Simple Linear Regression: Conditions and Transformations

Sleuth3 Chapter 8

Example: Exercise 8.17 in Sleuth3

Quote from book:

> In a study of the effectiveness of biological control fo the exotic weed tansy ragwort, researchers manipulated the exposure to the ragwort flea beetle on 15 plots that had been planted with a high density of ragword. Havesting the plots the next season, they measured the average dry mass of ragword remaining (grams/plant) and the flea beetle load (beetles/gram of ragwort dry mass) to see if the ragwort plants in plots with high flea beetle loads were smaller as a result of herbivory by the beetles. (Data from P. McEvoy and C. Cox, "Successful Biological Control of Ragwort, *Senecio jacobaea, by Introduced Insects in Oregon,", *Ecological Applications* 1 (4) (1991): 430-42.)

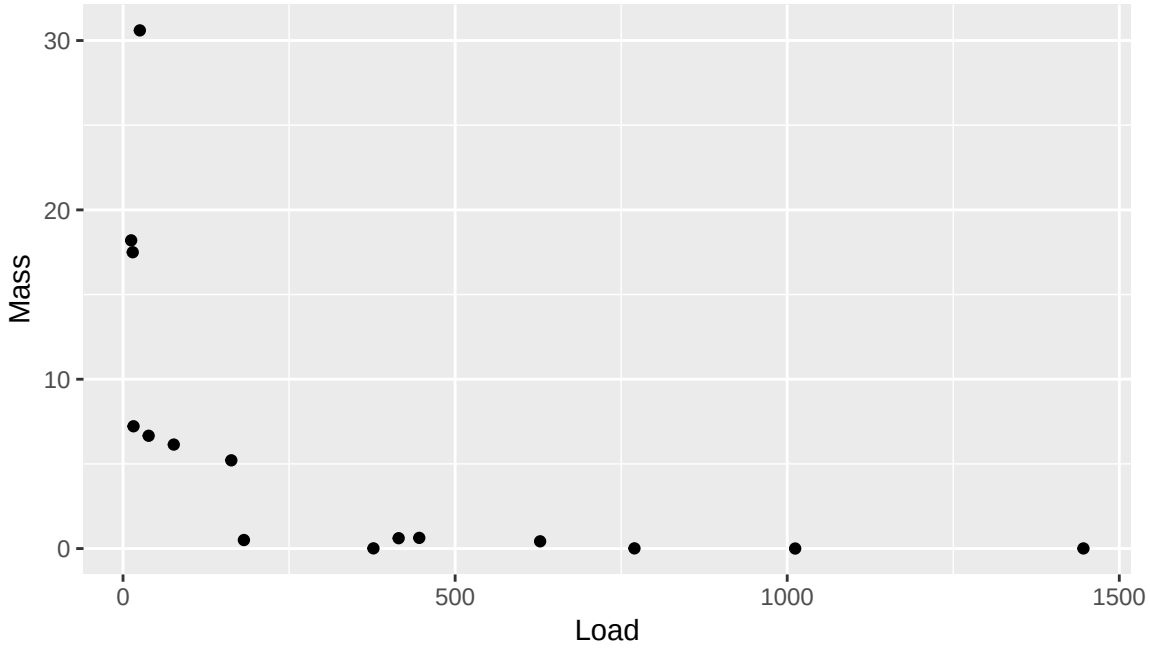
Here is the data:

```
## # A tibble: 6 x 2
##    Load  Mass
##   <dbl> <dbl>
## 1  12.2 18.2
## 2  14.6 17.5
## 3  15.8  7.22
## 4  25.3 30.6
## 5  38.6  6.66
## 6  76.4  6.14
```

Our explanatory variable is `Load`, and the response is `Mass`.

1. Make a suitable plot of the data.

```
ggplot(data = pest_control, mapping = aes(x = Load, y = Mass)) +
  geom_point()
```

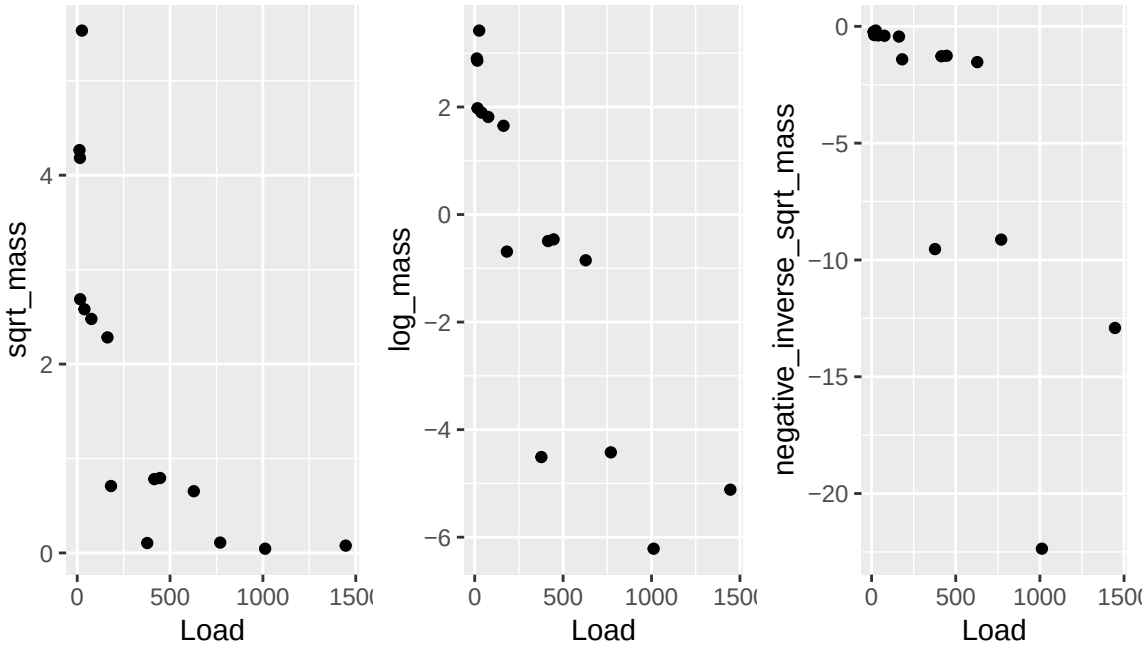


2. Through trial and error, find a suitable transformation of the data so that the linear regression conditions are satisfied as well as possible. (Let's assume the plots were in different areas so that they can be regarded as independent.)

The first thing I see is that the standard deviation of the response variable is not equal across the range of values for the explanatory variable. This suggests that we should start by considering transformations of the response variable. Since the response variable is skewed right (many small values of Mass, a few large outlying values), I will move down the ladder of powers.

I'm showing the next three steps down on the ladder here:

```
pest_control <- pest_control %>%
  mutate(
    sqrt_mass = sqrt(Mass),
    log_mass = log(Mass),
    negative_inverse_sqrt_mass = -1/sqrt(Mass)
  )

p_sqrt <- ggplot(data = pest_control, mapping = aes(x = Load, y = sqrt_mass)) +
  geom_point()

p_log <- ggplot(data = pest_control, mapping = aes(x = Load, y = log_mass)) +
  geom_point()

p_neg_inv_sqrt <- ggplot(data = pest_control, mapping = aes(x = Load, y = negative_inverse_sqrt_mass)) +
  geom_point()

grid.arrange(
  p_sqrt,
  p_log,
  p_neg_inv_sqrt,
  nrow = 1
)
```



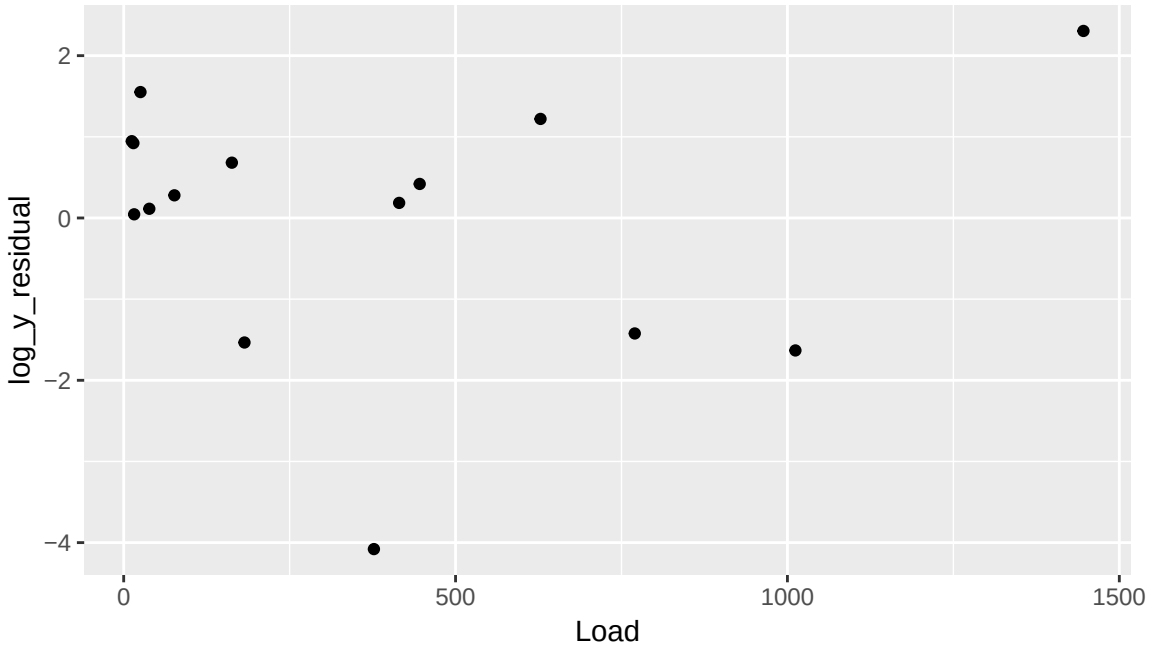
In these plots:

- The square root transformation doesn't seem to have done enough. There is still a larger standard deviation for small values of Load than for large values of Load
- The log transformation looks better, though maybe not perfect.
- The $-1/\sqrt{y}$ transformation went too far: the standard deviation is now too small on the left and too large on the right

Let's look at that log tranformation more closely:

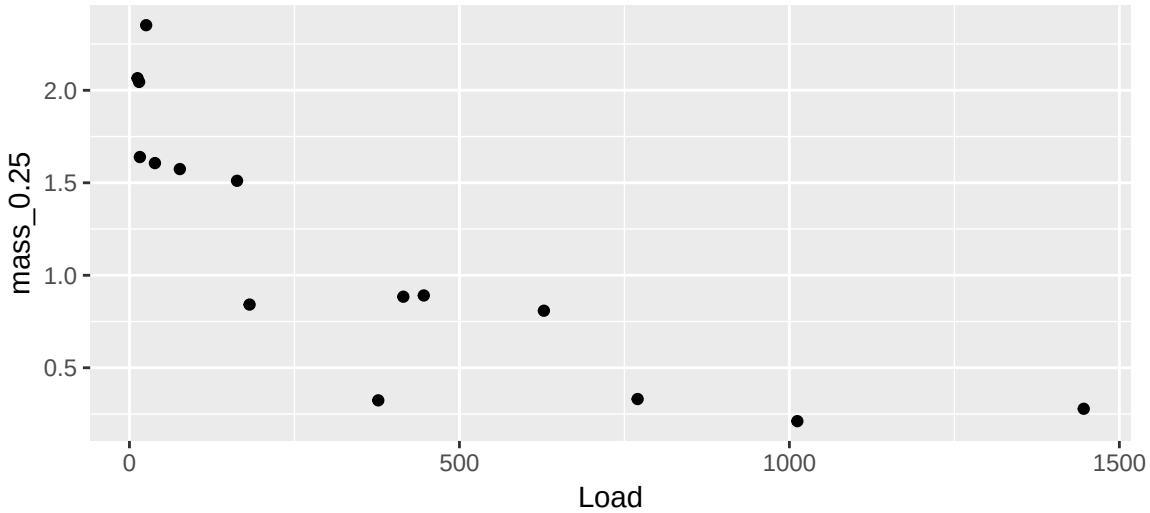
```
log_y_fit <- lm(log_mass ~ Load, data = pest_control)

pest_control <- pest_control %>%
```

```
  mutate(
    log_y_residual = residuals(log_y_fit)
  )

ggplot(data = pest_control, mapping = aes(x = Load, y = log_y_residual)) +
  geom_point()
```



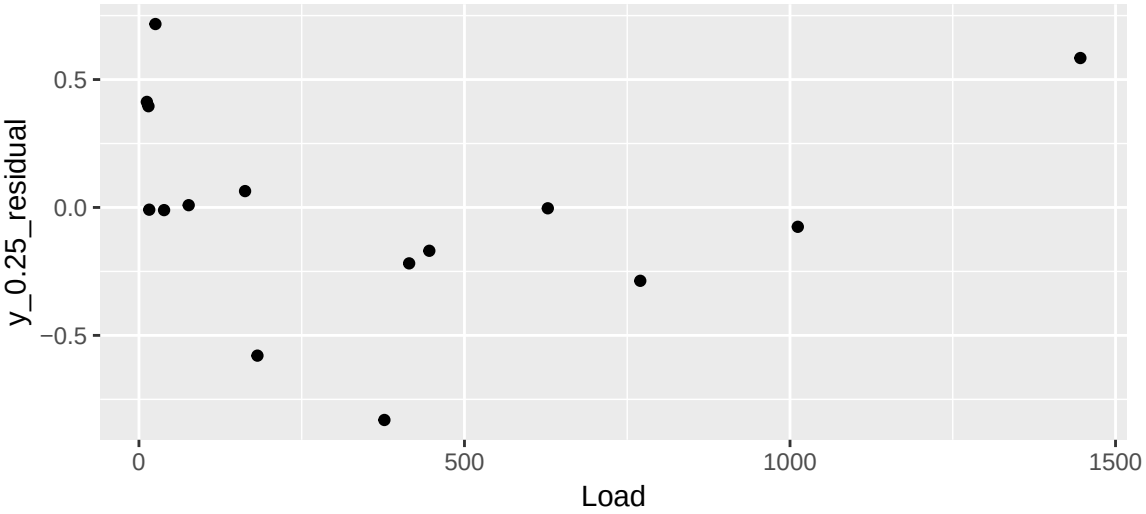
I see evidence of non-linearity here, as well as a smaller standard deviation for small Loads then for large Loads. Let's continue trying to fix the non-equal standard deviations. Since the square root transformation had too-large residuals on the left and the log transformation has too-small residuals on the left, let's try something inbetween those two: $y^{-0.25}$.

```
pest_control <- pest_control %>%
  mutate(
    mass_0.25 = Mass^0.25
  )

ggplot(data = pest_control, mapping = aes(x = Load, y = mass_0.25)) +
  geom_point()
```




```
y_0.25_fit <- lm(mass_0.25 ~ Load, data = pest_control)

pest_control <- pest_control %>%
  mutate(
```

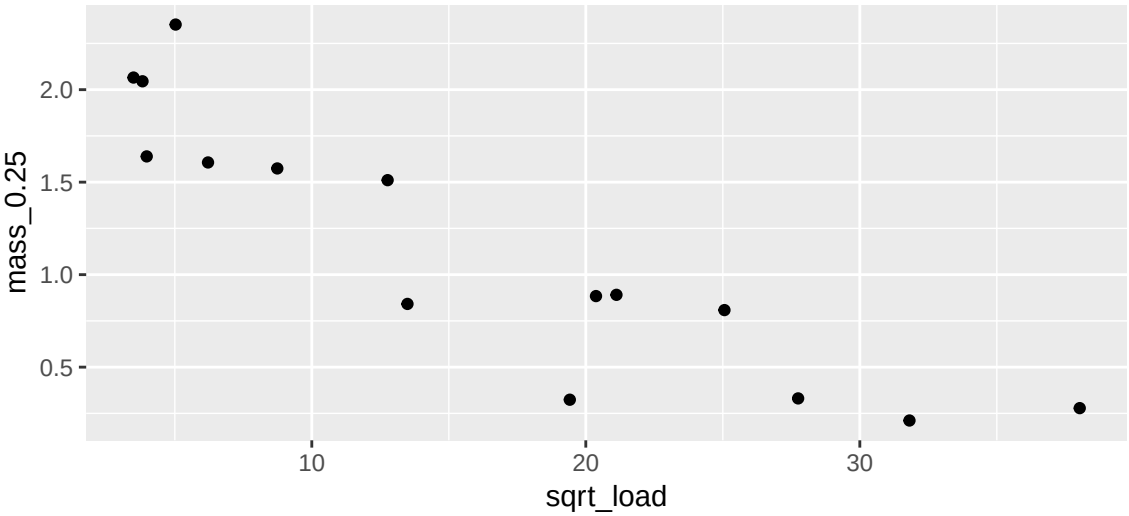
```
    y_0.25_residual = residuals(y_0.25_fit)
  )

ggplot(data = pest_control, mapping = aes(x = Load, y = y_0.25_residual)) +
  geom_point()
```



We are now in a place where the residual standard deviation is roughly equal across the range of values for load (at least, in the region where there is enough data to assess this).

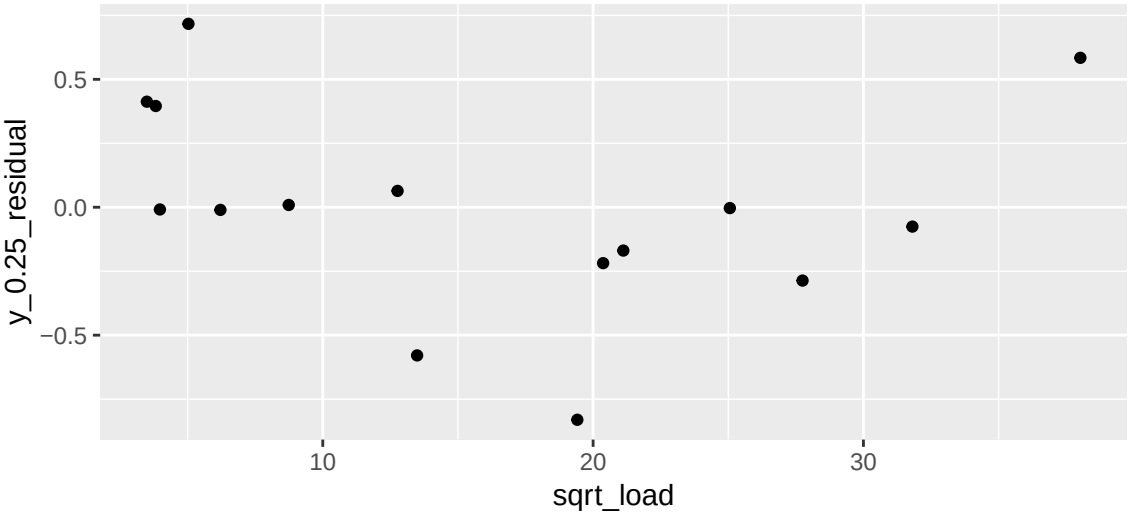
We still have a problem with non-linearity. We don't want to change the response variable any more though. We note that the Load variable is skewed right (many small values for Load, and a few large values). Let's try moving down the ladder of powers to pull in the outlying values.

```
pest_control <- pest_control %>%
  mutate(
    sqrt_load = Load^0.5,
    log_load = log(Load)
  )

ggplot(data = pest_control, mapping = aes(x = sqrt_load, y = mass_0.25)) +
  geom_point()
```

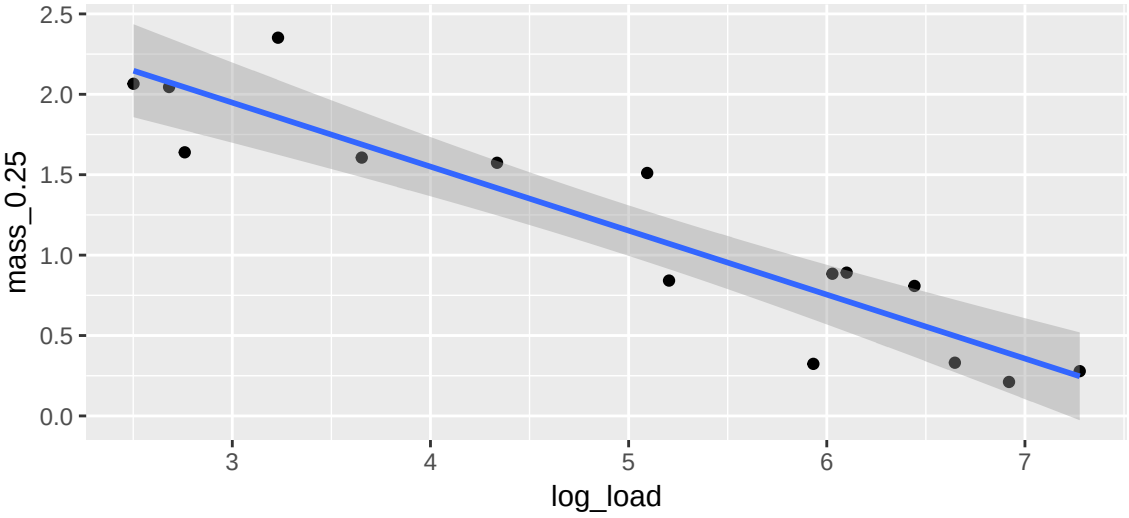



```
x_0.5_fit <- lm(mass_0.25 ~ sqrt_load, data = pest_control)

pest_control <- pest_control %>%
  mutate(
    x_0.5_residual = residuals(x_0.5_fit)
  )
```

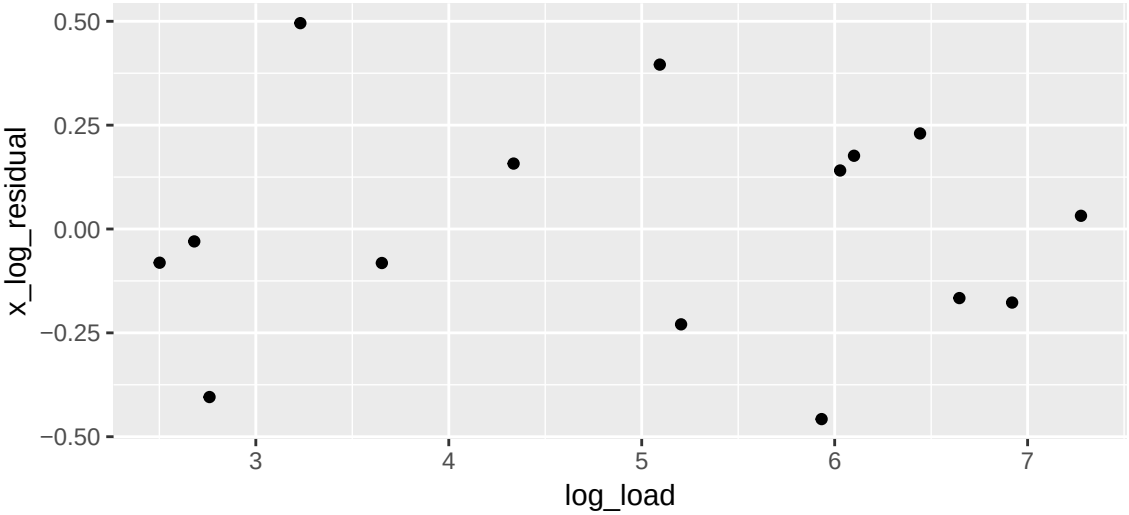
```
ggplot(data = pest_control, mapping = aes(x = sqrt_load, y = y_0.25_residual)) +
  geom_point()
```




```
ggplot(data = pest_control, mapping = aes(x = log_load, y = mass_0.25)) +
  geom_point() +
  geom_smooth(method = "lm")
```

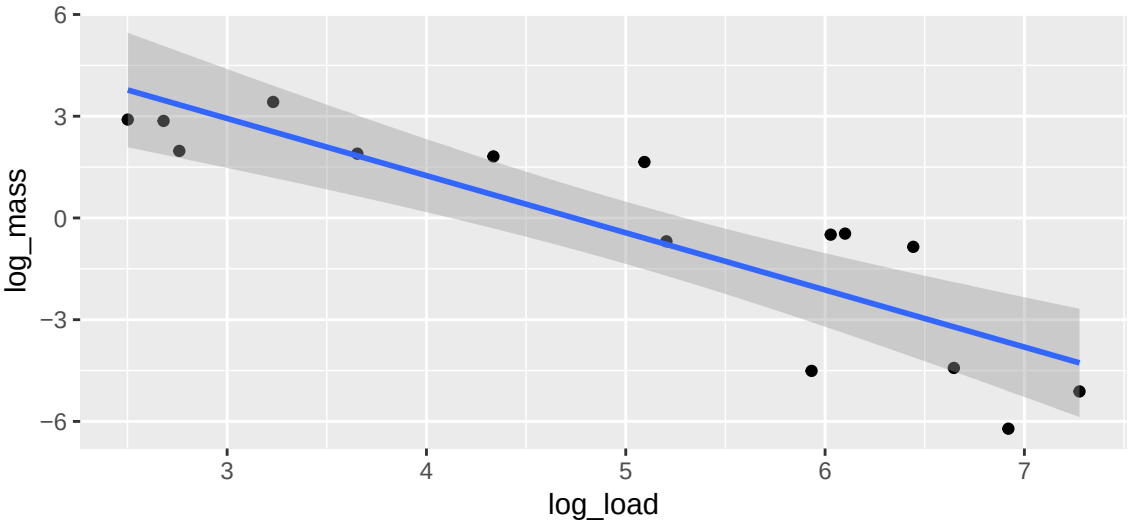



```
x_log_fit <- lm(mass_0.25 ~ log_load, data = pest_control)

pest_control <- pest_control %>%
  mutate(
    x_log_residual = residuals(x_log_fit)
  )

ggplot(data = pest_control, mapping = aes(x = log_load, y = x_log_residual)) +
  geom_point()
```



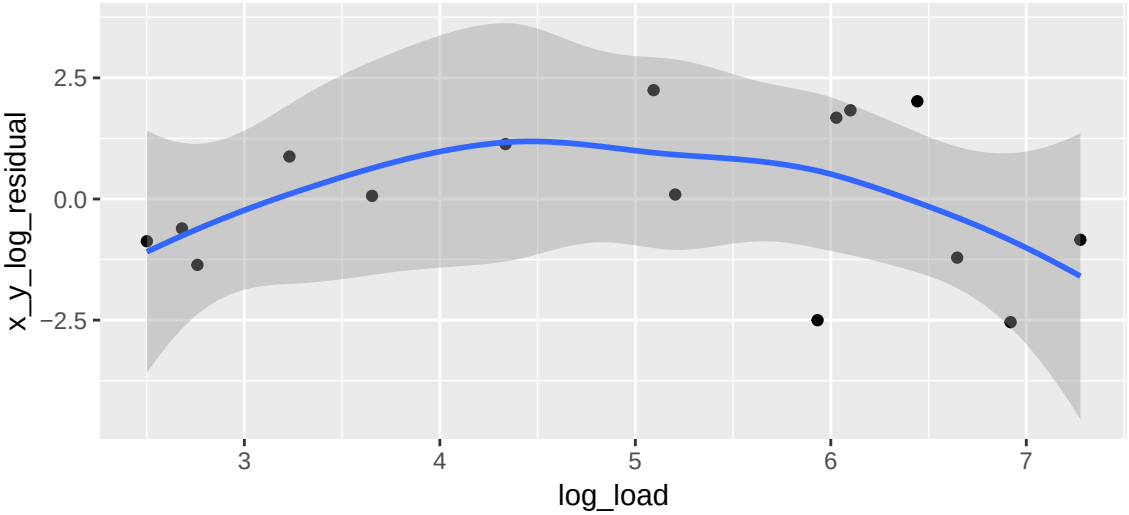
It would really be more convenient if we could use a log transformation for both… can we?

```
ggplot(data = pest_control, mapping = aes(x = log_load, y = log_mass)) +
  geom_point() +
  geom_smooth(method = "lm")
```




```
x_y_log_fit <- lm(log_mass ~ log_load, data = pest_control)

pest_control <- pest_control %>%
  mutate(
    x_y_log_residual = residuals(x_y_log_fit)
  )

ggplot(data = pest_control, mapping = aes(x = log_load, y = x_y_log_residual)) +
  geom_point() +
  geom_smooth()
```

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




```
pest_control <- pest_control %>%
  mutate(
    group = ifelse(log_load < 5, "small load", "large load")
  )

pest_control %>%
  group_by(group) %>%
  summarize(sd = sd(x_y_log_residual))
```

```
## # A tibble: 2 x 2
##   group         sd
##   <chr>      <dbl>
## 1 large load 1.94
## 2 small load 0.994
```

The log transformation is sortof ok, but just doesn't seem as good as the transformation to a power of 0.25.

3. Conduct a test of the claim that there is no association between the beetles load and the mean dry mass of ragweed harvested.

```
x_log_fit <- lm(mass_0.25 ~ log_load, data = pest_control)
summary(x_log_fit)
```

```
##
## Call:
## lm(formula = mass_0.25 ~ log_load, data = pest_control)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45759 -0.17168 -0.02985  0.16693  0.49557
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.14198    0.23669   13.28 6.17e-09 ***
## log_load    -0.39792    0.04517   -8.81 7.66e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2816 on 13 degrees of freedom
## Multiple R-squared:  0.8565, Adjusted R-squared:  0.8455
## F-statistic: 77.62 on 1 and 13 DF,  p-value: 7.659e-07
```

$H_0 : \beta_1 = 0$: There is no linear association between log-transformed beetles load and ragweed mass to the power of 1/4.

$H_A : \beta_1 \neq 0$ There is a linear association between log-transformed beetles load and ragweed mass to the power of 1/4.

The p-value for this test is 7.66e-07. The data provide extremely strong evidence against the null hypothesis of no linear association between log-transformed beetles load and ragweed mass to the power of 1/4.

4. (New question I added.) Find an estimate of the median ragweed mass when the beetle load is 50, and when it is 250. Additionally, report a confidence interval for the median ragweed mass when the beetle load is 250.

First: apply the transformation to the explanatory variable, save in a data frame

```
predict_data <- data.frame(
  Load = c(50, 250)
) %>%
  mutate(
    log_load = log(Load)
  )

predict_data
```

```
##   Load log_load
## 1   50 3.912023
## 2  250 5.521461
```

Second: generate a prediction/estimate for the mean of $Mass^{0.25}$

```
predict(x_log_fit, newdata = predict_data, level = 0.95, interval = "confidence")
```

```
##         fit       lwr      upr
## 1 1.5853238 1.3964518 1.774196
## 2 0.9449032 0.7794105 1.110396
```

Third: Undo the transformation.

Point estimates:

```
1.5853238^4
```

```
## [1] 6.316433
```

```
0.9449032^4
```

```
## [1] 0.7971669
```

Confidence interval for median when Load = 250:

```
0.7794105^4
```

```
## [1] 0.3690328
```

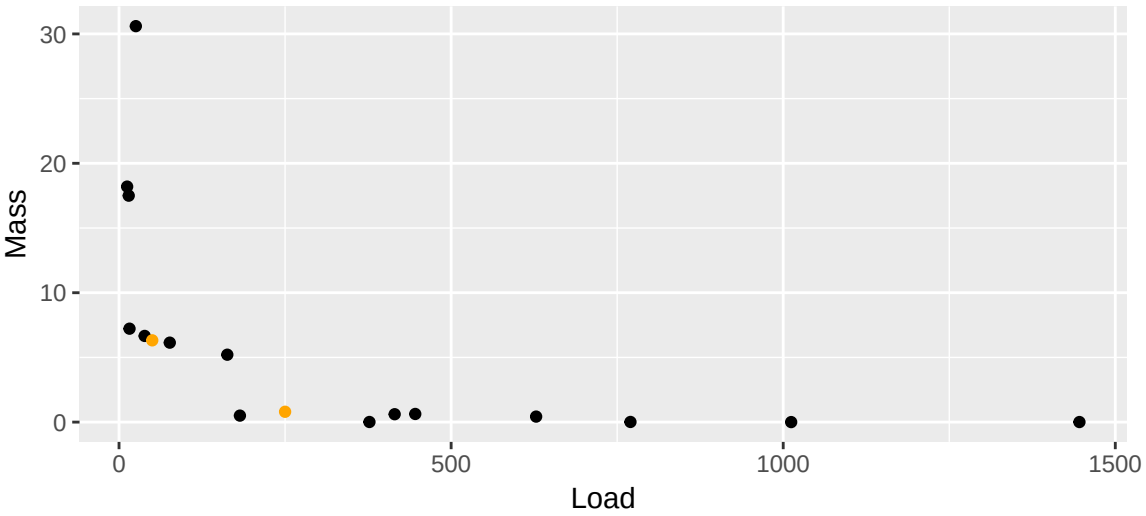
```
1.110396^4
```

```
## [1] 1.520238
```

We estimate that the median ragweed mass in fields where the beetles load is 250 beetles/gram of ragword dry mass is about 0.797 grams/plant. We are 95% confident that this median is between about 0.369 and 1.520.

Here is a plot:

```
predict_data <- predict_data %>%
  mutate(
    Mass = c(6.32, 0.80)
  )

ggplot(data = pest_control, mapping = aes(x = Load, y = Mass)) +
  geom_point() +
  geom_point(data = predict_data, color = "orange")
```


30
20
10
0
Mass
0
500
1000
1500
Load