Homework06

Mazunki Hoksaas

2023-10-08

Homework (in book: 4.10.1)

In this exercise, you will perform the analysis corresponding to Figure 4.1 on page 70.

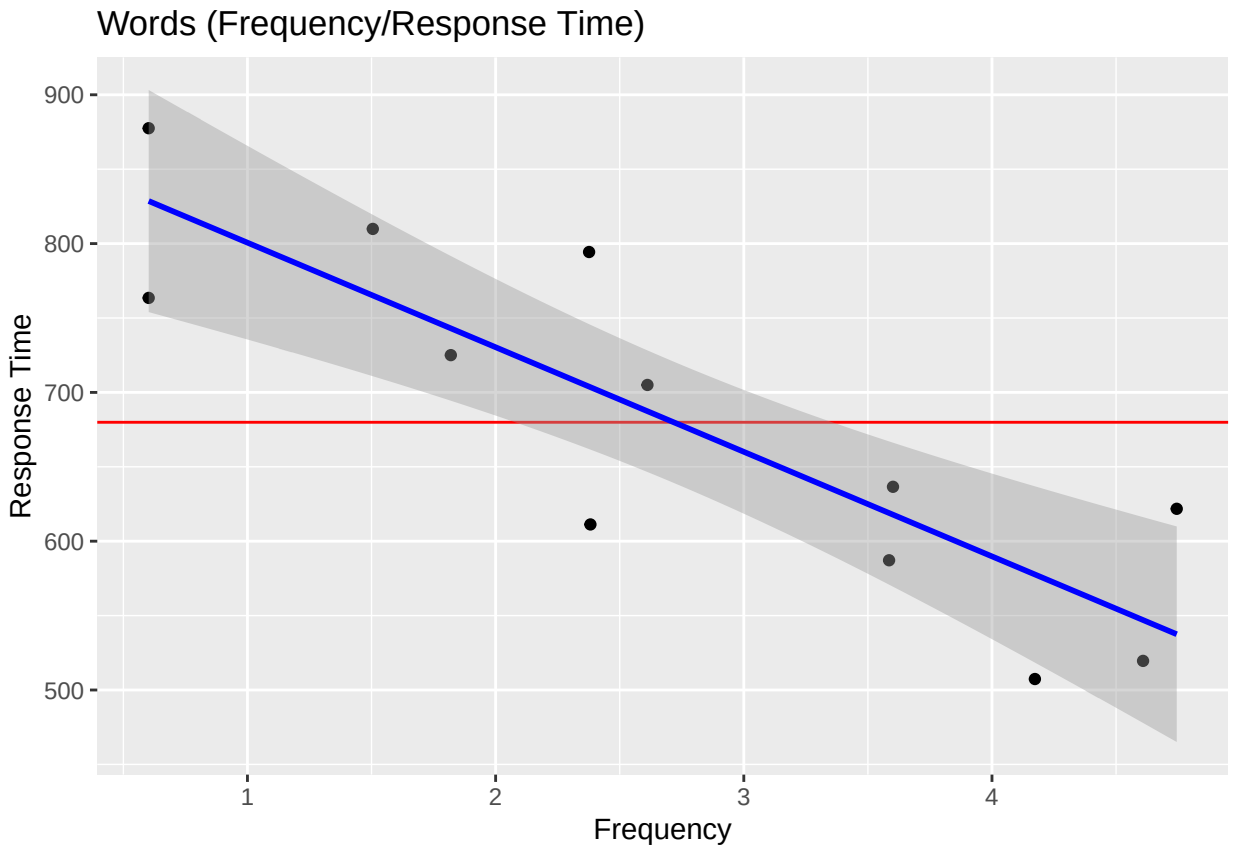
1. Load in the data set `ELP_frequency.csv` into your R session.
2. Use `mutate()` to apply the `log10()` function to the frequency column (`Freq`) (logarithms will be explained in Chapter 5).
3. Fit a linear model in which response times (`RT`, i.e. the y in our formula) are predicted as a function of log frequencies (i.e. the x in our formula).
4. Create a plot for the relationship between these two variables.
5. Can you add a horizontal line showing the mean response duration using `geom_hline()` and the `yintercept` aesthetic?
6. Can you add the regression line from your model?

```
dataset <-
  read_csv("ELP_frequency.csv", show_col_types=FALSE) %>%  # ex1
  mutate(log_frequency=log10(Freq))                        # ex2

fit <- lm(RT~log_frequency, data=dataset)                  # ex3

plot <- ggplot(data=dataset, aes(x=log_frequency, y=RT)) + # ex4
  geom_point() +  # scatter points
  geom_hline(yintercept=mean(dataset$RT), color="red") +   # ex5
  geom_smooth(method="lm", formula=y~x, color="blue") +    # ex6
  labs(x="Frequency", y="Response Time") +
  ggtitle("Words (Frequency/Response Time)")

print(plot)
```



regarding geom_smooth (ex6):

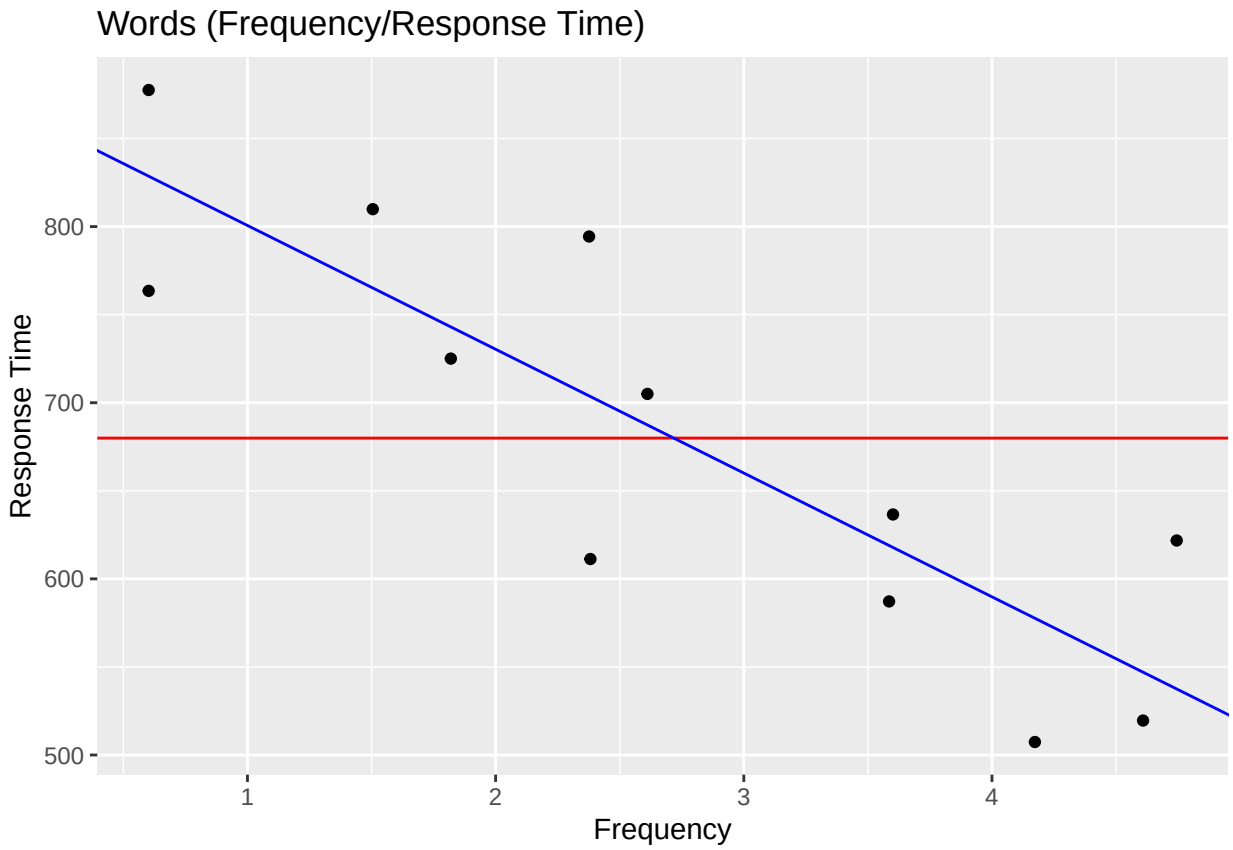
i wanted to reuse the model we had already trained on our data (aka `fit`), but i couldn't figure out how to do that. it feels unnecessary to re-calculate the model by telling it which method to use... is there a way by which i can i pass it the pre-fitted model, instead?

here's my closest attempt at that:

```
fit <- lm(RT~log_frequency, data=dataset)
fit$y0 <- coef(fit)[1]
fit$m <- coef(fit)[2]

plot <- ggplot(data=dataset, aes(x=log_frequency, y=RT)) +
  geom_point() +  # scatter points
  geom_hline(yintercept=mean(dataset$RT), color="red") +
  geom_abline(intercept=fit$y0, slope=fit$m, color="blue") +
  labs(x="Frequency", y="Response Time") +
  ggtitle("Words (Frequency/Response Time)")

print(plot)
```



but it seems both cursed, and is lacking the shaded regression area, which i don't really know how to access.