Simple Linear Regression: Conditions and Transformations

Sleuth3 Chapter 8

Example 1: Adapted from Exercise 8.24 in Sleuth3

Quote from the book:

> A high respiratory rate is a potential diagnostic indicator of respiratory infection in children. To judge whether a respiratory rate is truly "high", however, a physician must have a clear picture of the distribution of normal respiratory rates. To this end, Italian researchers measured the respiratory rates of 618 children between the ages of 15 days and 3 years. Analyze the data and provide a statistical summary.

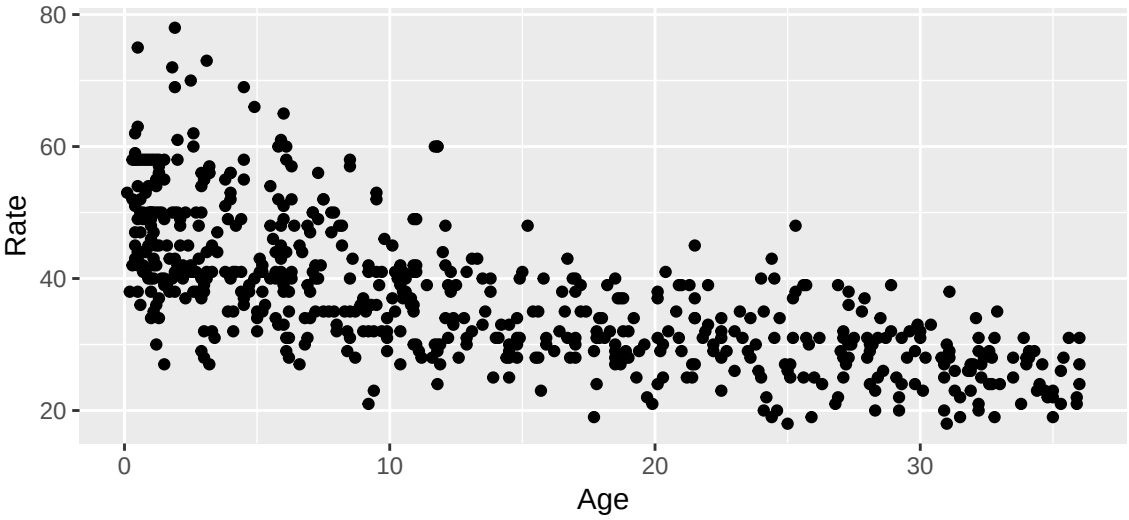
The following R code reads the data in.

```
## # A tibble: 6 x 2
##     Age  Rate
##   <dbl> <int>
## 1   0.1    53
## 2   0.2    38
## 3   0.3    58
## 4   0.3    52
## 5   0.3    42
## 6   0.4    62
```

Our explanatory variable is `Age` (in months), and the response is `Rate` (breaths per minute).

1. Make a suitable plot of the data.

```
ggplot(data = respiration, mapping = aes(x = Age, y = Rate)) +
  geom_point()
```

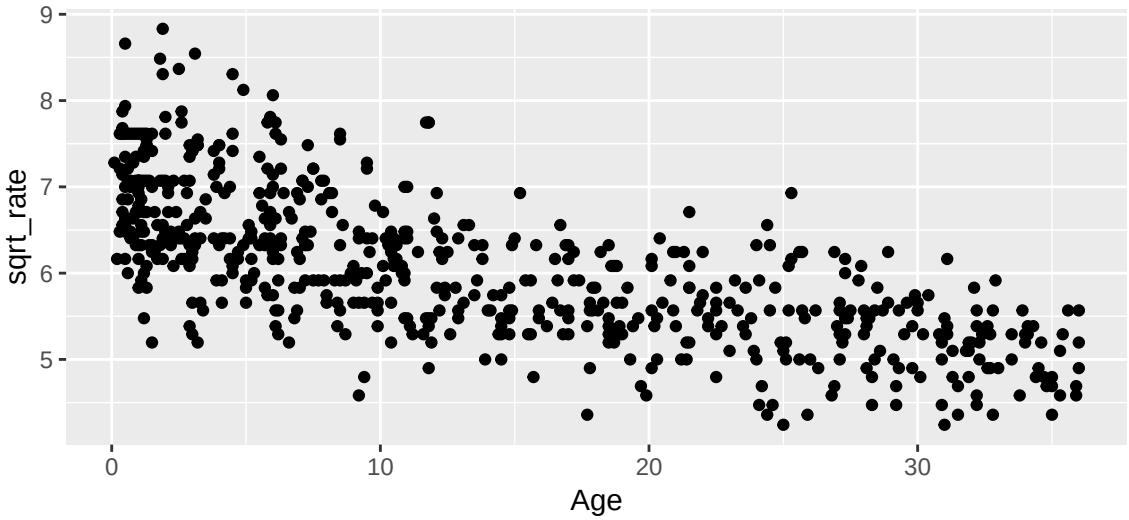


2. Through trial and error, find a suitable transformation of the data so that the linear regression conditions are satisfied as well as possible. (Let's assume the measurements for different children in the sample can be regarded as independent.) You can probably get away with only a transformation of the response variable.

In the initial plot, we see a larger standard deviation for the response for small ages, and a smaller standard deviation for the response for larger ages. This suggests that we try a transformation of the response variable. That variable is skewed to the right, so we should move down the ladder. I will start with a square root transformation:

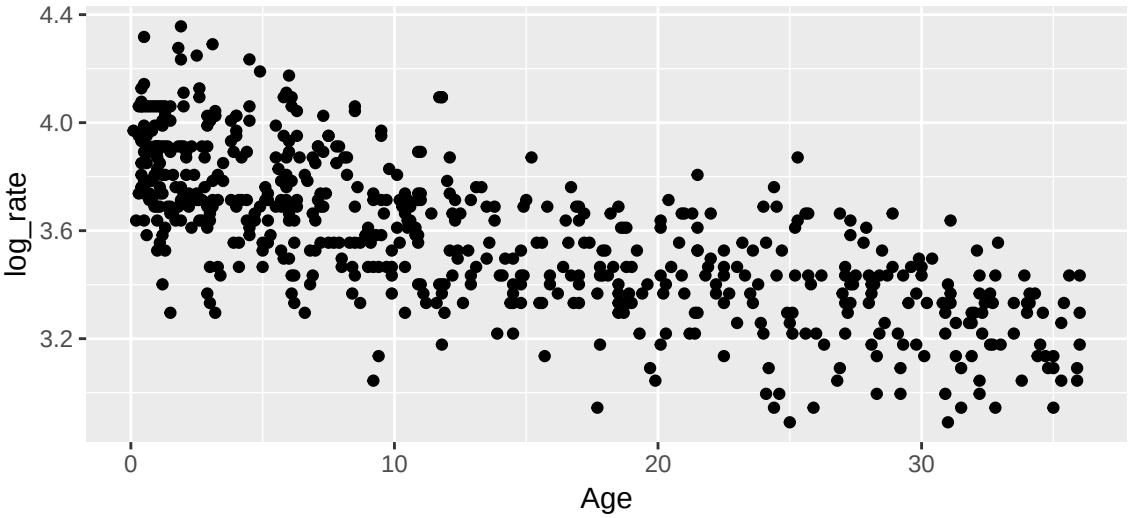
```
respiration <- respiration %>%
  mutate(
    sqrt_rate = sqrt(Rate)
```

```
  )

ggplot(data = respiration, mapping = aes(x = Age, y = sqrt_rate)) +
  geom_point()
```



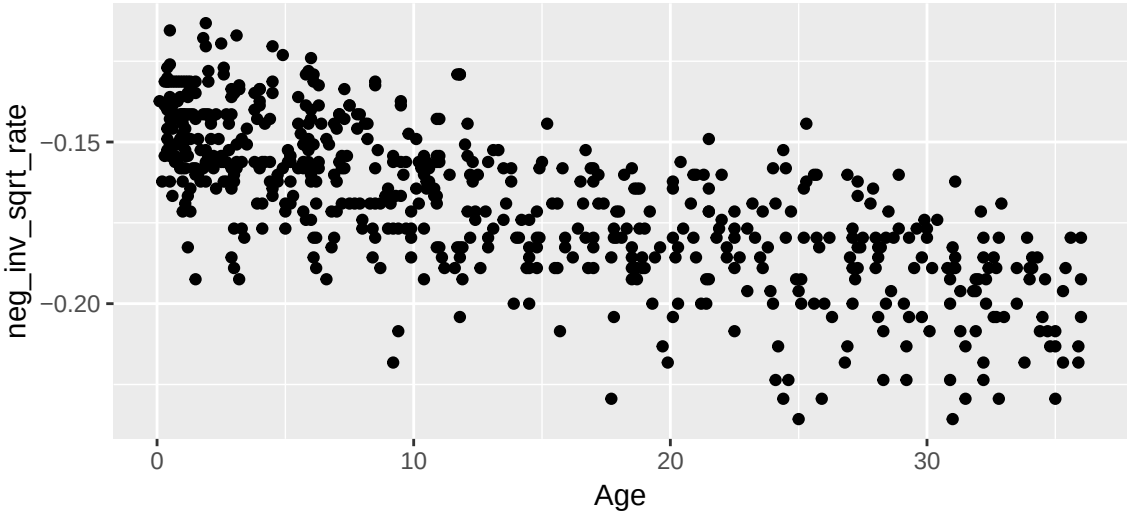
This is better, but not good enough. Let's try a log transformation:

```
respiration <- respiration %>%
  mutate(
    log_rate = log(Rate)
  )

ggplot(data = respiration, mapping = aes(x = Age, y = log_rate)) +
  geom_point()
```



This is still not quite good enough. One more step down?

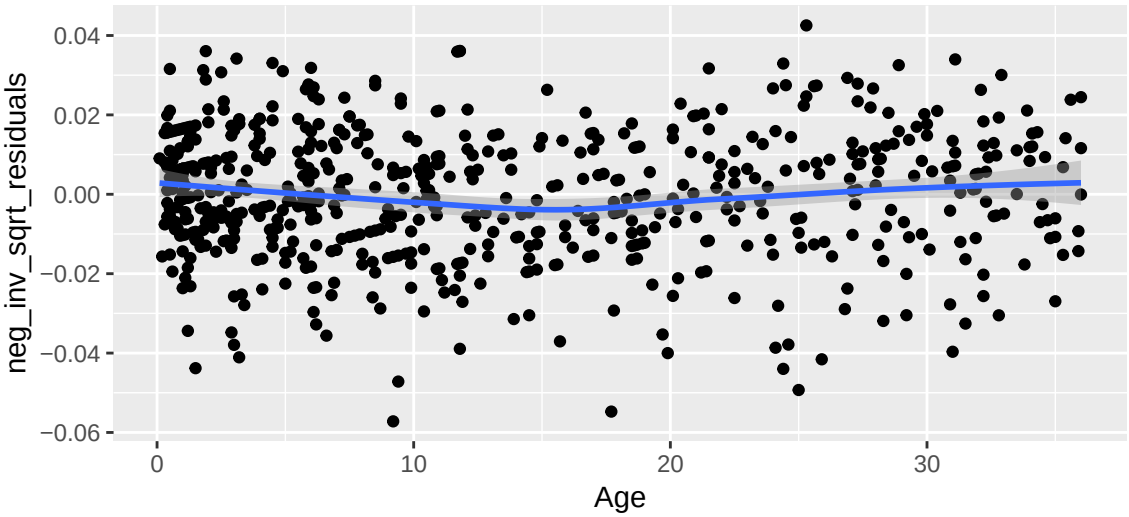
```
respiration <- respiration %>%
  mutate(
    neg_inv_sqrt_rate = -1/sqrt(Rate)
  )

ggplot(data = respiration, mapping = aes(x = Age, y = neg_inv_sqrt_rate)) +
  geom_point()
```



Let's fit a line to the data with this transformation and examine the residuals to see how we're doing.

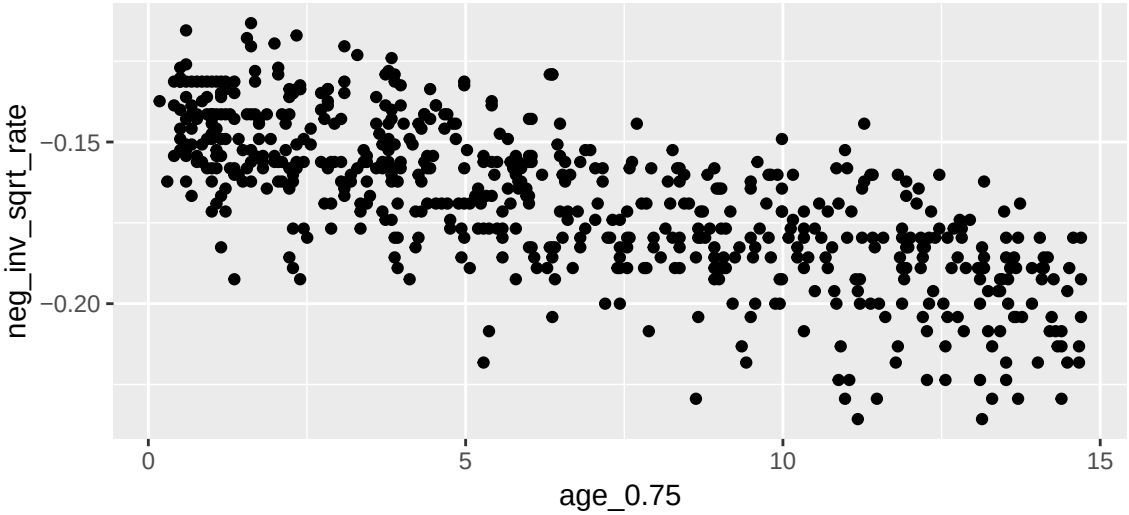
```
lm_fit <- lm(neg_inv_sqrt_rate ~ Age, data = respiration)
respiration <- respiration %>%
  mutate(
    neg_inv_sqrt_residuals = residuals(lm_fit)
  )

ggplot(data = respiration, mapping = aes(x = Age, y = neg_inv_sqrt_residuals)) +
  geom_point() +
  geom_smooth()
```

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



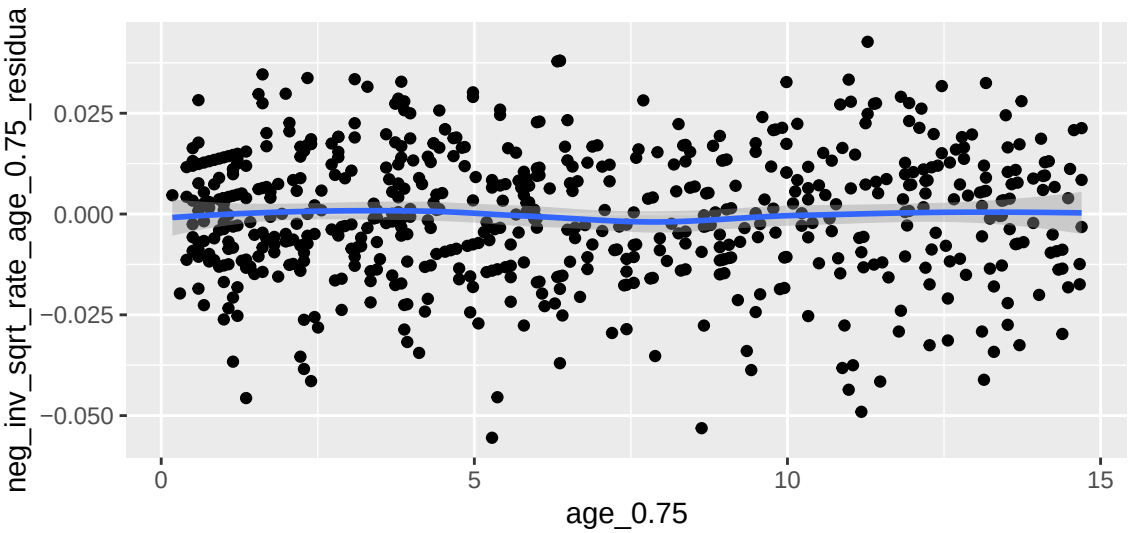
This residuals plot shows a consistent standard deviation of the residuals across all values of Age. There is a very slight indication of curvature. The relationship is close enough to linear at this point that the resulting model would probably be good enough for most purposes. If we really wanted to we could try adusting Age as well.

```
respiration <- respiration %>%
  mutate(
    age_0.75 = Age^0.75
  )

ggplot(data = respiration, mapping = aes(x = age_0.75, y = neg_inv_sqrt_rate)) +
  geom_point()
```




```
lm_fit <- lm(neg_inv_sqrt_rate ~ age_0.75, data = respiration)

respiration <- respiration %>%
  mutate(
    neg_inv_sqrt_rate_age_0.75_residuals = residuals(lm_fit)
  )

ggplot(data = respiration, mapping = aes(x = age_0.75, y = neg_inv_sqrt_rate_age_0.75_residuals)) +
  geom_point() +
  geom_smooth()
```

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



This model is ever so slightly better than the model with Age as the explanatory variable. Enough better that it's worth the effort of dealing with the transformation of the explanatory variable? Unclear.

3. Obtain 95% prediction intervals for the respiratory rates *on the original data scale* for children of ages 5 months, 10 months, 20 months, and 30 months. Visually compare your numbers with the plot from part 1; they should seem reasonable.

I'm going to work with the model where I had transformed age just to demonstrate the ideas. If we hadn't transformed the explanatory variable age, everything would be the same but we would not need to transform age in the first step.

The first thing I will do is set up a data frame with the ages at which we want to make predictions, and apply the selected transformation to those ages.

```
predict_data <- data.frame(
  Age = c(5, 10, 20, 30)
) %>%
  mutate(
    age_0.75 = Age^0.75
  )
```

Next, we will use our final model, with $Age^{0.75}$ as the explanatory variable and $-1/\sqrt{Rate}$ as the response, to generate predictions. These will be predictions for $-1/\sqrt{Rate}$

```
predict(lm_fit, newdata = predict_data, interval = "prediction")
```

```
##          fit        lwr        upr
## 1 -0.1548627 -0.1867942 -0.1229313
## 2 -0.1641093 -0.1960276 -0.1321909
## 3 -0.1796600 -0.2115893 -0.1477306
## 4 -0.1932930 -0.2252660 -0.1613200
```

We now need to undo the transformation of the response. If $Y = -1/\sqrt{Rate}$, then $\sqrt{Rate} = -1/Y$, or $Rate = 1/Y^2$. So to get back to rate, we need to take the predicted values above, square them, and then take the reciprocal of that. We can do this by converting the results above to a data frame and then using mutate on the columns of that data frame.

```
predictions <- predict(lm_fit, newdata = predict_data, interval = "prediction") %>%
  as.data.frame() %>%
  mutate(
    fit_original = 1/fit^2,
    lwr_original = 1/lwr^2,
    upr_original = 1/upr^2,
    Age = c(5, 10, 20, 30)
  )

predictions
```

```
##          fit        lwr        upr fit_original lwr_original upr_original
## 1 -0.1548627 -0.1867942 -0.1229313     41.69713     28.65982     66.17212
## 2 -0.1641093 -0.1960276 -0.1321909     37.13076     26.02349     57.22647
## 3 -0.1796600 -0.2115893 -0.1477306     30.98113     22.33636     45.82041
## 4 -0.1932930 -0.2252660 -0.1613200     26.76503     19.70646     38.42586
##   Age
## 1   5
## 2  10
## 3  20
## 4  30
```

We are 95% confident that a child of age 5 months will have a respiratory rate between 28.66 and 66.17 breaths per minute.

We are 95% confident that a child of age 10 months will have a respiratory rate between 26.02 and 57.23 breaths per minute.

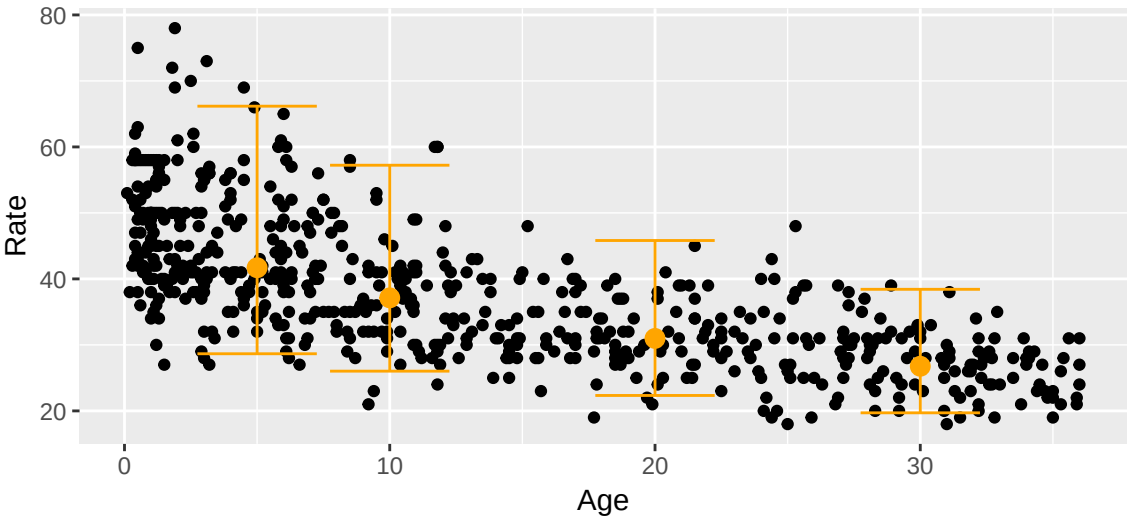
We are 95% confident that a child of age 20 months will have a respiratory rate between 22.34 and 45.82 breaths per minute.

We are 95% confident that a child of age 30 months will have a respiratory rate between 19.71 and 38.43 breaths per minute.

(These are four separate prediction intervals.)

Here is a plot of the original data with these intervals overlaid on top.

```
ggplot() +
  geom_point(data = respiration, mapping = aes(x = Age, y = Rate)) +
  geom_point(data = predictions, mapping = aes(x = Age, y = fit_original), color = "orange", size = 3) +
  geom_errorbar(data = predictions, mapping = aes(x = Age, ymin = lwr_original, ymax = upr_original), color =
```



Each of the prediction intervals does appear to contain most, but not all, of the observed values around the corresponding age. Note that the intervals at smaller ages are wider than the intervals at larger ages; this corresponds to the fact that the data have a higher standard deviation for small ages than for large ages. Also note that the intervals are asymmetric; they are longer on the top than on the bottom. This also matches a feature of the data, that the rates are skewed right for each age.