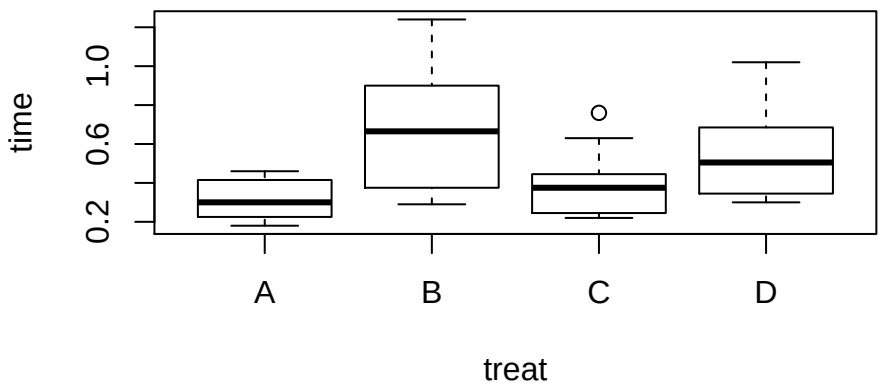
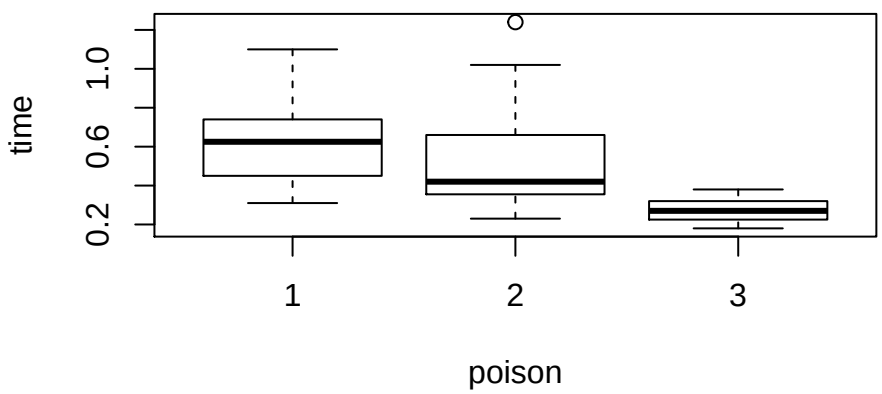
For this example, we will use data from a hypothetical experiment.[1] The survival time of the animal is given in units of 10 hours. The explanatory variables are poison (a factor with levels 1, 2, and 3 giving the type of poison used) and treatment (a factor with levels A, B, C, and D corresponding to the treatment given).

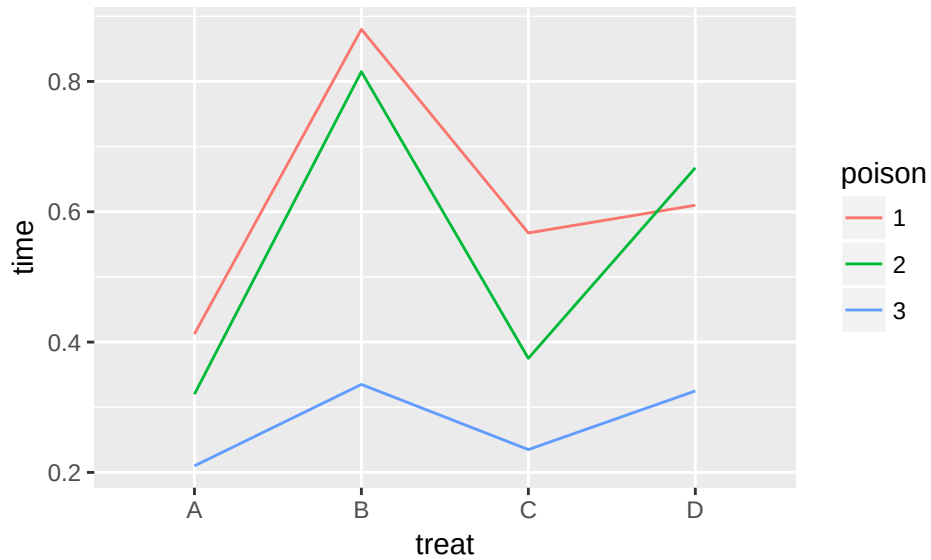
```
library(boot)
plot(time ~ treat + poison, data=poisons)
```



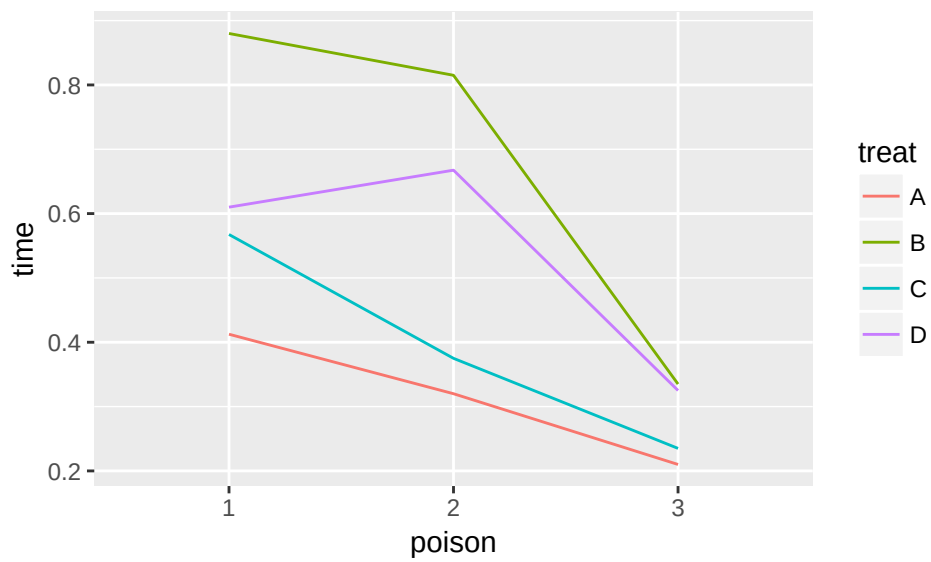


[1]Data is from Box & Cox (1964) An analysis of transformations. *Journal of the Royal Statistical Society, B* **26**, 211-252. This is *the* Box-Cox transformation paper.

```r
ggplot(poisons, aes(y=time, x=treat, group=poison, col = poison)) +
  stat_summary(fun.y = "mean", geom="line")
```




```r
ggplot(poisons, aes(y=time, x=poison, group=treat, col = treat)) +
  stat_summary(fun.y = "mean", geom="line")
```



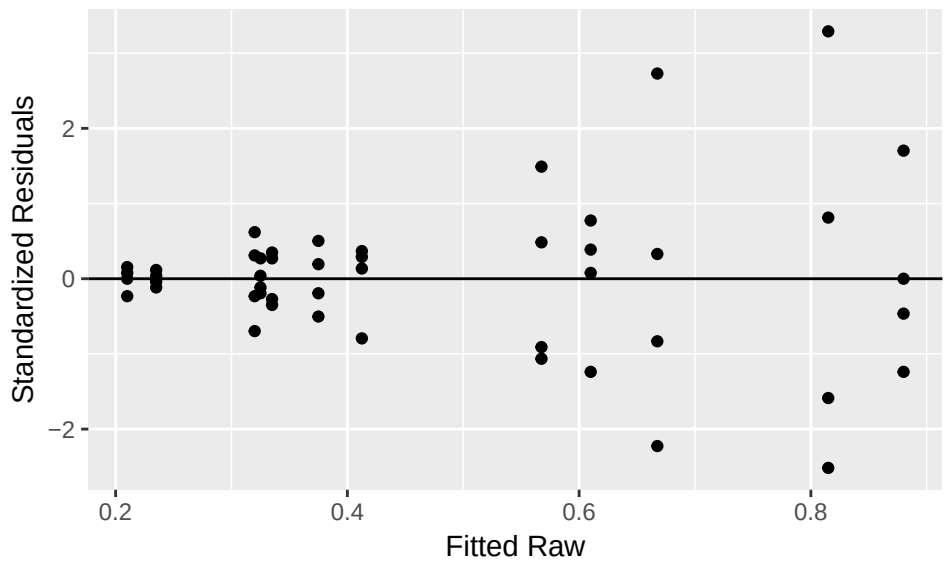
Do the plots look parallel? One problem with interaction plots is that we - of course - expect there to be some random variation regardless of whether or not the null hypothesis is true. It can be difficult to distinguish true interaction from noise. However, we can test for interaction.

```
rat.aov = aov(time ~ factor(poison)*treat, data=poisons)
summary(rat.aov)

##                       Df Sum Sq Mean Sq F value  Pr(>F)
## factor(poison)         2  1.033   0.517   23.22 3.3e-07 ***
## treat                  3  0.921   0.307   13.81 3.8e-06 ***
## factor(poison):treat   6  0.250   0.042    1.87    0.11
## Residuals             36  0.801   0.022
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that the interaction effect is not significant but the main effects are. Before we proceed (or conclude anything!), we need to check the diagnositcs (that is, does our model fit?). From below we can see that we do not have constant variance across all different conditions. We try transforming the response variable (time) using the natural log and the reciprocal. The reciprocal transformation seems to work for the model; additionally, we can use the natural interpret of the reciprocal transformation as the *rate of dying*.

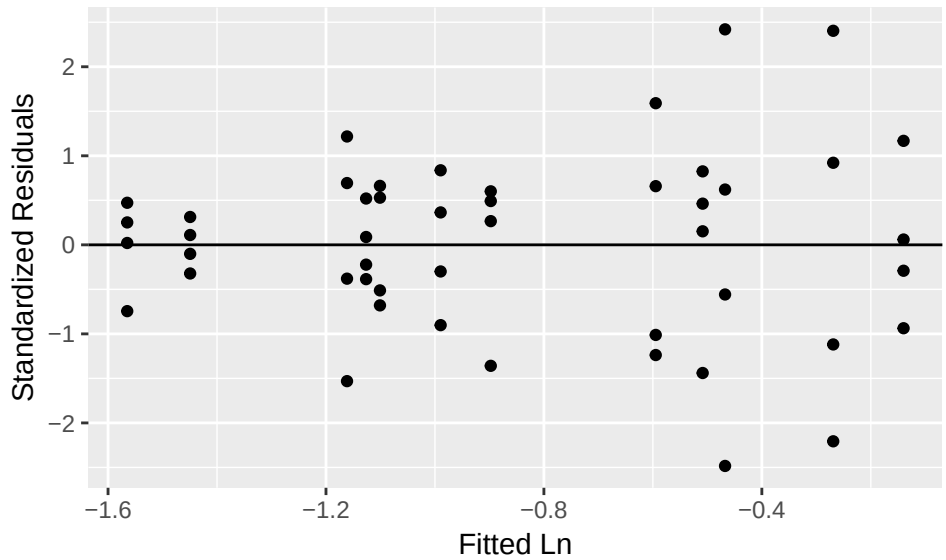
NOTE: There are *exatly* 12 possible fitted values (why?). If you look closely, you should see 12 vertical lines on which the residuals fall.

```
rat.aov <- aov(time ~ factor(poison)*treat, data=poisons)
rat.aov1 <- aov(log(time) ~ factor(poison)*treat, data=poisons)
rat.aov2 <- aov(1/time ~ factor(poison)*treat, data=poisons)

rat.aov %>% augment() %>%
  ggplot(aes(x=.fitted, y = .std.resid)) + geom_point() +
  geom_hline(yintercept = 0) +
  xlab("Fitted Raw")  +
  ylab("Standardized Residuals")
```

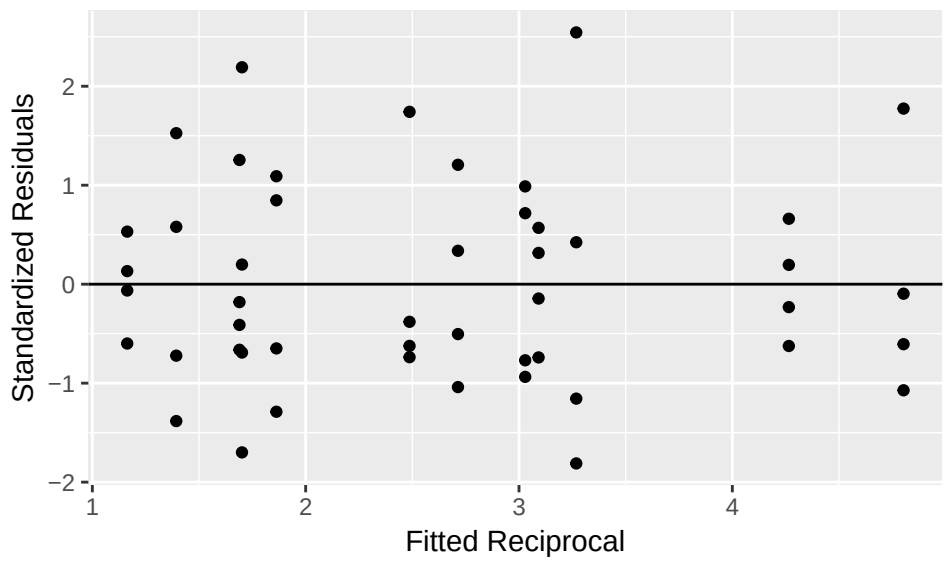



```
rat.aov1 %>% augment() %>%
  ggplot(aes(x=.fitted, y = .std.resid)) + geom_point() +
  geom_hline(yintercept = 0) +
```

```
  xlab("Fitted Ln") +
  ylab("Standardized Residuals")
```




```
rat.aov2 %>% augment() %>%
  ggplot(aes(x=.fitted, y = .std.resid)) + geom_point() +
  geom_hline(yintercept = 0) +
  xlab("Fitted Reciprocal") +
  ylab("Standardized Residuals")
```



We now summarize the model with 1/time as the response variable.

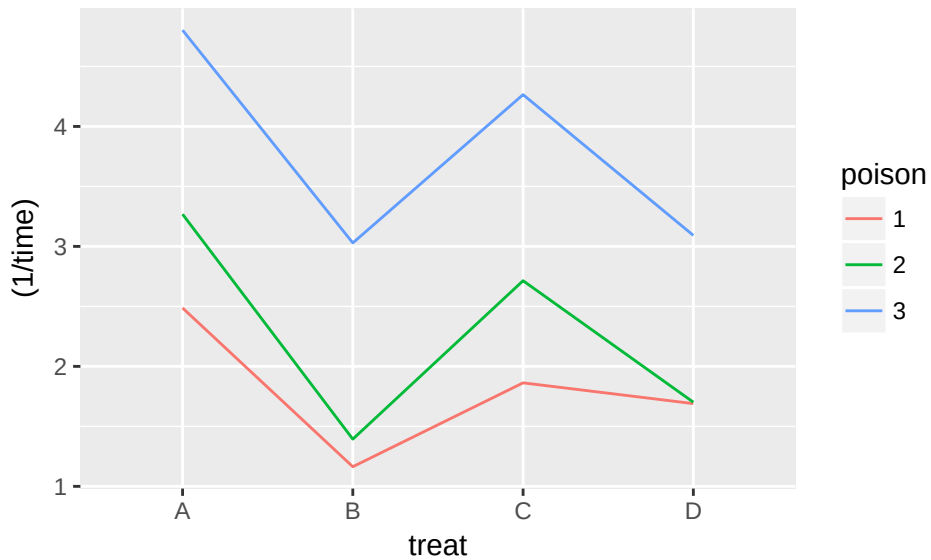
```
summary(rat.aov2)

##                        Df Sum Sq Mean Sq F value  Pr(>F)
```

```
## factor(poison)        2   34.9   17.44   72.63 2.3e-13 ***
## treat                 3   20.4    6.80   28.34 1.4e-09 ***
## factor(poison):treat  6    1.6    0.26    1.09    0.39
## Residuals            36    8.6    0.24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

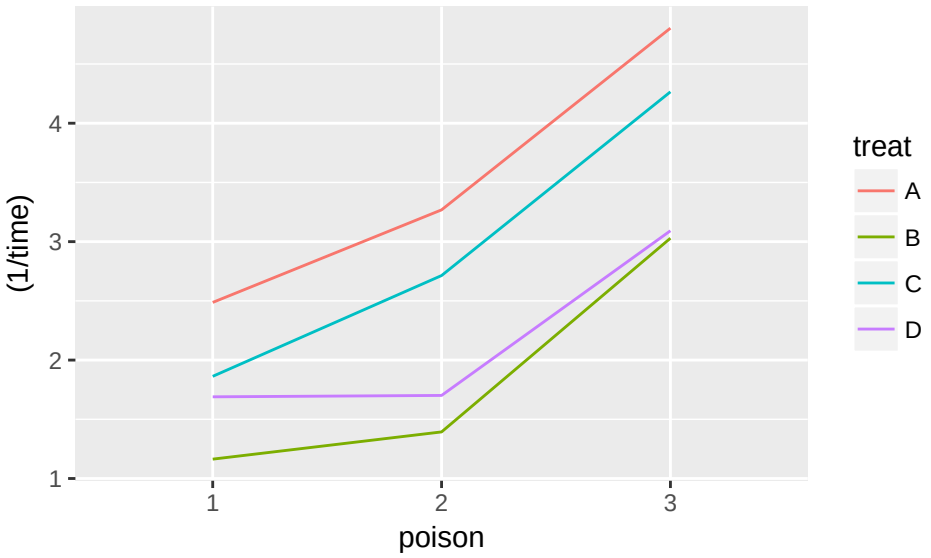
Interaction is still not significant, so we run the model without the interaction term. Both main effects remain significant.

```
summary(aov(1/time ~ factor(poison) + treat, data=poisons))

##                Df Sum Sq Mean Sq F value  Pr(>F)
## factor(poison)  2   34.9   17.44    71.7 2.9e-14 ***
## treat           3   20.4    6.80    28.0 4.2e-10 ***
## Residuals      42   10.2    0.24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ggplot(poisons, aes(y=(1/time), x=treat, group=poison, col = poison)) +
  stat_summary(fun.y = "mean", geom="line")
```




```
ggplot(poisons, aes(y=(1/time), x=poison, group=treat, col = treat)) +
  stat_summary(fun.y = "mean", geom="line")
```


(1/time)
4
3
2
1
1
2
3
poison
treat
A
B
C
D