final_project_419

Tyler Chun

11/24/2021

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

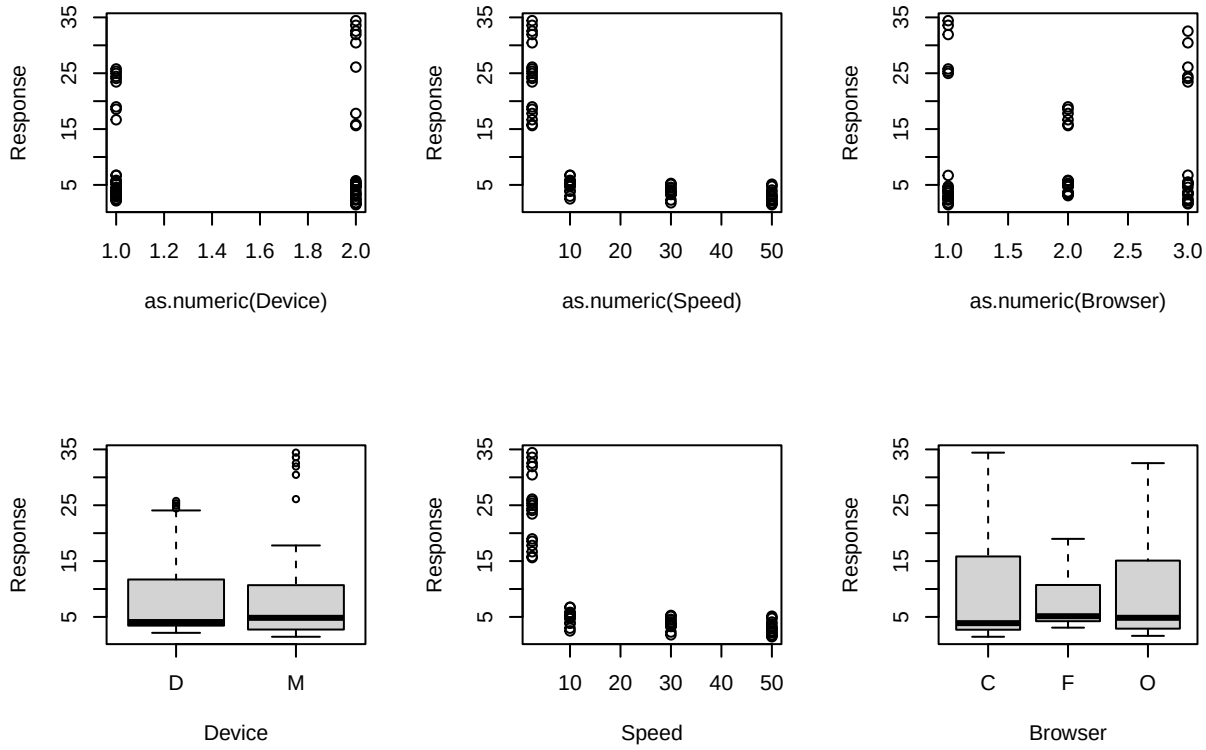
- Treat Speed as numeric
- Transformation of response

```
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))))

new_df <- new_df[-8,]

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)
```

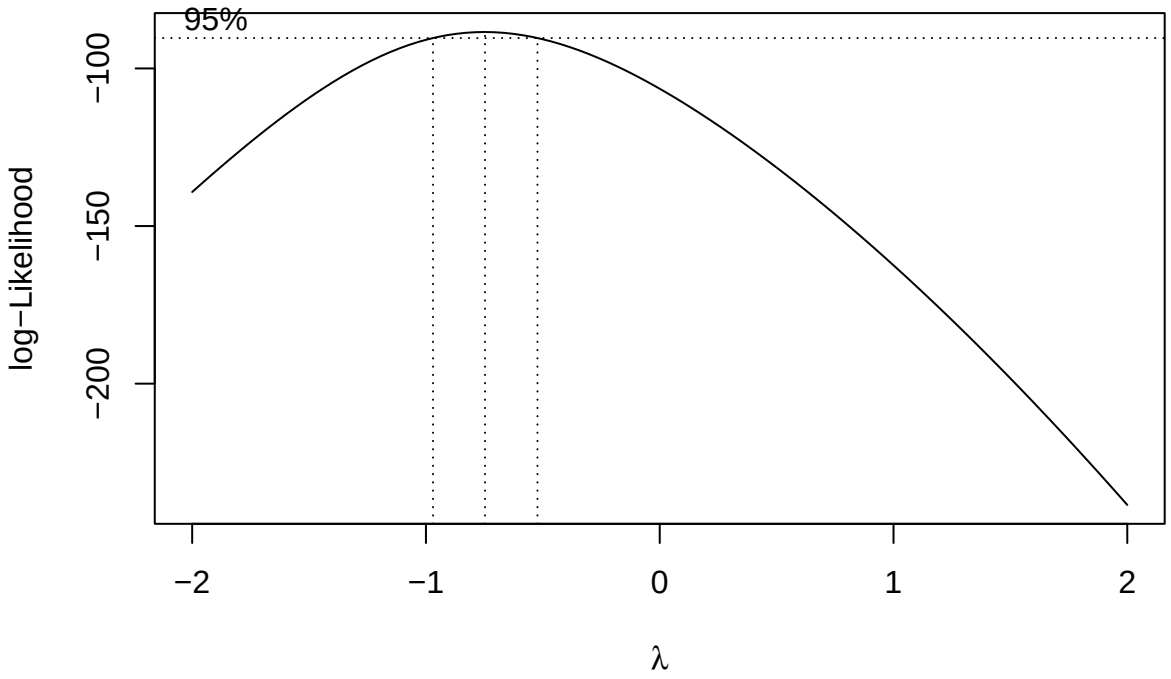
```
##
## Call:
## lm(formula = Response ~ Device + Speed + Browser + Device:Speed +
##     Device:Browser + Speed:Browser + replicate, data = new_df)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.6535  -4.8458   0.1637   4.0344  14.9680
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      18.31812    3.33611   5.491 8.89e-07 ***
## DeviceM           2.13989    3.76272   0.569 0.571713
## Speed            -0.38910    0.09523  -4.086 0.000134 ***
## BrowserF         -5.86815    4.18030  -1.404 0.165631
## BrowserO         -0.70849    4.06674  -0.174 0.862292
## replicate2        0.03600    2.17990   0.017 0.986880
## replicate3        0.09350    2.17990   0.043 0.965933
## DeviceM:Speed    -0.05070    0.09552  -0.531 0.597561
## DeviceM:BrowserF -1.12717    4.35979  -0.259 0.796892
## DeviceM:BrowserO  0.09417    4.30798   0.022 0.982634
## Speed:BrowserF    0.19771    0.11722   1.687 0.096947 .
## Speed:BrowserO    0.01406    0.11651   0.121 0.904350
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.462 on 59 degrees of freedom
## Multiple R-squared:  0.4921, Adjusted R-squared:  0.3974
## F-statistic: 5.197 on 11 and 59 DF,  p-value: 1.177e-05
```

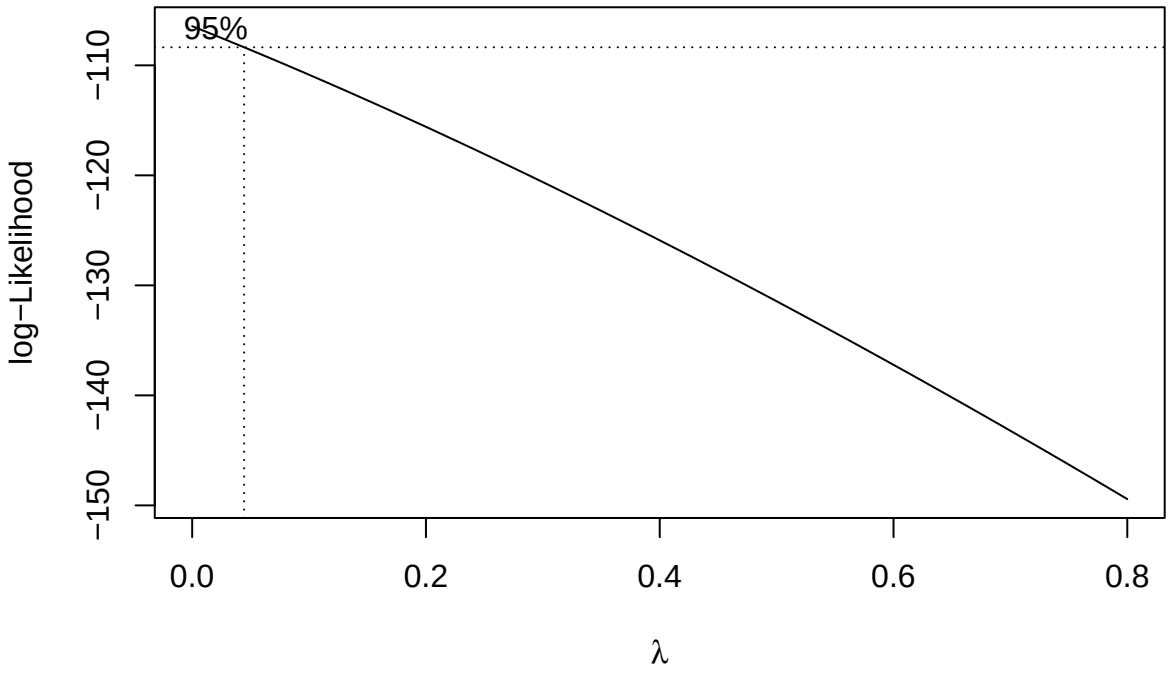
```
anova(m1)
```

```
## Analysis of Variance Table
##
## Response: Response
##                Df Sum Sq Mean Sq F value   Pr(>F)
## Device          1    8.0    8.03  0.1442   0.7055
## Speed           1 2914.7 2914.73 52.3516 1.08e-09 ***
## Browser         2   42.0   21.02  0.3776   0.6872
## replicate       2    0.0    0.02  0.0004   0.9996
## Device:Speed    1   14.8   14.77  0.2653   0.6084
## Device:Browser  2    6.9    3.46  0.0621   0.9398
## Speed:Browser   2  196.5   98.23  1.7643   0.1802
## Residuals      59 3284.9   55.68
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

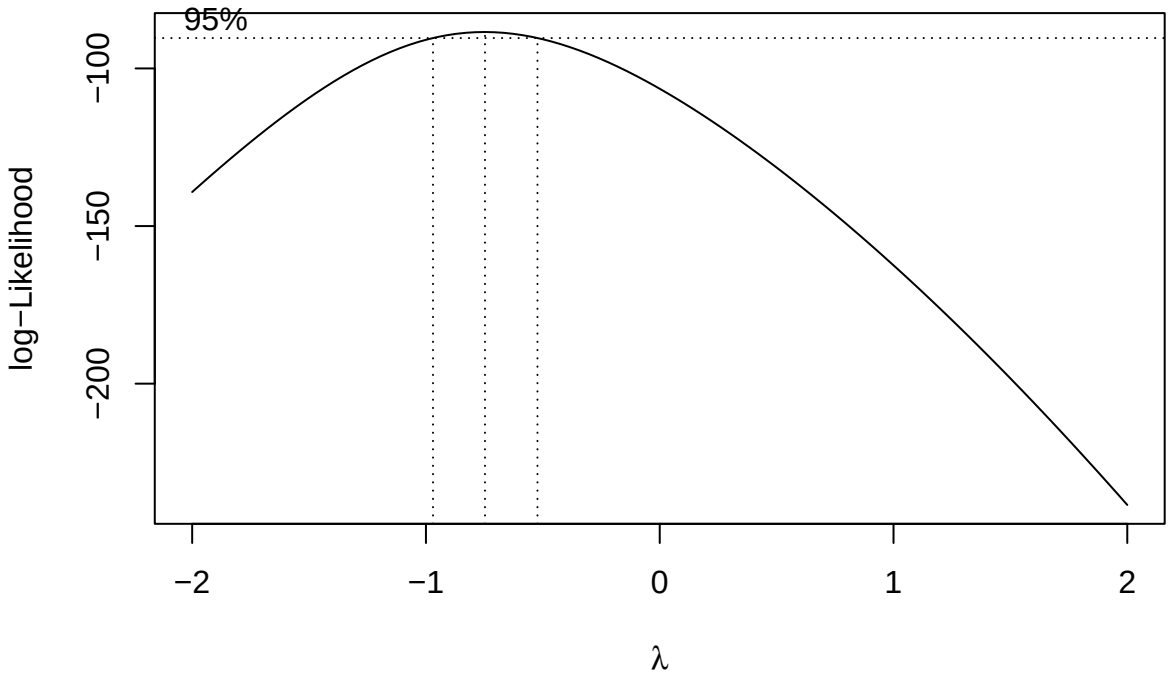
```
library(MASS)


# 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)
```

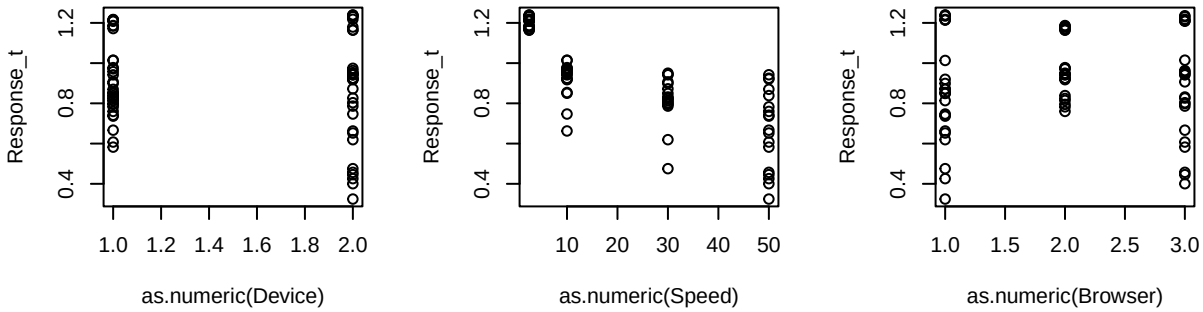


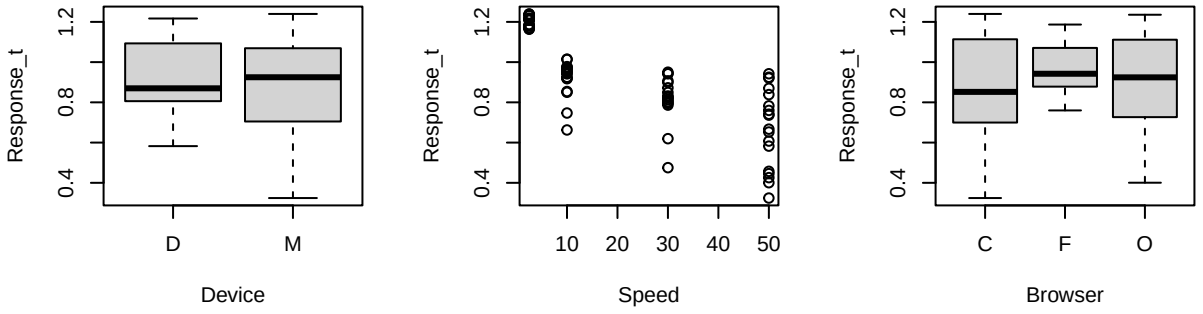

```
lambda <- lmod1_bc$x[which.max(lmod1_bc$y)]
lambda
```

```
## [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 <- round(lambda,2)
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))))

new_df <- new_df[-8,]

# 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)
```

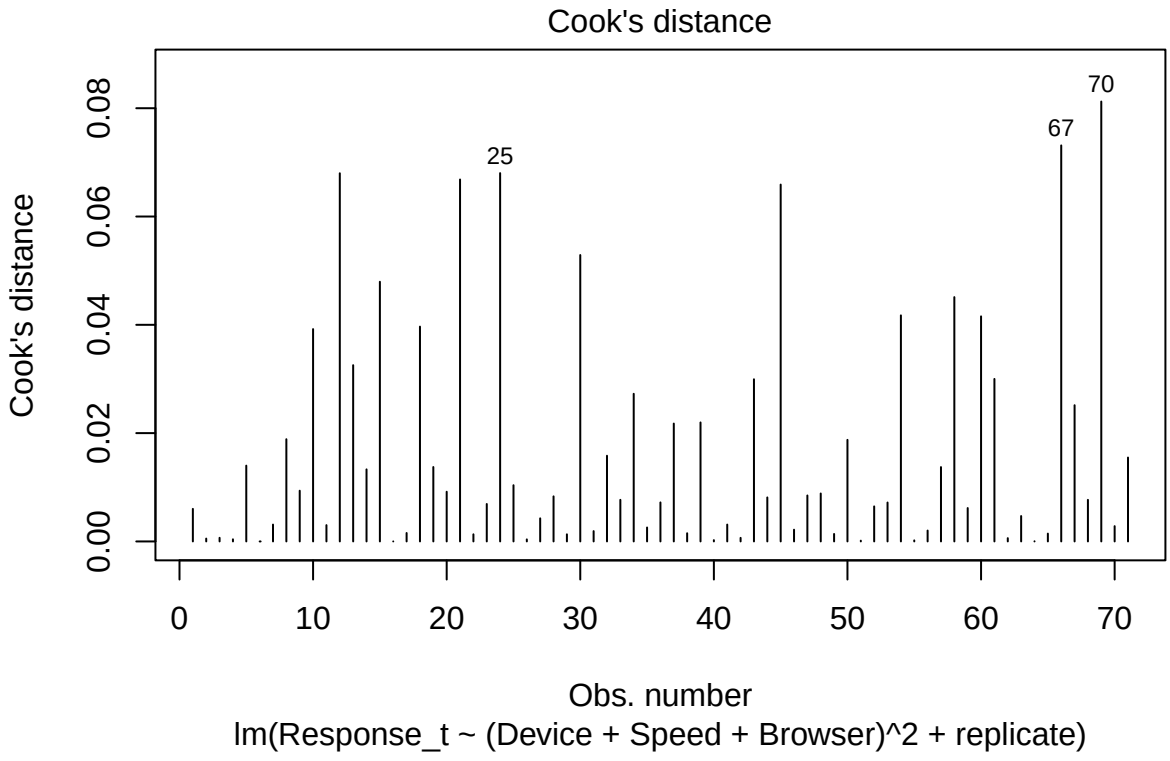
```
##
## Call:
## lm(formula = Response_t ~ (Device + Speed + Browser)^2 + replicate,
##     data = new_df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.260630 -0.075307 -0.000254  0.072551  0.229674
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.161262   0.054020  21.497  < 2e-16 ***
## DeviceM           -0.107019   0.060928  -1.756  0.08420 .
## Speed             -0.009452   0.001542  -6.130 7.86e-08 ***
## BrowserF          -0.109487   0.067690  -1.617  0.11111
## BrowserO           0.033066   0.065851   0.502  0.61745
## replicate2         0.005484   0.035298   0.155  0.87706
## replicate3        -0.015578   0.035298  -0.441  0.66060
## DeviceM:Speed     -0.002081   0.001547  -1.345  0.18365
## DeviceM:BrowserF   0.208216   0.070596   2.949  0.00456 **
## DeviceM:BrowserO   0.098848   0.069757   1.417  0.16173
## Speed:BrowserF     0.004880   0.001898   2.571  0.01268 *
## Speed:BrowserO    -0.002633   0.001887  -1.396  0.16804
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1208 on 59 degrees of freedom
## Multiple R-squared:  0.773,  Adjusted R-squared:  0.7306
## F-statistic: 18.26 on 11 and 59 DF,  p-value: 3.878e-15
```

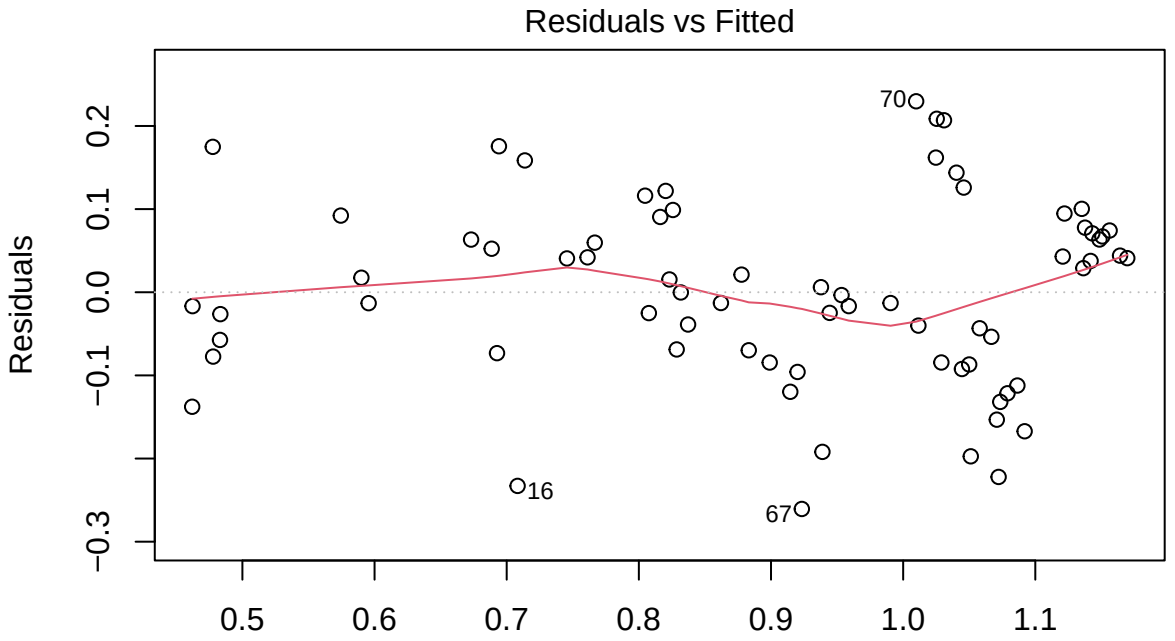
```
anova(m1)
```

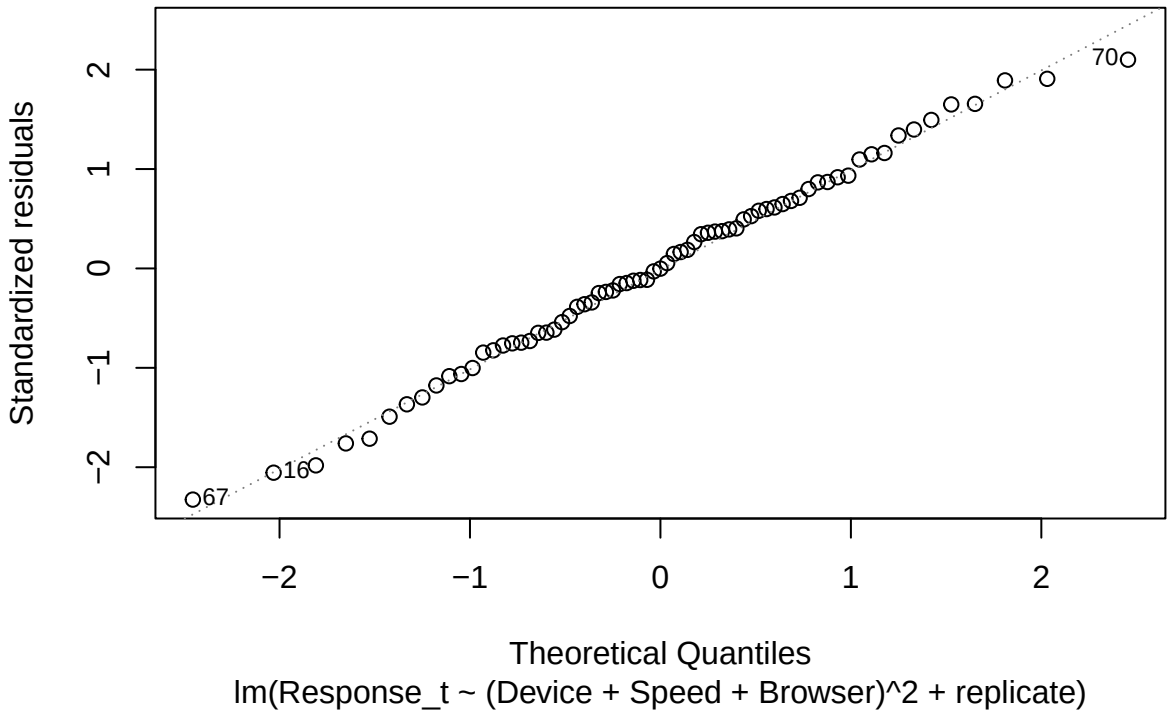
```
## Analysis of Variance Table
##
## Response: Response_t
##                Df  Sum Sq Mean Sq  F value    Pr(>F)
## Device          1 0.04552 0.04552   3.1179  0.082612 .
## Speed           1 2.33365 2.33365 159.8587 < 2.2e-16 ***
## Browser         2 0.16692 0.08346   5.7171  0.005377 **
## replicate       2 0.00706 0.00353   0.2419  0.785946
## Device:Speed    1 0.02331 0.02331   1.5971  0.211286
## Device:Browser  2 0.12081 0.06041   4.1379  0.020810 *
## Speed:Browser   2 0.23505 0.11753   8.0508  0.000810 ***
## Residuals      59 0.86130 0.01460
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

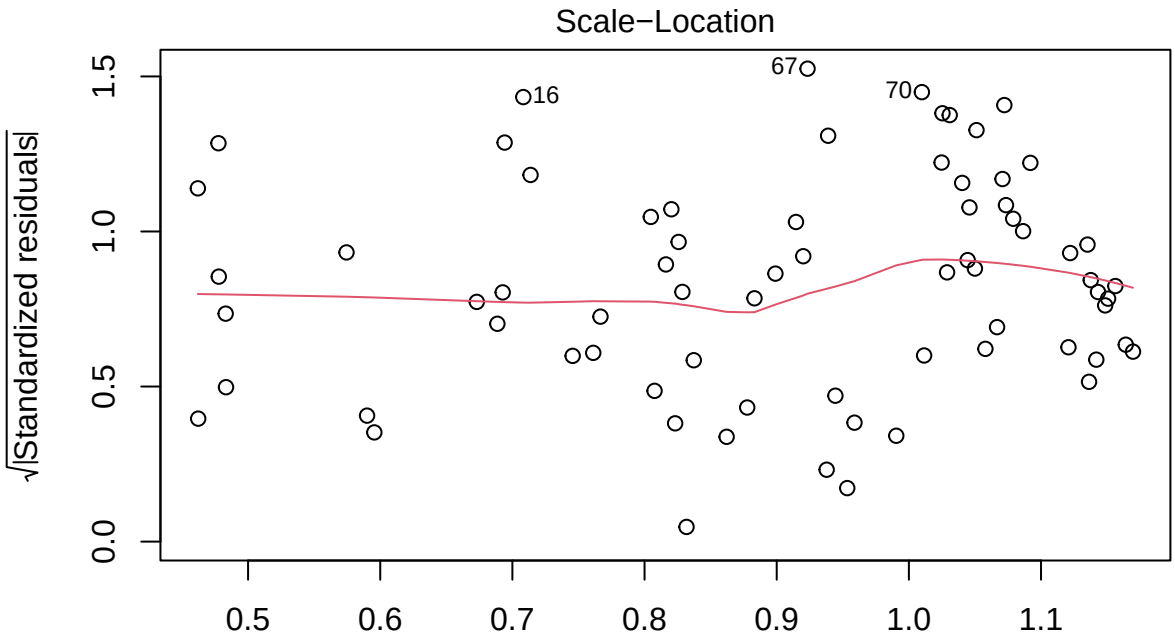
```
# Cook's distance
plot(m1,4)
```

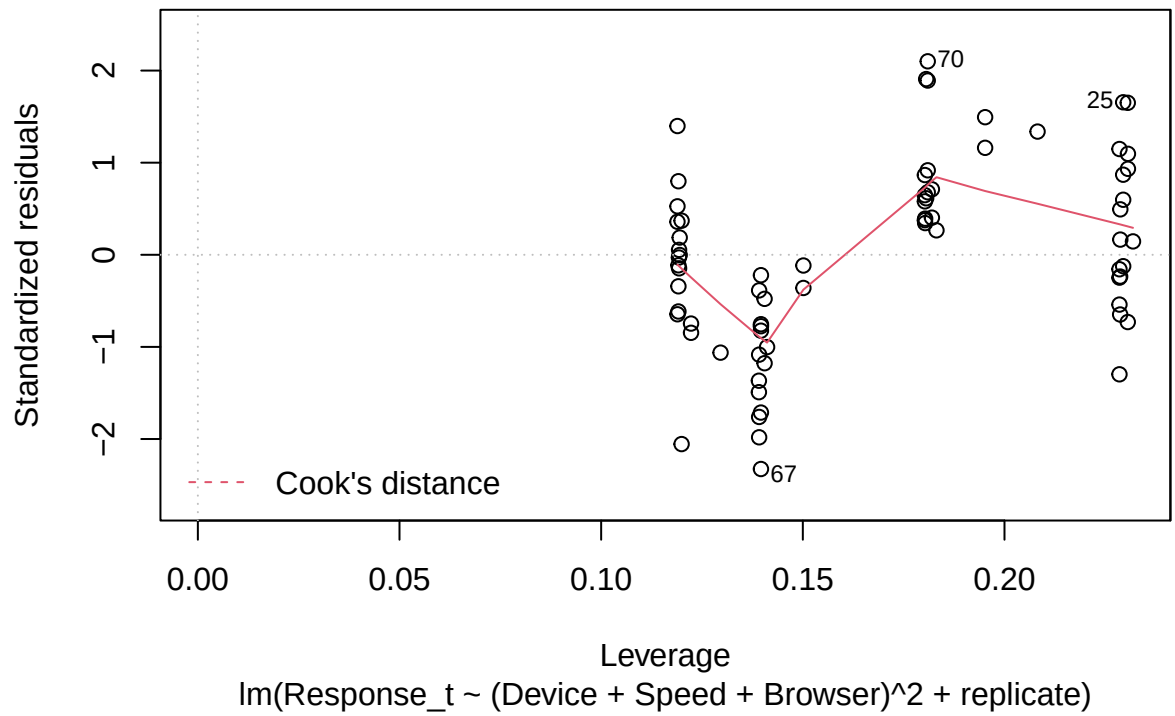



```
# Residual vs Fitted
plot(m1)
```

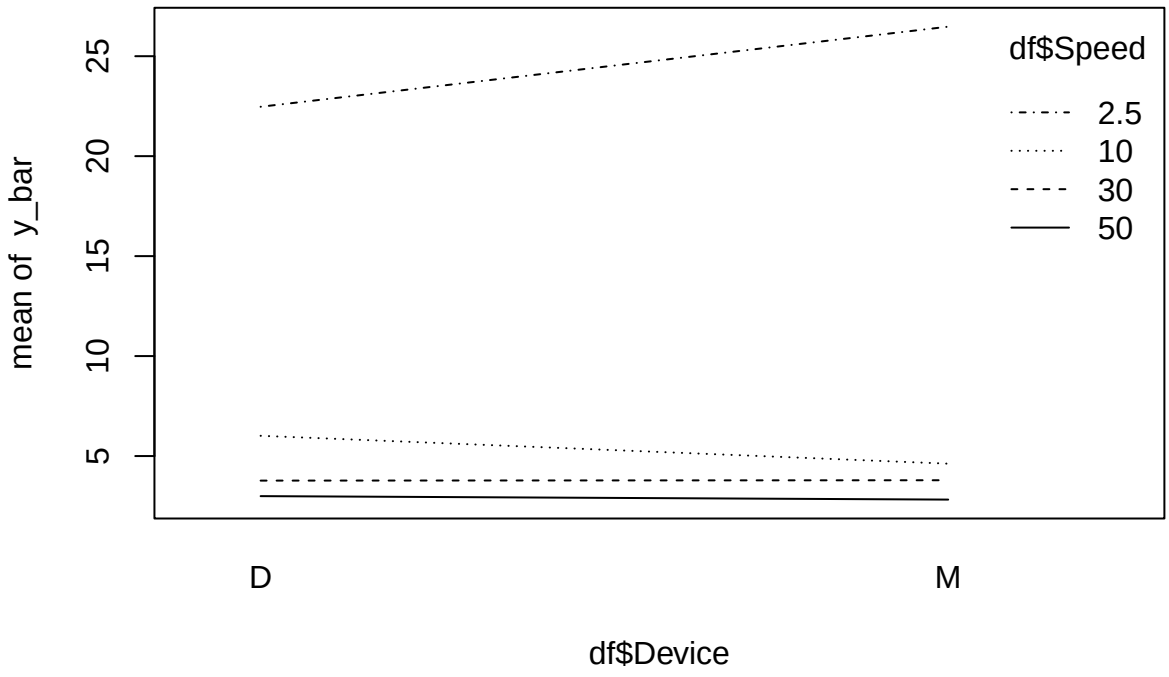

Residuals vs Fitted
Residuals
Fitted values
lm(Response_t ~ (Device + Speed + Browser)^2 + replicate)
70
16
67


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Response_t ~ (Device + Speed + Browser)^2 + replicate)
70
16
67

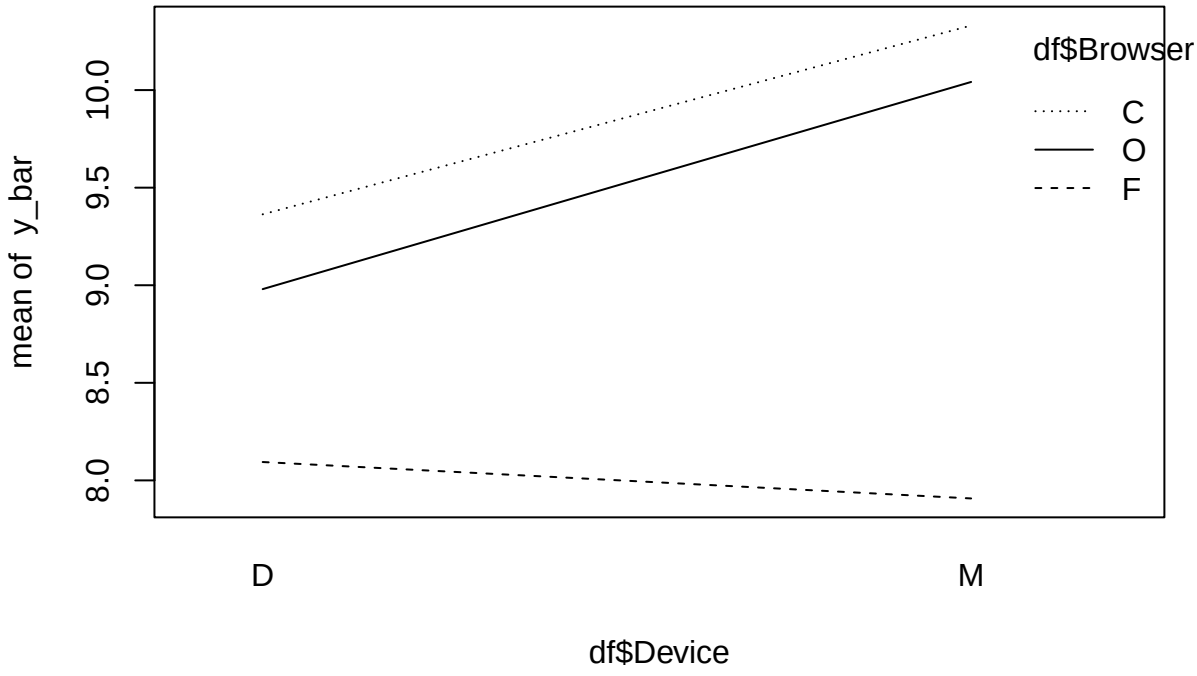




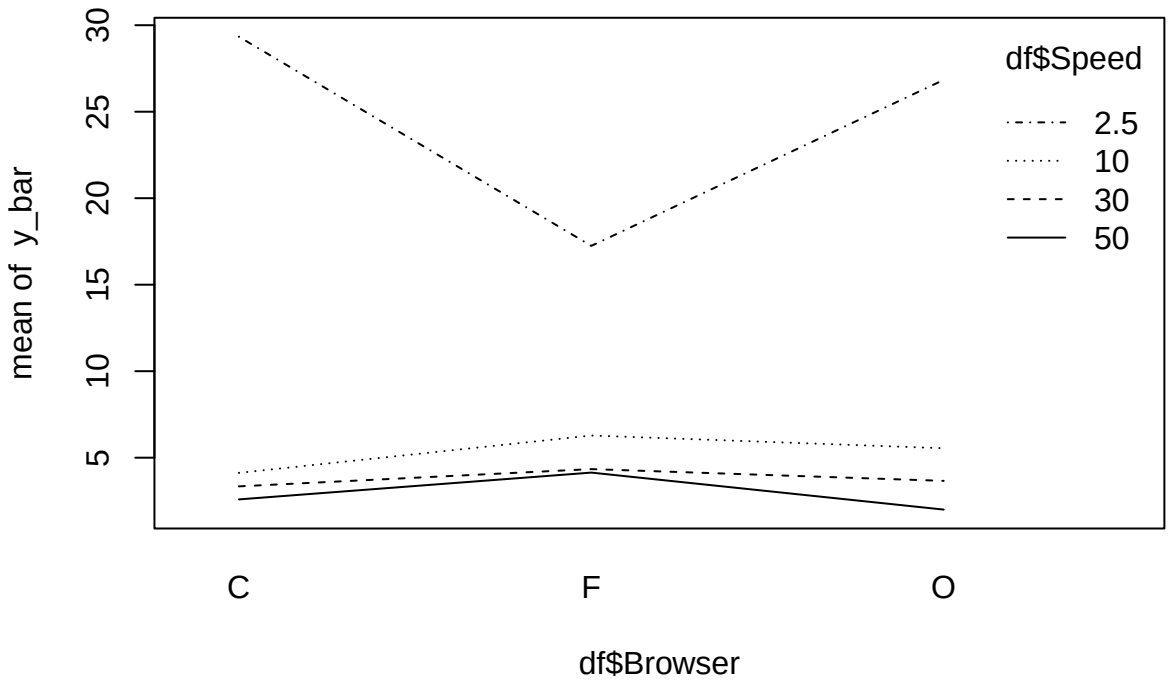

```
# Interaction plots
y_bar = (df$Replicate.1 + df$Replicate.2 + df$Replicate.3)/3
interaction.plot(df$Device, df$Speed, y_bar)
```




```
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))
```