final_project_419

Tyler Chun

11/24/2021

Approach: Remove point 8 and Treat Speed as numeric and Box-cox

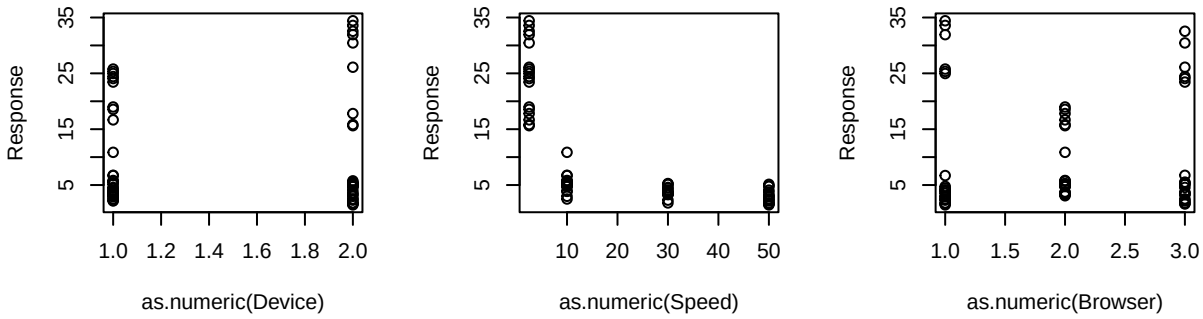
- Treat Speed as numeric
- Transformation of response

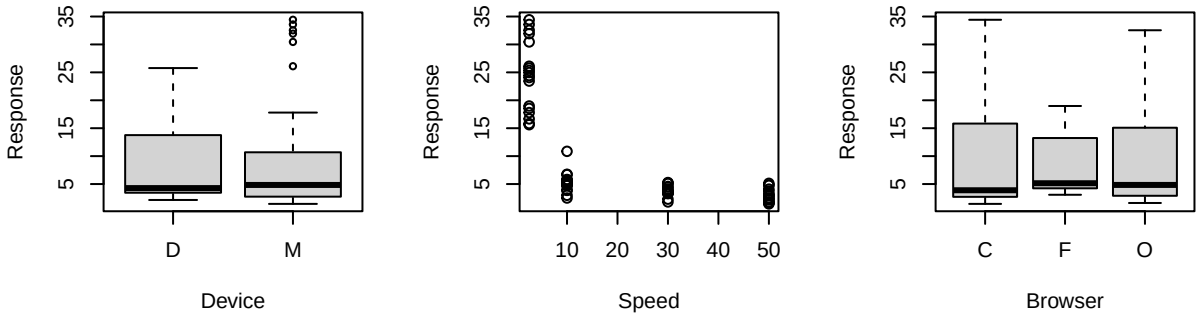
```
df = read.csv("df.csv")
# remove point 8
# df <- df[-8,]

# Do initial Analysis and fit model
Response = c(df$Replicate.1,df$Replicate.2,df$Replicate.3)

# Change Speed to original speed numbers
df$Speed_f=as.factor(df$Speed)

df$Speed[df$Speed_f==1]=50
df$Speed[df$Speed_f==2]=30
df$Speed[df$Speed_f==3]=10
df$Speed[df$Speed_f==4]=2.5

df
```

```
##     X Run Device Speed Browser Replicate.1 Replicate.2 Replicate.3 Speed_f
## 1   1   1      D  50.0       C        2.95        4.09        2.92       1
## 2   2   2      D  50.0       F        3.75        3.08        3.25       1
## 3   3   3      D  50.0       O        2.25        2.15        2.52       1
## 4   4   4      D  30.0       C        4.46        3.51        3.86       2
## 5   5   5      D  30.0       F        3.35        3.61        3.52       2
## 6   6   6      D  30.0       O        3.68        3.38        4.57       2
## 7   7   7      D  10.0       C        6.70        3.87        3.91       3
## 8   8   8      D  10.0       F       10.86        5.69        5.82       3
## 9   9   9      D  10.0       O        5.12        5.41        6.74       3
## 10 19  19      D   2.5       C       25.35       24.97       25.77       4
## 11 20  20      D   2.5       F       18.55       16.67       18.98       4
## 12 21  21      D   2.5       O       23.44       24.07       24.43       4
## 13 10  10      M  50.0       C        2.45        1.67        1.45       1
## 14 11  11      M  50.0       F        5.13        4.84        4.78       1
## 15 12  12      M  50.0       O        1.61        1.75        1.72       1
## 16 13  13      M  30.0       C        1.80        4.12        2.30       2
## 17 14  14      M  30.0       F        5.27        5.13        5.16       2
## 18 15  15      M  30.0       O        3.42        3.63        3.28       2
## 19 16  16      M  10.0       C        2.99        4.76        2.50       3
## 20 17  17      M  10.0       F        5.75        4.84        4.73       3
```

```
## 21 18  18      M  10.0       O        5.31        5.52        5.17       3
## 22 22  22      M   2.5       C       31.93       33.58       34.42       4
## 23 23  23      M   2.5       F       15.83       17.80       15.63       4
## 24 24  24      M   2.5       O       26.10       30.45       32.54       4
```

```
# With the new df (We don't remove run 8)
df$Device = as.factor(df$Device)
df$Browser = as.factor(df$Browser)
new_df =
data.frame(Device = rep(df$Device,3),
Speed = rep(df$Speed,3),
Browser = rep(df$Browser,3),
Response,
replicate = as.factor(c(rep(1,24),rep(2,24),rep(3,24))))


par(mfrow=c(2,3))
plot(data= new_df, Response~ as.numeric(Device) + as.numeric(Speed)+as.numeric(Browser))
plot(data= new_df, Response~ (Device) + (Speed)+(Browser))
```






```
par(mfrow=c(1,1))
m1 <-lm(data=new_df, Response~Device + Speed + Browser +
          Device:Speed + Device:Browser + Speed:Browser +
          replicate)
summary(m1)
```

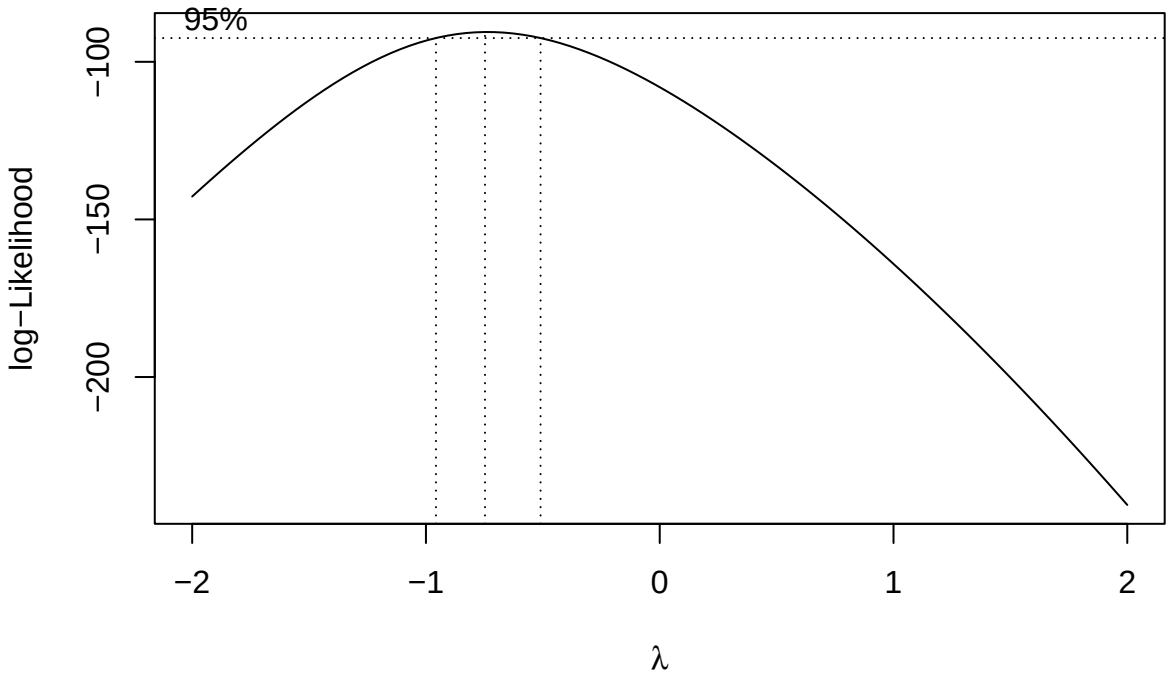
```
##
## Call:
## lm(formula = Response ~ Device + Speed + Browser + Device:Speed +
##     Device:Browser + Speed:Browser + replicate, data = new_df)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.647  -4.859   0.216   4.000  14.976
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      18.33111    3.29274   5.567 6.41e-07 ***
## DeviceM           2.13190    3.72610   0.572 0.569356
## Speed            -0.38927    0.09433  -4.127 0.000115 ***
## BrowserF         -5.82917    4.03277  -1.445 0.153534
## BrowserO         -0.70849    4.03277  -0.176 0.861134
## replicate2        0.02250    2.13599   0.011 0.991630
## replicate3        0.08000    2.13599   0.037 0.970248
## DeviceM:Speed    -0.05035    0.09433  -0.534 0.595477
## DeviceM:BrowserF -1.15417    4.27198  -0.270 0.787956
## DeviceM:BrowserO  0.09417    4.27198   0.022 0.982487
## Speed:BrowserF    0.19719    0.11554   1.707 0.093041 .
## Speed:BrowserO    0.01406    0.11554   0.122 0.903541
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.399 on 60 degrees of freedom
## Multiple R-squared:  0.4924, Adjusted R-squared:  0.3993
## F-statistic:  5.29 on 11 and 60 DF,  p-value: 8.865e-06
```

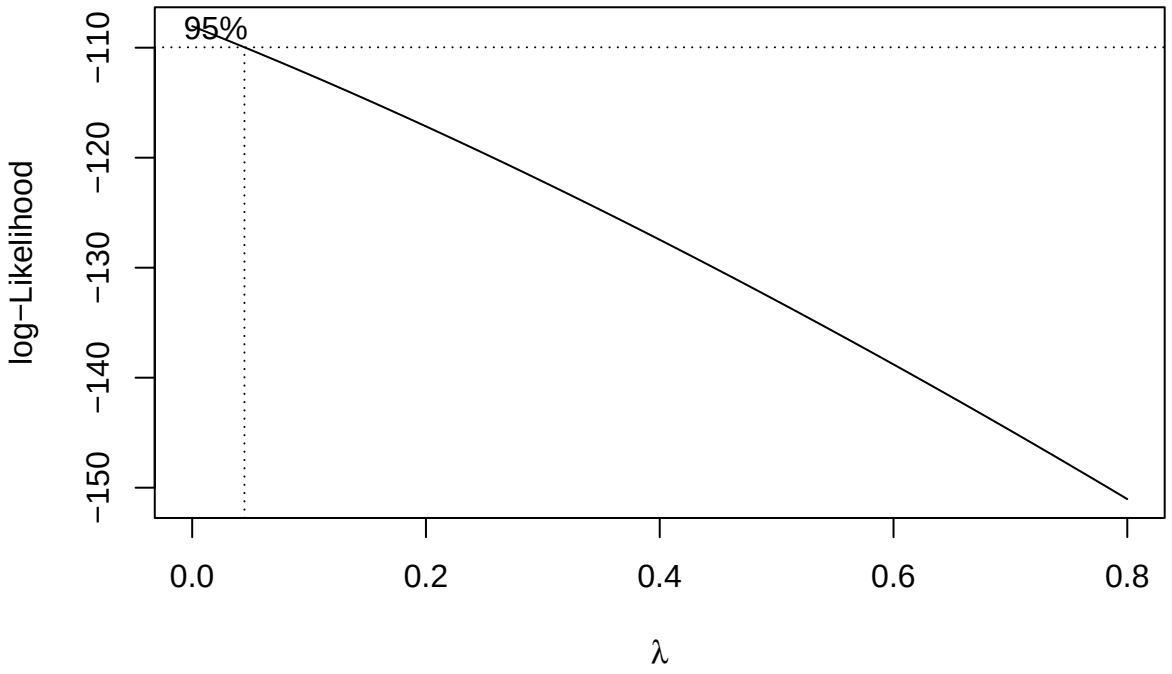
```
anova(m1)
```

```
## Analysis of Variance Table
##
## Response: Response
##                Df Sum Sq Mean Sq F value    Pr(>F)
## Device          1    6.8    6.79  0.1240    0.7260
## Speed           1 2912.7 2912.73 53.2010 7.879e-10 ***
## Browser         2   46.4   23.21  0.4239    0.6564
## replicate       2    0.1    0.04  0.0007    0.9993
## Device:Speed    1   15.6   15.60  0.2849    0.5955
## Device:Browser  2    5.8    2.90  0.0530    0.9485
## Speed:Browser   2  198.6   99.28  1.8134    0.1719
## Residuals      60 3285.0   54.75
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

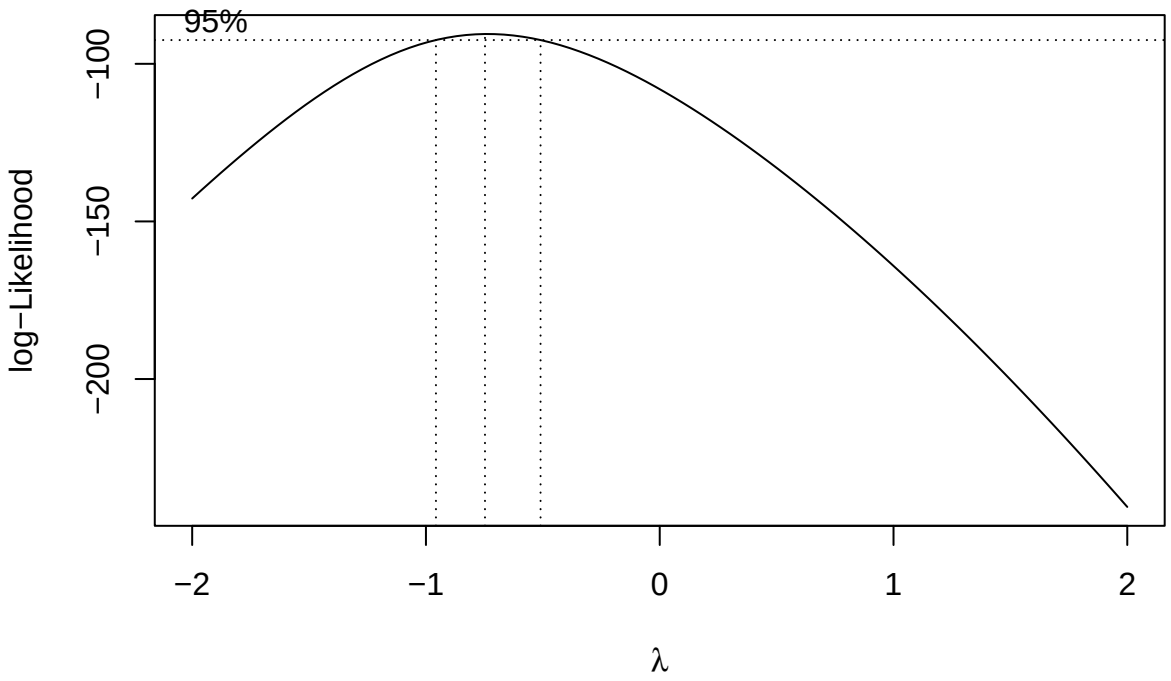
```
library(MASS)


# Box-Cox Transformation
boxcox(m1, plotit=T)
```




```
boxcox(m1, plotit=T, lambda = seq(0, 0.8, by=0.1))
```




```
# To see the exact best lambda:
lmod1_bc <- boxcox(m1)
```

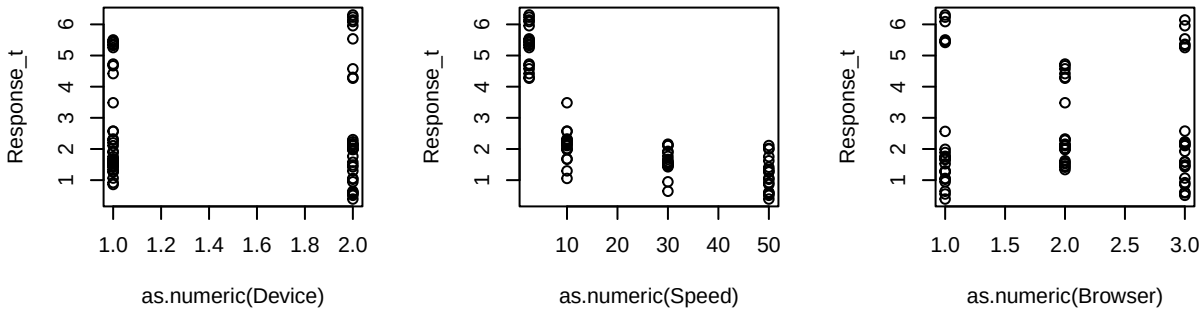


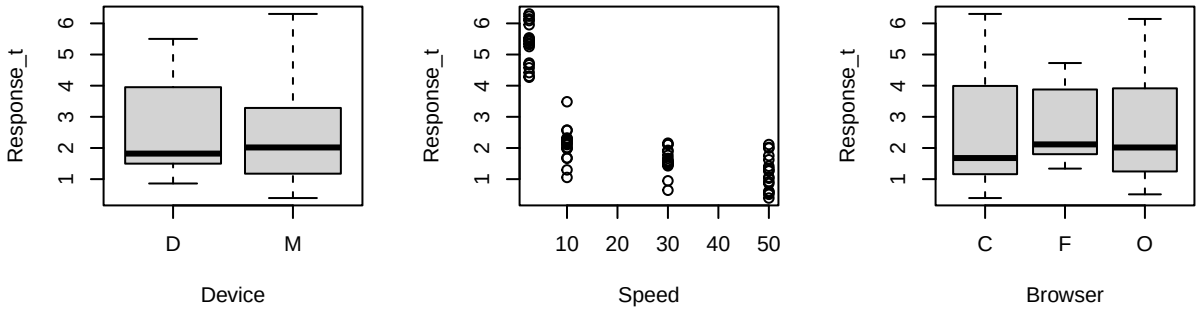

```
lambda <- lmod1_bc$x[which.max(lmod1_bc$y)]
lambda
```

```
## [1] -0.7474747
```

```
#2. We see the best lambda is 0.3030303, round to 0.3. Create a new transformed response using Box-Cox
lambda <- 0.3
Response_t <- (Responseˆlambda - 1) / lambda
```

```
# Re-do all analysis with transformed y

# After removing 8, only has 17 rows
# Set variables to factors
df$Device = as.factor(df$Device)
df$Browser = as.factor(df$Browser)

# Create a new dataframe with the variables and transformed response
new_df =
data.frame(Device = rep(df$Device,3),
Speed = rep(df$Speed,3),
Speed_f = rep(df$Speed_f,3),
Browser = rep(df$Browser,3),
Response_t,
replicate = as.factor(c(rep(1,24),rep(2,24),rep(3,24))))

# Plot
par(mfrow=c(2,3))
plot(data= new_df, Response_t~ as.numeric(Device) + as.numeric(Speed)+as.numeric(Browser))
plot(data= new_df, Response_t~ (Device) + (Speed)+(Browser))
```






```
par(mfrow=c(1,1))
m1 <-lm(data=new_df, Response_t~(Device + Speed + Browser)^2 + replicate)
summary(m1)
```

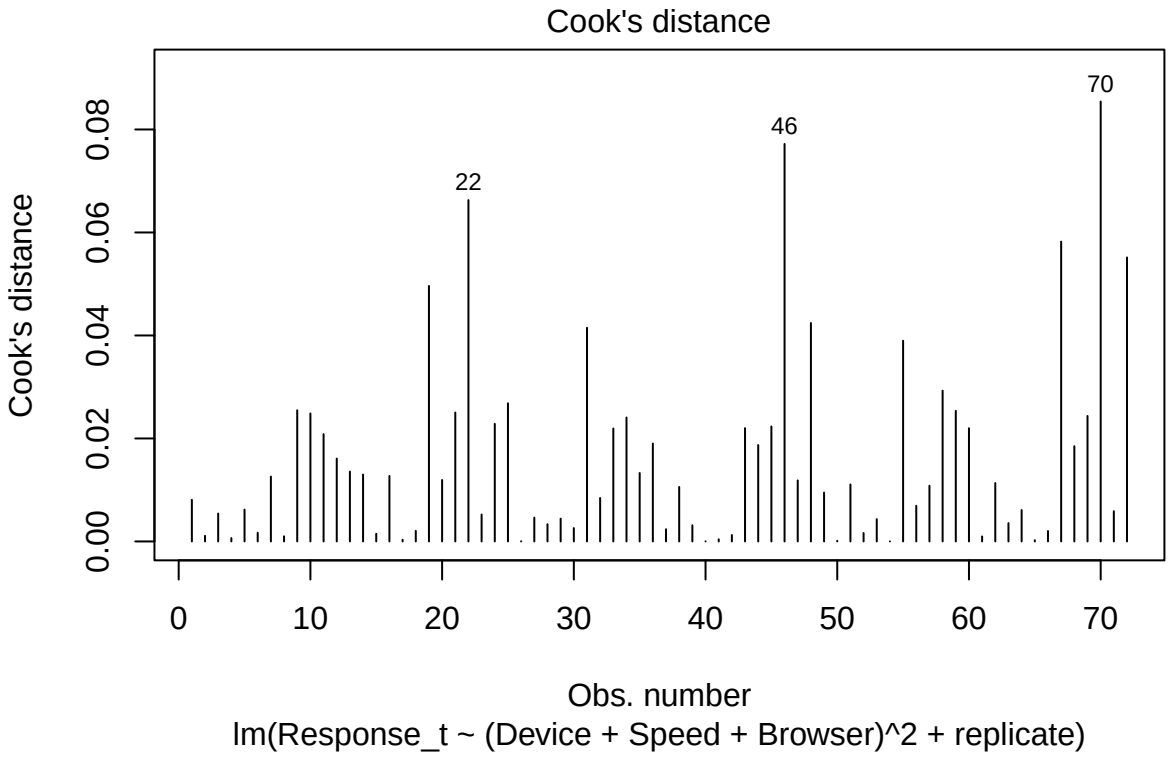
```
##
## Call:
## lm(formula = Response_t ~ (Device + Speed + Browser)^2 + replicate,
##     data = new_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.31075 -0.82984  0.01824  0.76724  2.34137
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.380477   0.532977   8.219 2.08e-11 ***
## DeviceM           -0.155540   0.603122  -0.258    0.797
## Speed             -0.074095   0.015269  -4.853 9.04e-06 ***
## BrowserF          -0.753399   0.652761  -1.154    0.253
## BrowserO           0.058087   0.652761   0.089    0.929
## replicate2        -0.019607   0.345741  -0.057    0.955
## replicate3        -0.064245   0.345741  -0.186    0.853
## DeviceM:Speed     -0.005438   0.015269  -0.356    0.723
## DeviceM:BrowserF   0.374418   0.691481   0.541    0.590
## DeviceM:BrowserO   0.256429   0.691481   0.371    0.712
## Speed:BrowserF     0.029565   0.018701   1.581    0.119
## Speed:BrowserO    -0.006176   0.018701  -0.330    0.742
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.198 on 60 degrees of freedom
## Multiple R-squared:  0.5906, Adjusted R-squared:  0.5155
## F-statistic: 7.868 on 11 and 60 DF,  p-value: 3.012e-08
```

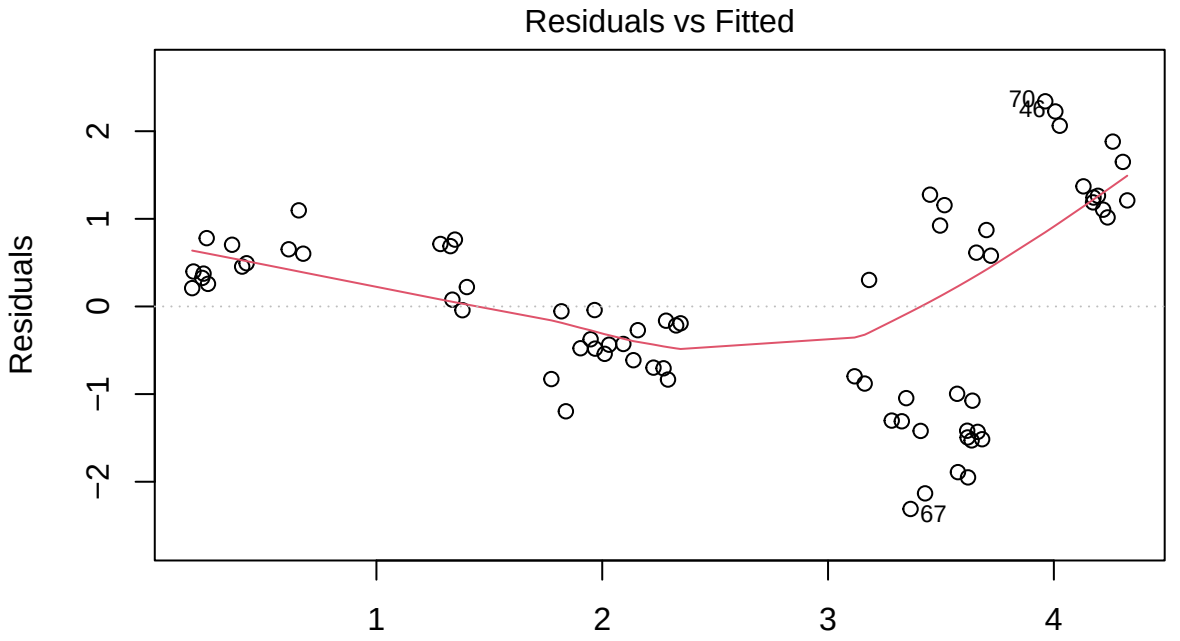
```
anova(m1)
```

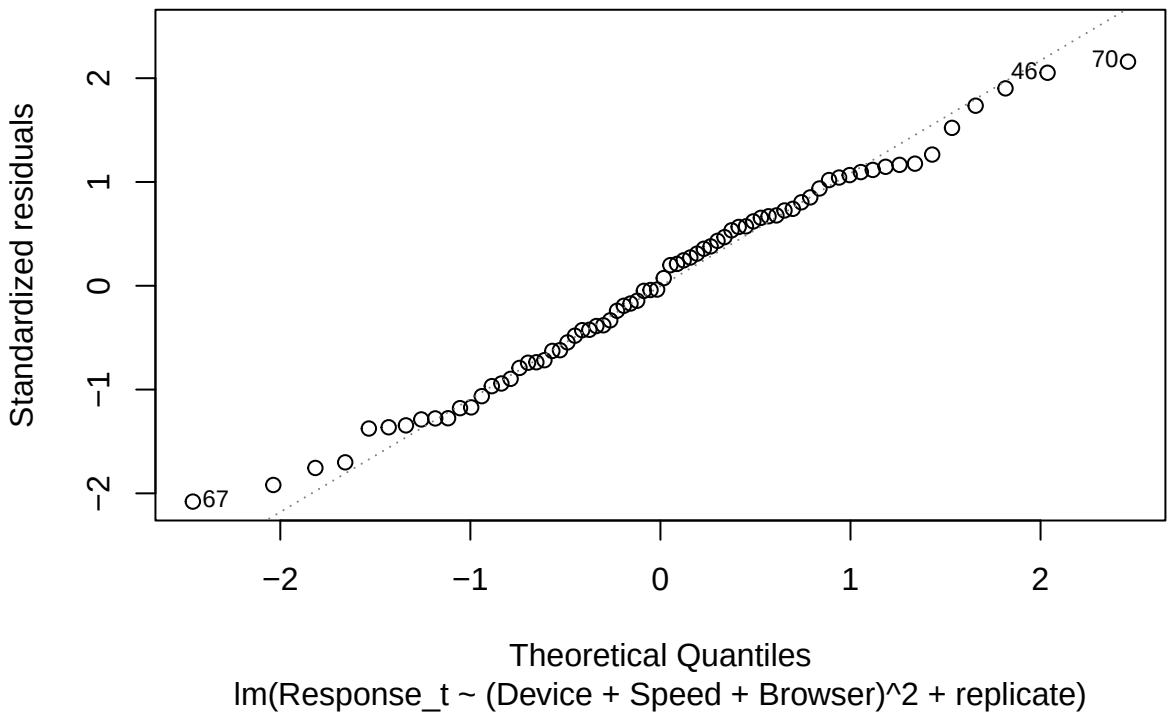
```
## Analysis of Variance Table
##
## Response: Response_t
##                Df  Sum Sq Mean Sq F value    Pr(>F)
## Device          1   0.091   0.091  0.0633    0.8022
## Speed           1 117.225 117.225 81.7221 8.453e-13 ***
## Browser         2   0.169   0.085  0.0591    0.9427
## replicate       2   0.052   0.026  0.0181    0.9820
## Device:Speed    1   0.182   0.182  0.1268    0.7230
## Device:Browser  2   0.440   0.220  0.1533    0.8582
## Speed:Browser   2   5.987   2.994  2.0870    0.1330
## Residuals      60  86.066   1.434
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

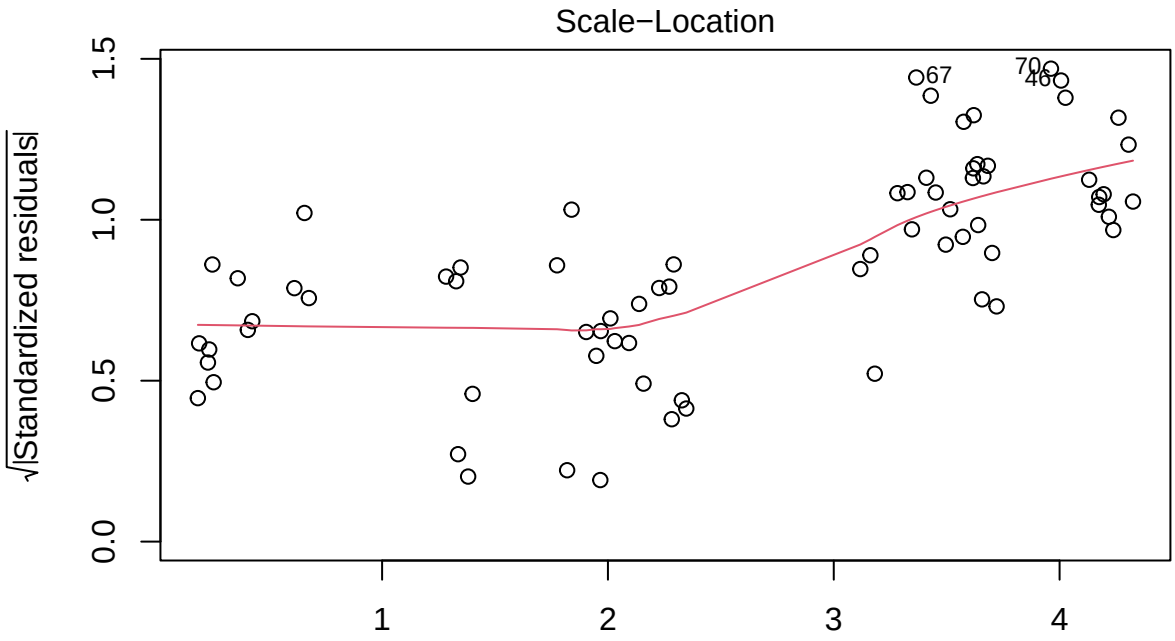
```
# Cook's distance
plot(m1,4)
```

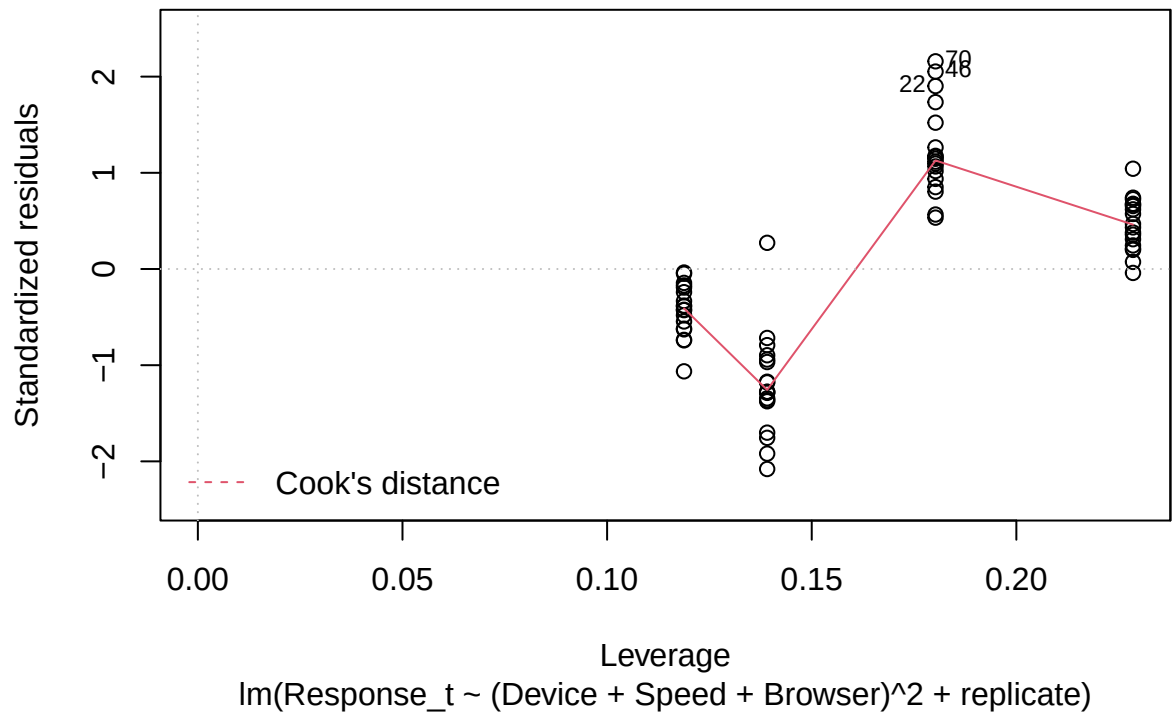



```
# Residual vs Fitted
plot(m1)
```

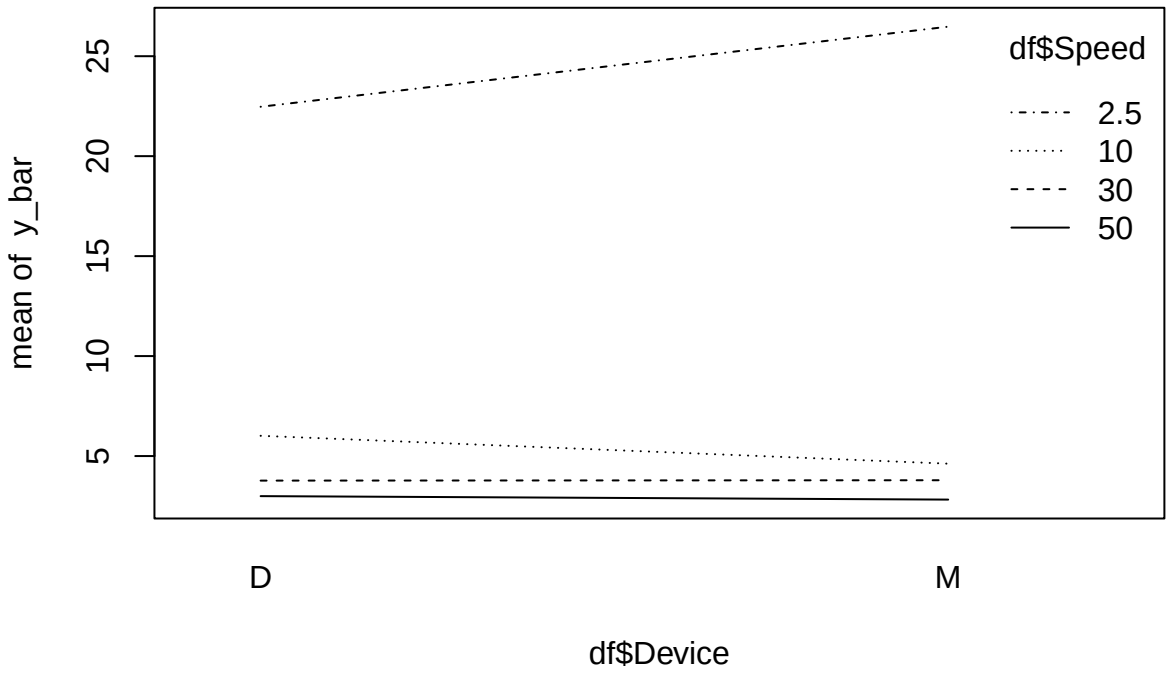

Residuals vs Fitted
Residuals
Fitted values
lm(Response_t ~ (Device + Speed + Browser)^2 + replicate)
70
46
67


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Response_t ~ (Device + Speed + Browser)^2 + replicate)
70
46
67

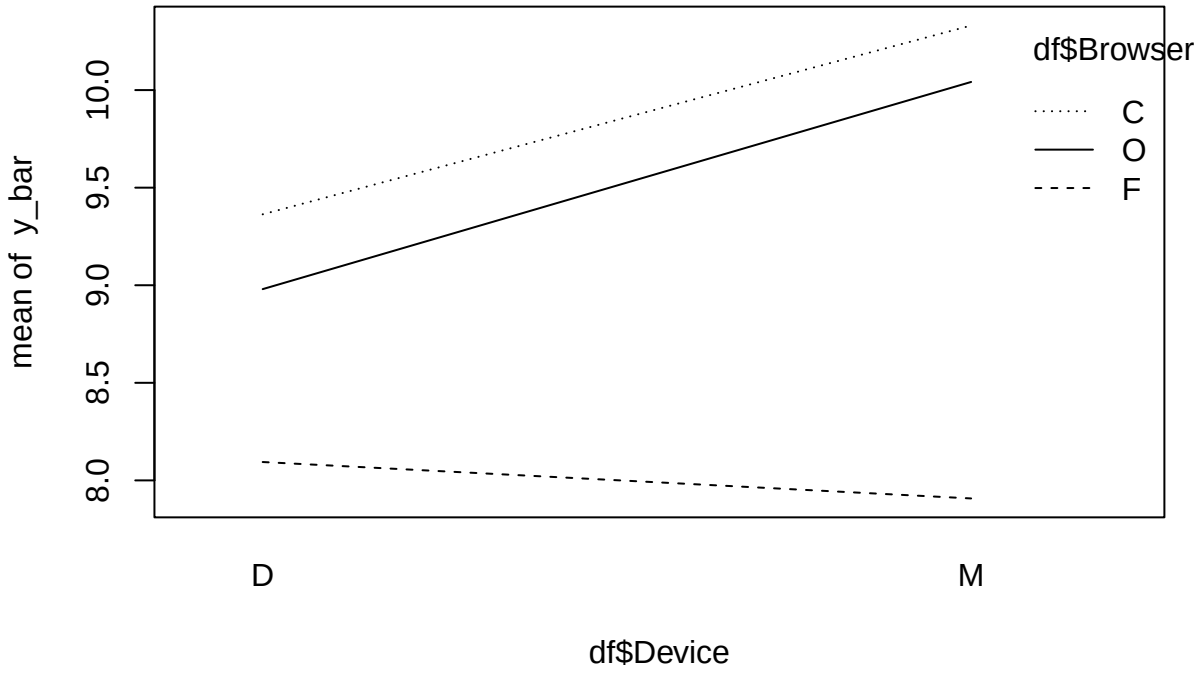




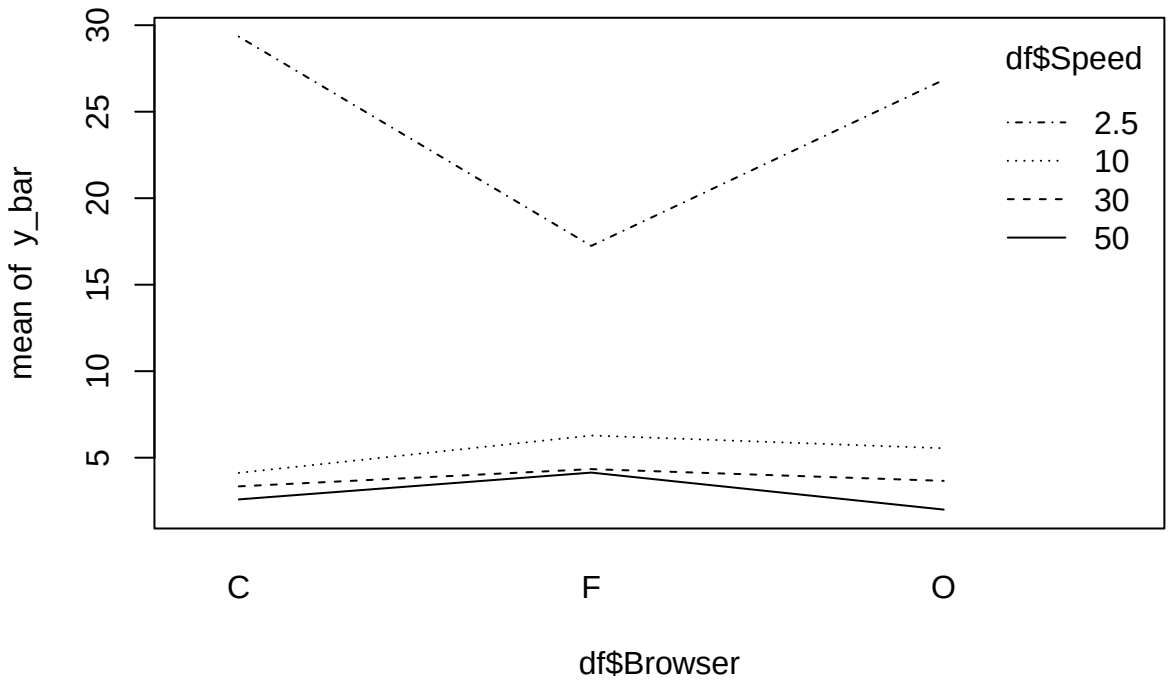

```
# Interaction plots
y_bar = (df$Replicate.1 + df$Replicate.2 + df$Replicate.3)/3
interaction.plot(df$Device, df$Speed, y_bar)
```




```
interaction.plot(df$Device, df$Browser, y_bar)
```




```
interaction.plot(df$Browser, df$Speed, y_bar)
```




```
# Tukey doesn't work with numeric factor
#plot(TukeyHSD(aov(m1)))

#TukeyHSD(aov(m1))
```