final_project_419

Tyler Chun

11/24/2021

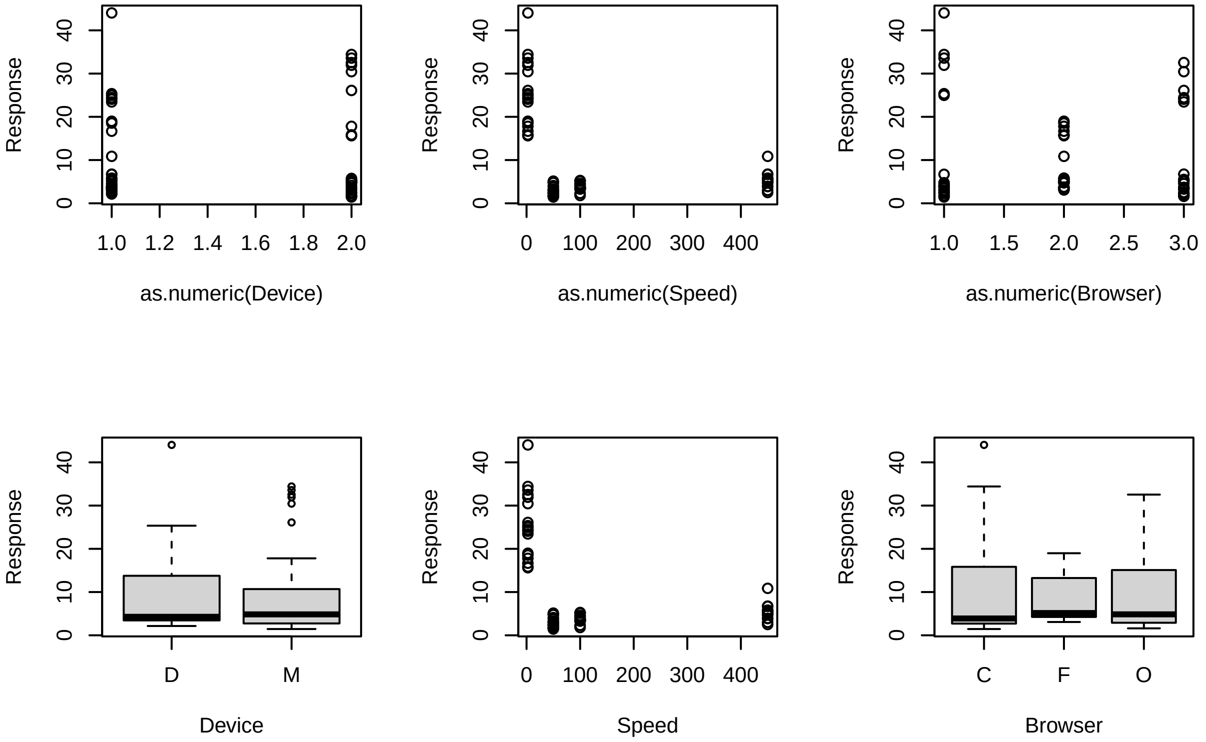
Approach: Remove point 8 and Treat Speed as numeric and Box-cox

- 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]=100
df$Speed[df$Speed_f==3]=450
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 100.0       C        4.46        3.51        3.86       2
## 5   5   5      D 100.0       F        3.35        3.61        3.52       2
## 6   6   6      D 100.0       O        3.68        3.38        4.57       2
## 7   7   7      D 450.0       C        6.70        3.87        3.91       3
## 8   8   8      D 450.0       F       10.86        5.69        5.82       3
## 9   9   9      D 450.0       O        5.12        5.41        6.74       3
## 10 19  19      D   2.5       C       44.03       25.35       24.97       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 100.0       C        1.80        4.12        2.30       2
## 17 14  14      M 100.0       F        5.27        5.13        5.16       2
## 18 15  15      M 100.0       O        3.42        3.63        3.28       2
## 19 16  16      M 450.0       C        2.99        4.76        2.50       3
## 20 17  17      M 450.0       F        5.75        4.84        4.73       3
## 21 18  18      M 450.0       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)
```

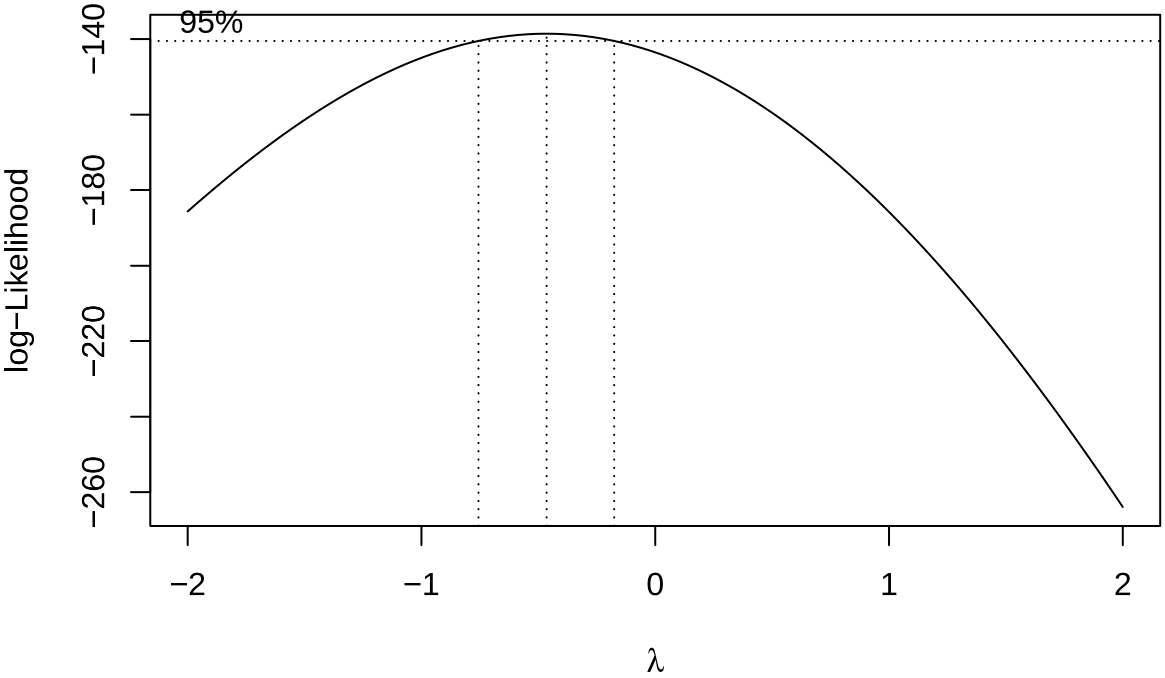
```
##
## Call:
## lm(formula = Response ~ Device + Speed + Browser + Device:Speed +
##     Device:Browser + Speed:Browser + replicate, data = new_df)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -12.153  -7.854  -1.508   5.214  28.309 
## 
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      15.794249   3.922449   4.027 0.000161 ***
## DeviceM           0.286841   4.584823   0.063 0.950322    
## Speed            -0.029336   0.013469  -2.178 0.033340 *  
## BrowserF         -6.042343   4.803999  -1.258 0.213347    
## BrowserO         -3.468841   4.803999  -0.722 0.473054    
## replicate2       -0.740000   2.907317  -0.255 0.799956    
## replicate3       -0.731667   2.907317  -0.252 0.802161    
## DeviceM:Speed    -0.005583   0.013469  -0.415 0.679946    
## DeviceM:BrowserF  0.367500   5.814634   0.063 0.949815    
## DeviceM:BrowserO  1.615833   5.814634   0.278 0.782052    
## Speed:BrowserF    0.021587   0.016496   1.309 0.195645    
## Speed:BrowserO    0.010382   0.016496   0.629 0.531474    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 10.07 on 60 degrees of freedom
## Multiple R-squared:  0.1785, Adjusted R-squared:  0.02785 
## F-statistic: 1.185 on 11 and 60 DF,  p-value: 0.3168
```

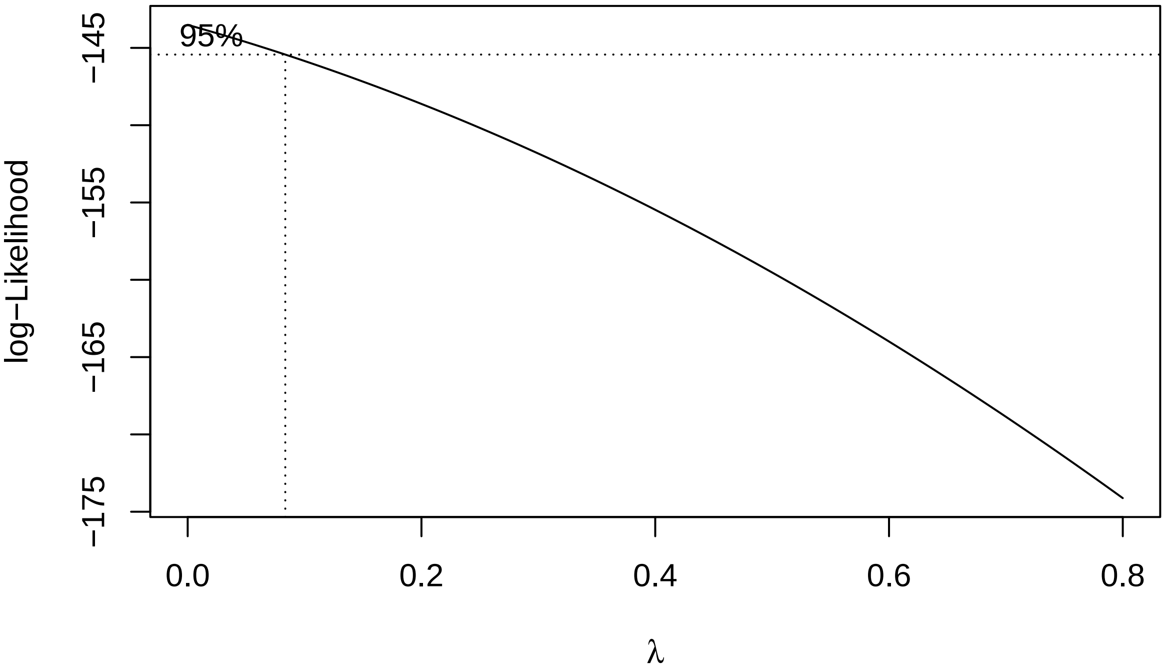
```
anova(m1)
```

```
## Analysis of Variance Table
## 
## Response: Response
##                Df Sum Sq Mean Sq F value   Pr(>F)   
## Device          1    0.2    0.21  0.0020 0.964216   
## Speed           1 1031.1 1031.08 10.1655 0.002274 **
## Browser         2   82.2   41.12  0.4054 0.668505   
## replicate       2    8.7    4.33  0.0427 0.958219   
## Device:Speed    1   17.4   17.43  0.1719 0.679946   
## Device:Browser  2    8.6    4.30  0.0424 0.958480   
## Speed:Browser   2  173.8   86.89  0.8567 0.429692   
## Residuals      60 6085.8  101.43                    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

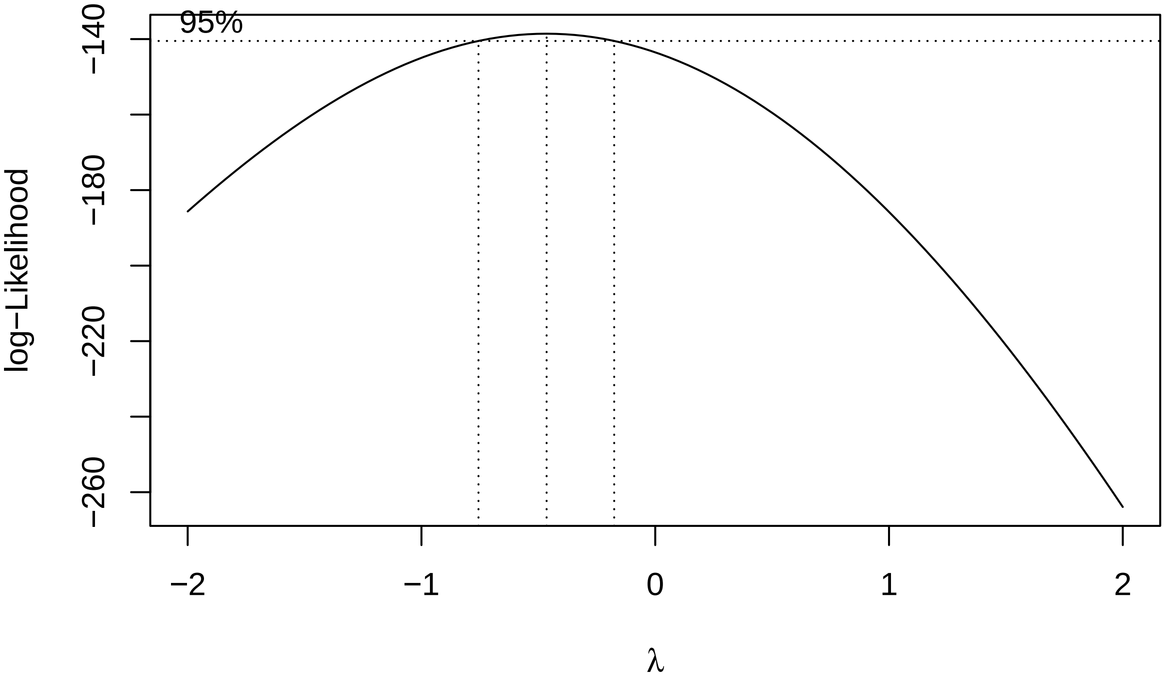
```
library(MASS)


# 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)
```

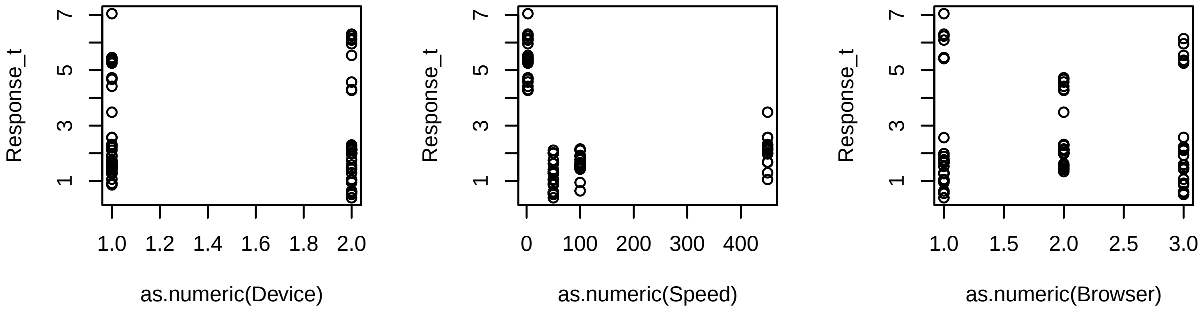


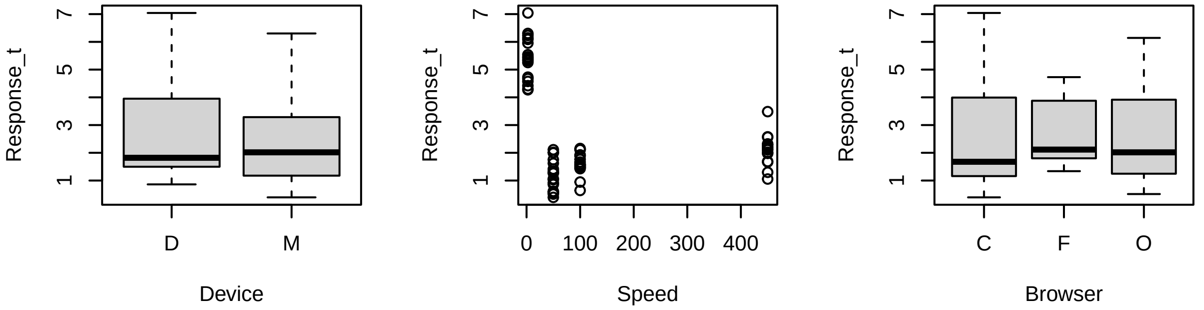
```
lambda <- lmod1_bc$x[which.max(lmod1_bc$y)]
lambda
```

```
## [1] -0.4646465
```

```
#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 <- (Response^lambda - 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))
```





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

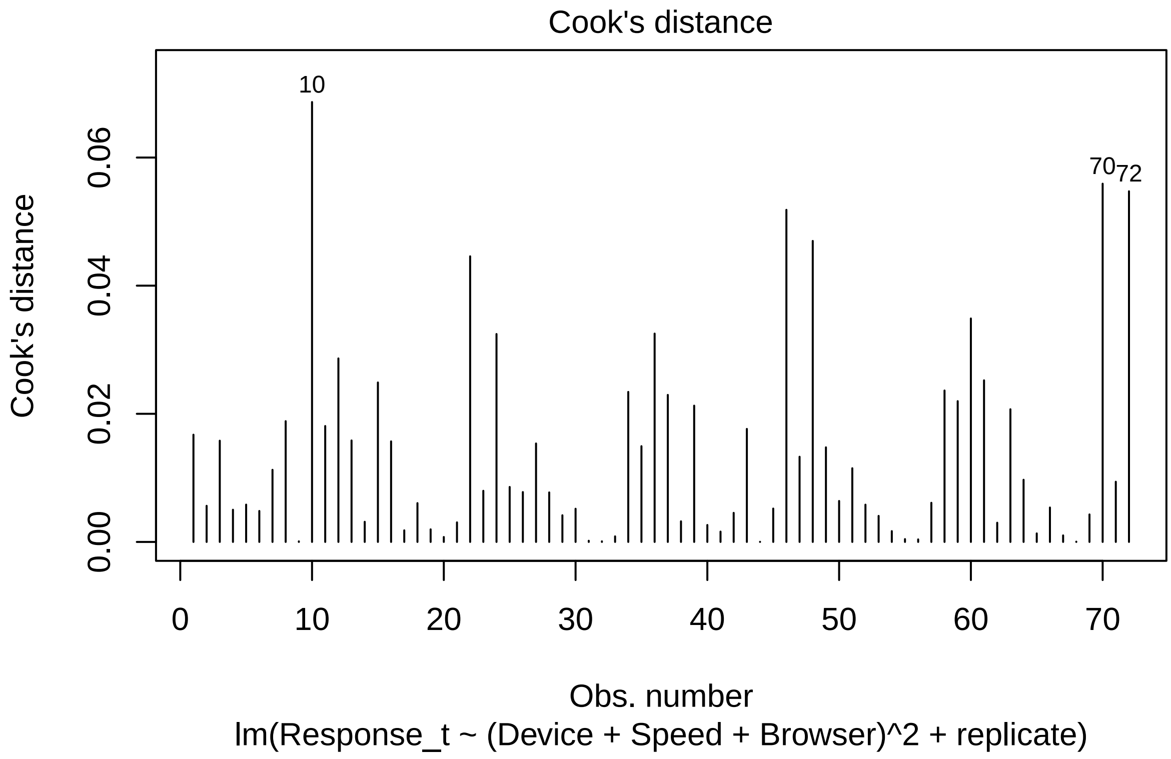
```
##
## Call:
## lm(formula = Response_t ~ (Device + Speed + Browser)^2 + replicate,
##     data = new_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4155 -1.2453 -0.2758  1.2114  3.6049
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.447259   0.705849   4.884 8.07e-06 ***
## DeviceM           -0.250681   0.825044  -0.304    0.762
## Speed             -0.004032   0.002424  -1.663    0.101
## BrowserF          -0.622590   0.864485  -0.720    0.474
## BrowserO          -0.516319   0.864485  -0.597    0.553
## replicate2        -0.083937   0.523175  -0.160    0.873
## replicate3        -0.133698   0.523175  -0.256    0.799
## DeviceM:Speed     -0.001055   0.002424  -0.435    0.665
## DeviceM:BrowserF   0.502770   1.046350   0.480    0.633
## DeviceM:BrowserO   0.384782   1.046350   0.368    0.714
## Speed:BrowserF     0.002818   0.002968   0.949    0.346
## Speed:BrowserO     0.002013   0.002968   0.678    0.500
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.812 on 60 degrees of freedom
## Multiple R-squared:  0.1108, Adjusted R-squared:  -0.05217
## F-statistic: 0.6799 on 11 and 60 DF,  p-value: 0.7518
```

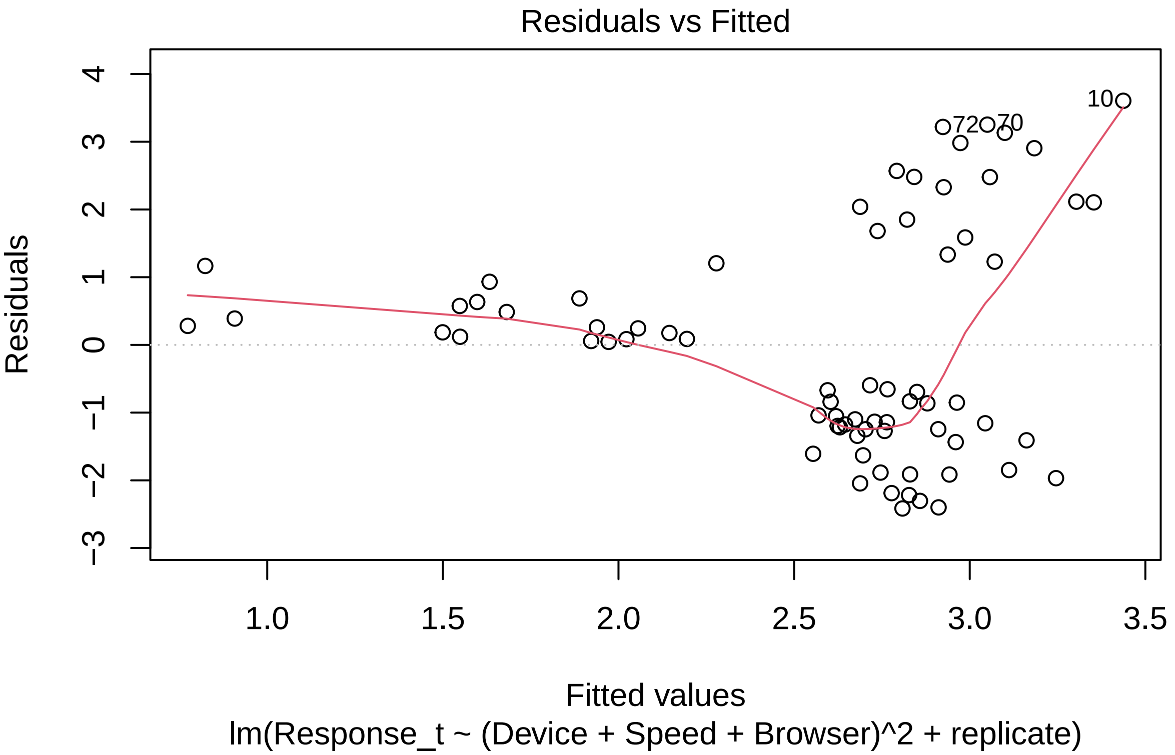
```
anova(m1)
```

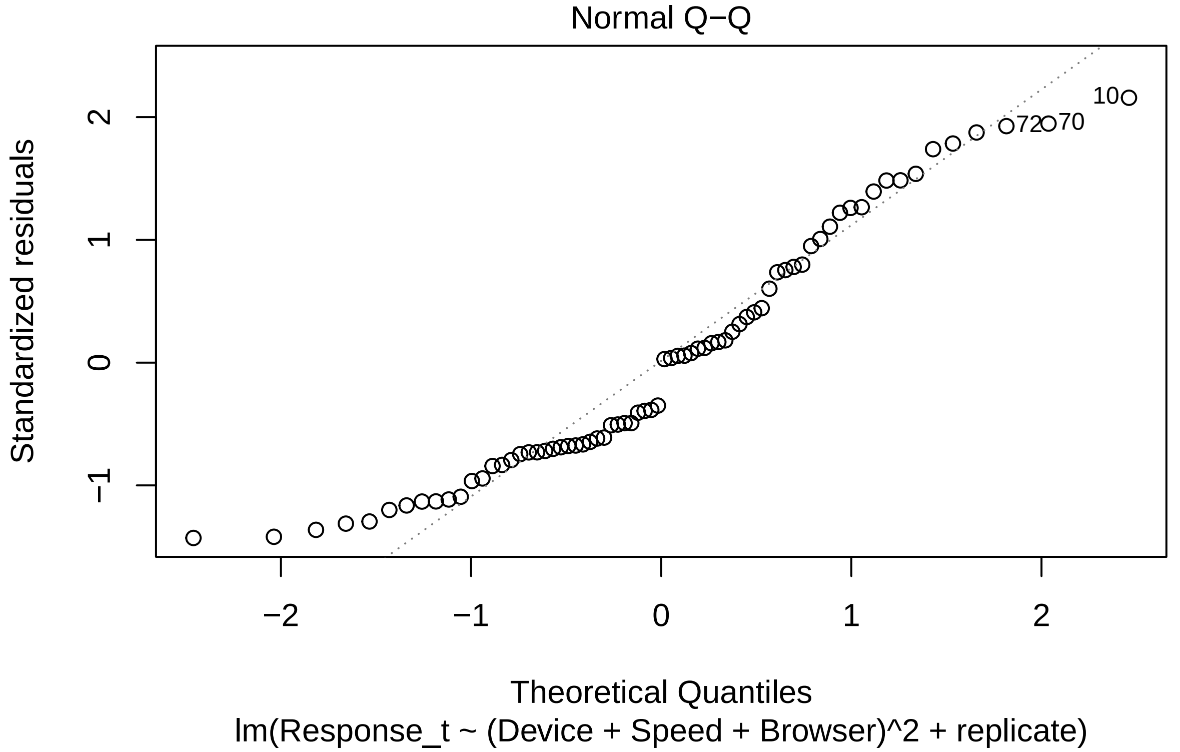
```
## Analysis of Variance Table
##
## Response: Response_t
##                 Df  Sum Sq Mean Sq F value  Pr(>F)
## Device           1   0.233  0.2331  0.0710 0.79085
## Speed            1  19.449 19.4494  5.9215 0.01795 *
## Browser          2   0.070  0.0350  0.0107 0.98940
## replicate        2   0.219  0.1096  0.0334 0.96720
## Device:Speed     1   0.623  0.6228  0.1896 0.66480
## Device:Browser   2   0.830  0.4148  0.1263 0.88161
## Speed:Browser    2   3.142  1.5712  0.4784 0.62214
## Residuals       60 197.073  3.2845
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

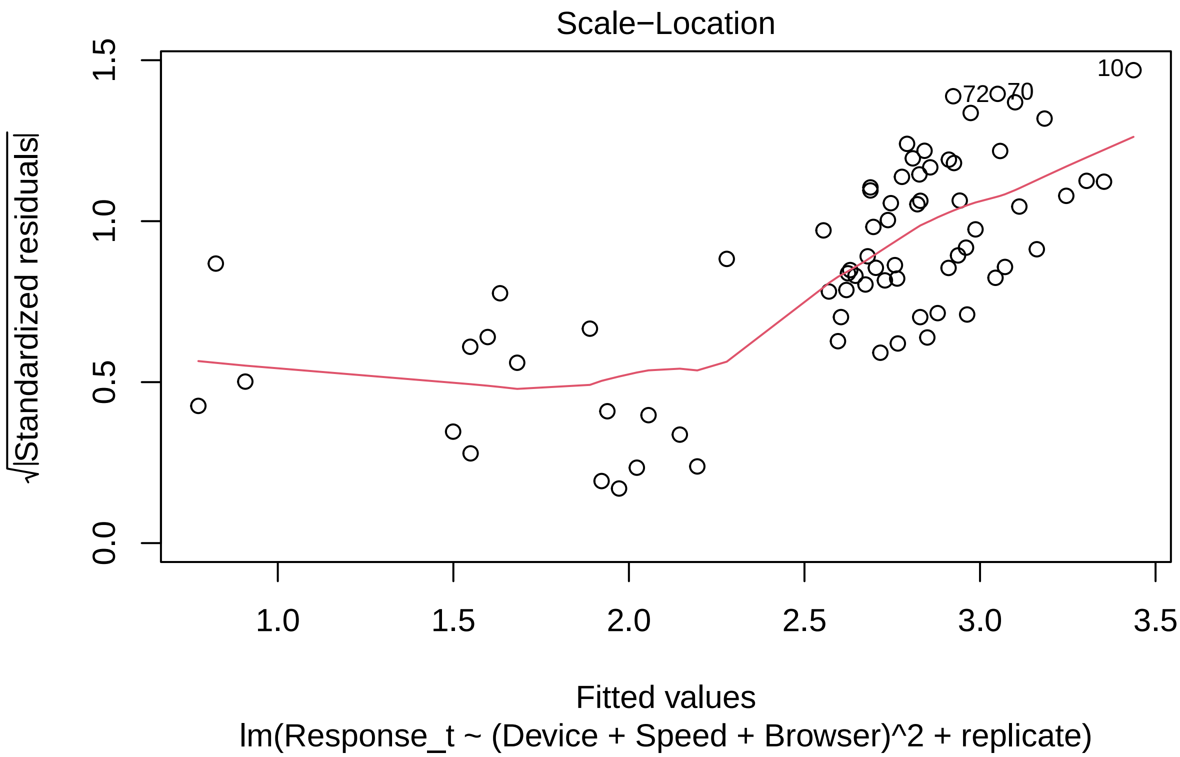
```
# Cook's distance
plot(m1,4)
```

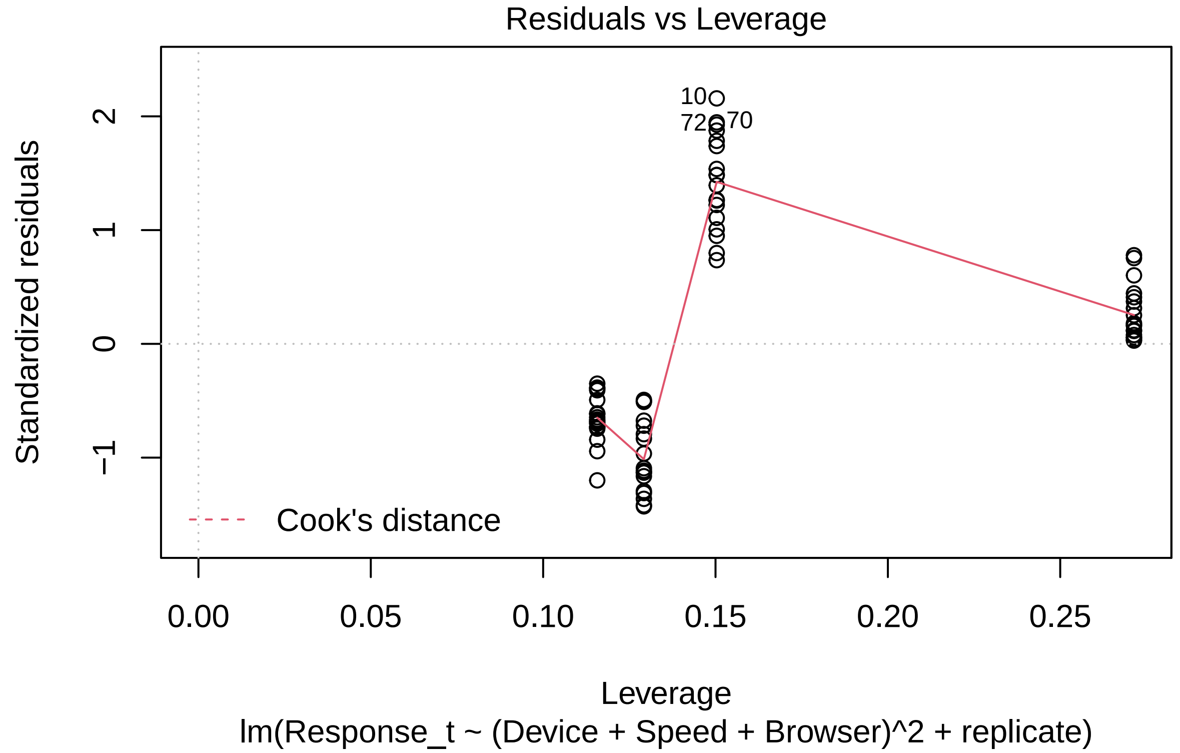


```
# Residual vs Fitted
plot(m1)
```

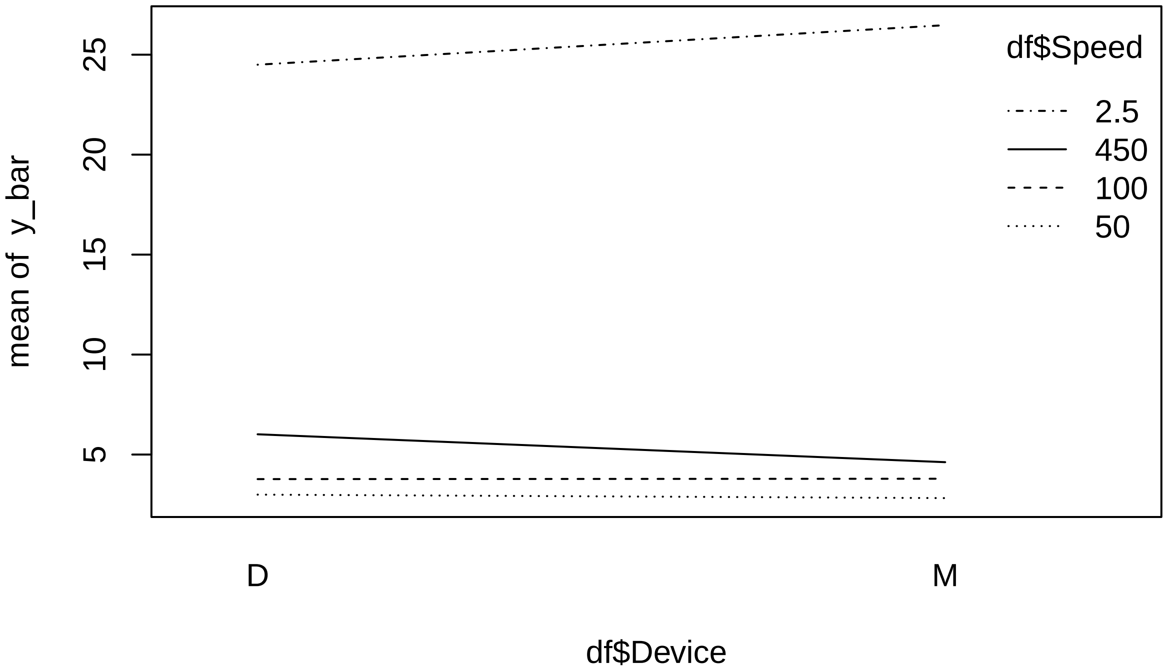

Residuals vs Fitted
Residuals
4
3
2
1
0
−1
−2
−3
10
72
70
1.0
1.5
2.0
2.5
3.0
3.5
Fitted values
lm(Response_t ~ (Device + Speed + Browser)^2 + replicate)


Normal Q−Q
Standardized residuals
2
1
0
−1
10
72
70
−2
−1
0
1
2
Theoretical Quantiles
lm(Response_t ~ (Device + Speed + Browser)^2 + replicate)

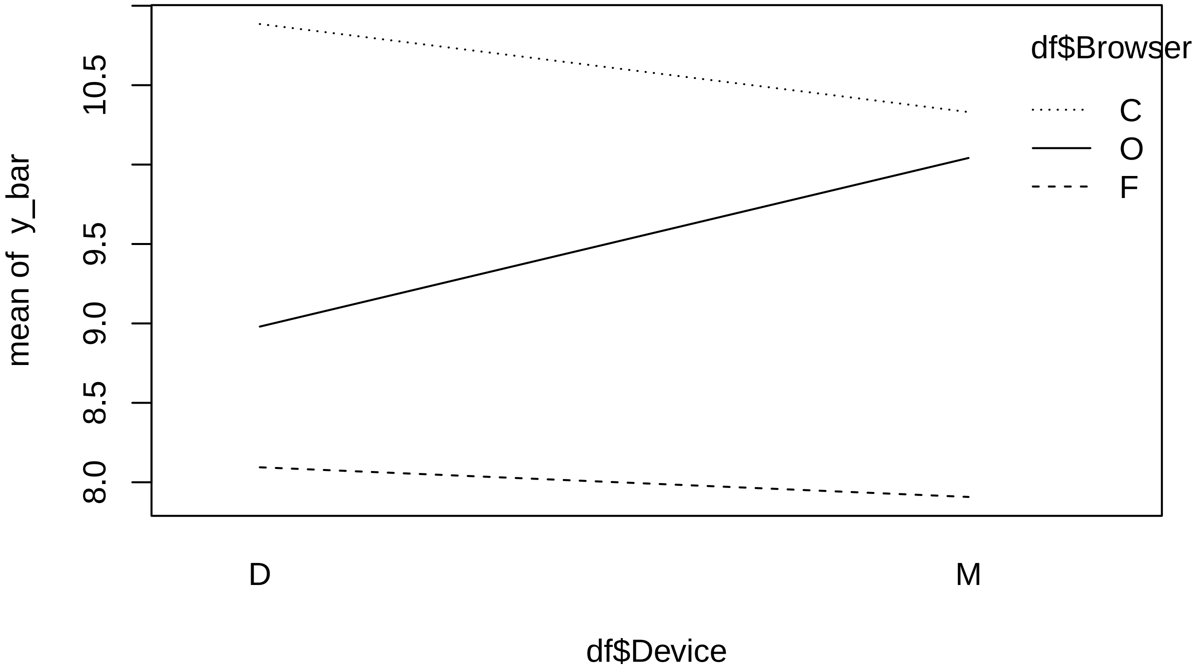




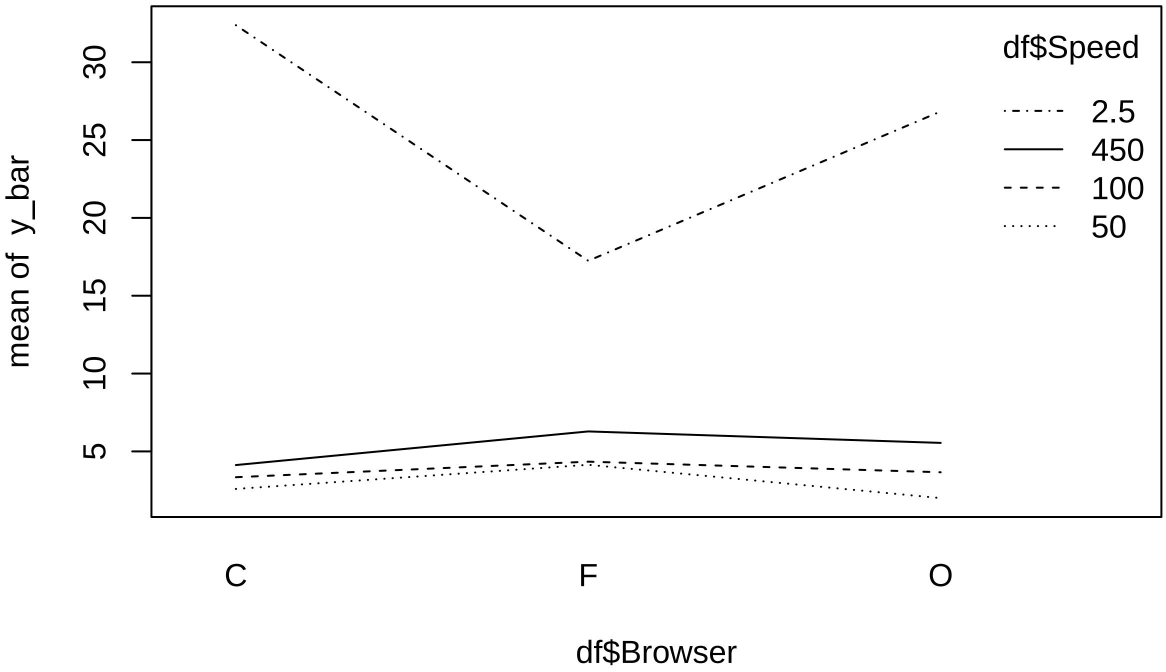
```
# Interaction plots
y_bar = (df$Replicate.1 + df$Replicate.2 + df$Replicate.3)/3
interaction.plot(df$Device, df$Speed, y_bar)
```



```
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))
```