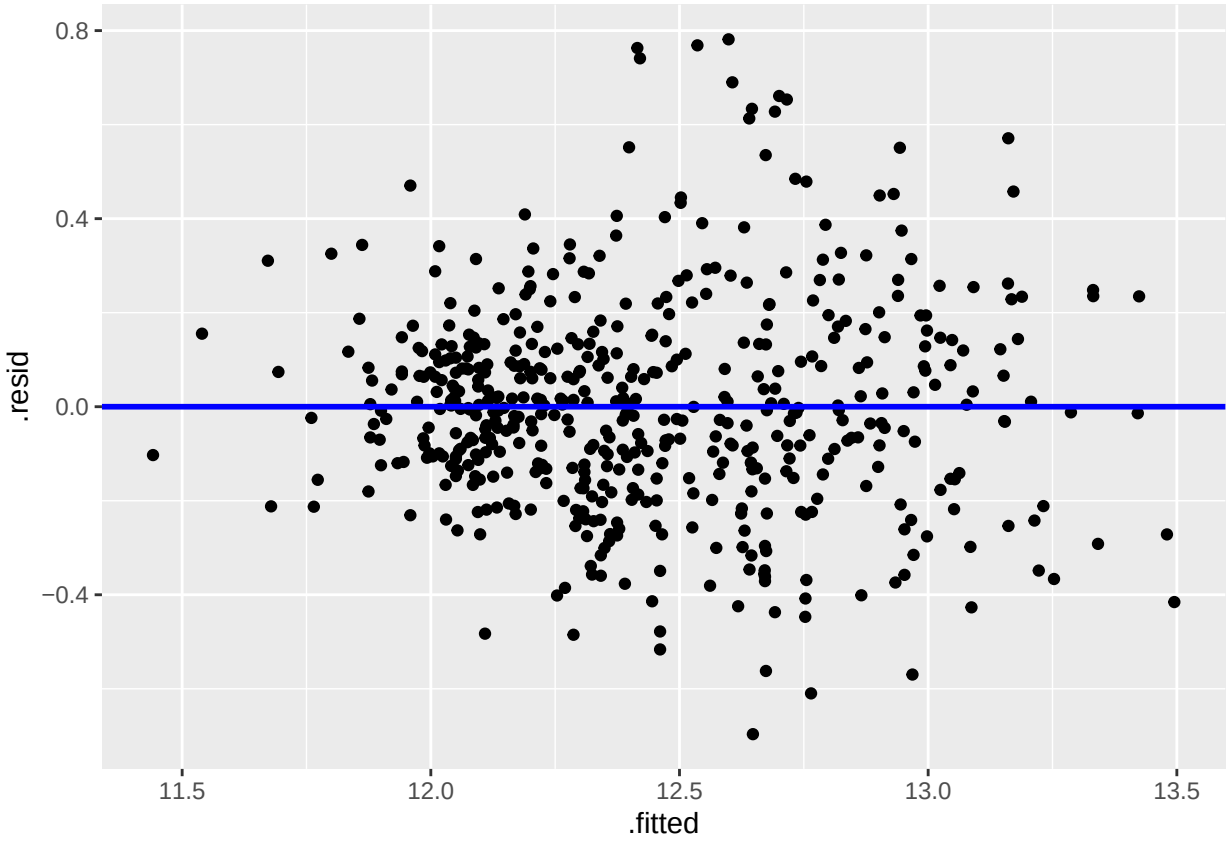
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
new_lmestate <- lm(log_price ~ log_area, new_estate)
new_lmestate
```

```
##
## Call:
## lm(formula = log_price ~ log_area, data = new_estate)
##
## Coefficients:
## (Intercept)     log_area
##       2.796        1.255
```

```
a_newestate <- augment(new_lmestate)
ggplot(a_newestate, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "blue", linewidth = 1)
```




```
tidy(new_lmestate, conf.int = TRUE)
```

```
## # A tibble: 2 x 7
##   term        estimate std.error statistic   p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)     2.80    0.259       10.8 1.31e- 24     2.29      3.31
## 2 log_area        1.26    0.0337      37.2 1.20e-148     1.19      1.32
```

```
newdf <- data.frame(log_area = c(576, 1020, 3067))
newdf <- mutate(newdf, log_area = log(log_area))
predict(object = new_lmestate, newdata = newdf, interval = "prediction") |>
  cbind(newdf) -> newdf
newdf <- mutate(newdf, fit = exp(fit), lwr = exp(lwr), upr = exp(upr), log_area = exp(log_area))
newdf
```

```
##         fit       lwr       upr log_area
## 1  47780.24  30385.70  75132.42      576
## 2  97894.03  62616.93 153045.54     1020
## 3 389829.98 249894.86 608125.42     3067
```

Report

For the analysis, we are using the estat data from dcgerard.github.io (https://dcgerard.github.io/stat_415_615/data/estate.csv). The variables of concern are

- `price`: Sales price of houses in US Dollars
- `area`: Size of houses finished in square feet

The first thing we did is to plot the data and we found some violations to the linear model. Characteristics of the data points constructed include:

- `Linearity`: There is definitely a positive association between price and area, and the data points seems linear
- `Uncorrelated error`: We assume the data was randomly sampled
- `Nonequal variance`: We can clearly see that the standard errors of the data points are off. The variance tends to be larger as we move up to higher area sizes.

To resolve this, we transform our x and/or y variables. We started off by transforming the y(price) to log(y). Than still did not resolve our variance, so we advanced into transforming x(area) as well, to log(x), and that seem to work better. We fit using the log transformation of x and y, and that seems to also work well.

We did a **T**-test to see if the linear model is insufficient:

- `H0`:B1 = 0 - No linear association
- `HA`:B1 != 0 - Linear association

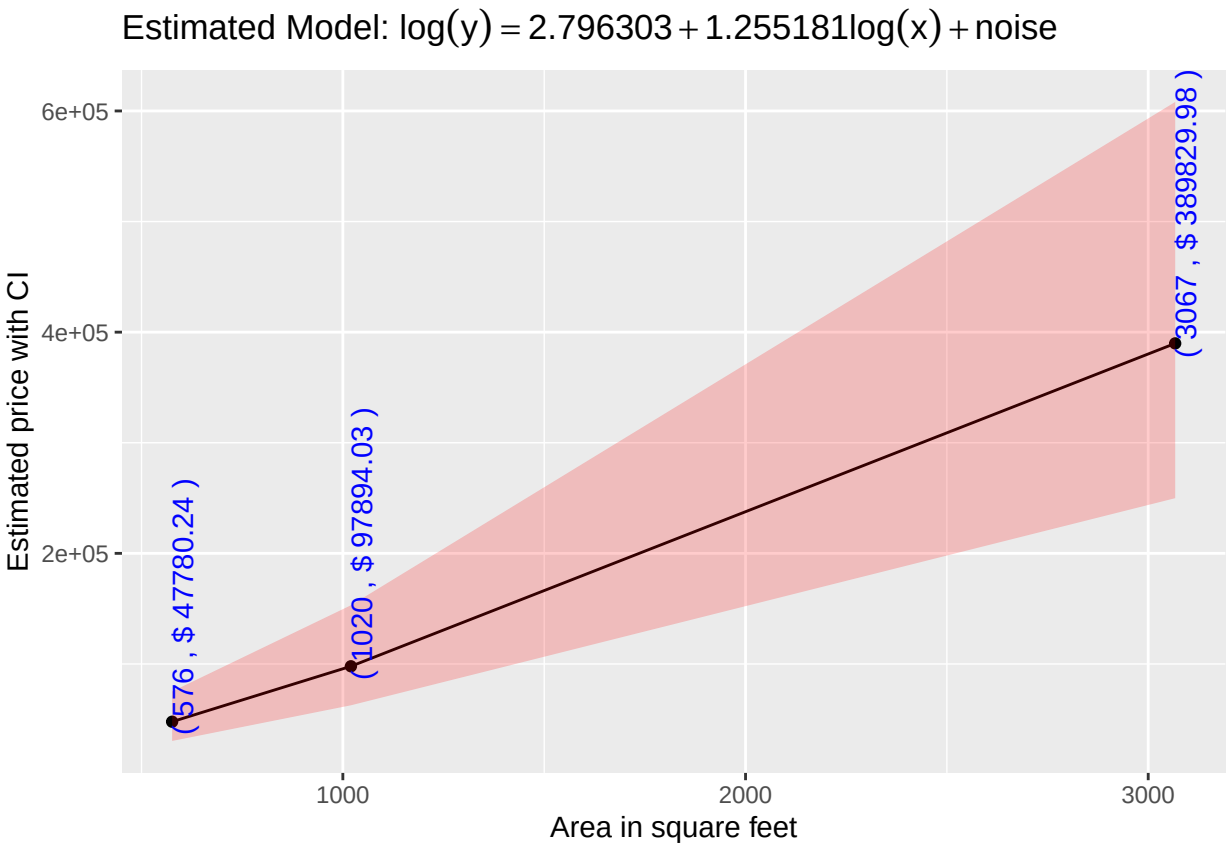
This shows that we have a very strong evidence against the Null hypothesis ($p < 0.001$). This is a very strong evidence against the NULL Hypothesis (`p-value = 1.196211e-148, n = 522`). We estimate that houses to have on average **2.4** times higher price for area that are twice as large(**95%** confidence interval of **2.3** lower and **2.5** higher).

Since our goal was to predict the prices of houses with areas of 576, 1020, 3067 square feet, we transform the data back to its original scale.

- Houses with area of 576 are estimated to be at $47,780.24 with confidence range of $30,385.70 to $75,132.42
- Houses with area of 1020 are estimated to be at $97,894.03 with confidence range $62,616.93 to $153,045.54
- Houses with area of 3067 are estimated to be at $389,829.98 with confidence range $249,894.86 to $608,125.42

This is represented in the graph below

```
ggplot(data = newdf, aes(x = log_area, y = fit)) +
  geom_point() +
  geom_line(data = newdf, aes(x = log_area, y = fit)) +
  geom_ribbon(data = newdf, mapping = aes(x = log_area, ymin = lwr, ymax = upr), alpha = 0.2, fill = "r
  geom_text(aes(label = paste("(", log_area, ", $", round(fit, 2), ")")), nudge_x = 0.1, nudge_y = 0.1,
  ggtitle(TeX("Estimated Model: $log(y) = 2.796303 + 1.255181log(x) + noise$")) +
  xlab("Area in square feet") +
  ylab("Estimated price with CI")
```


Estimated Model: log(y) = 2.796303 + 1.255181log(x) + noise
Estimated price with CI
6e+05
4e+05
2e+05
1000
2000
3000
Area in square feet
(576 , $ 47780.24)
(1020 , $ 97894.03)
(3067 , $ 389829.98)