Statistical Analysis using R

2023-09-03

QUESTION 1

Question 1 tests whether measures of vocal entropy (PPE and RPDE) significantly impact a subject's UPDRS score - a measure of Parkinson's disease - even when taking subject age and the effects of passing time into account.

Load libraries and import data:

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.1.3
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(fmsb)
```

```
## Warning: package 'fmsb' was built under R version 4.1.3
```

```
library(ALSM)
```

```
## Warning: package 'ALSM' was built under R version 4.1.3
```

```
## Loading required package: leaps
```

```
## Loading required package: SuppDists
```

```r
library(onewaytests)
```

```
## Warning: package 'onewaytests' was built under R version 4.1.3
```

```r
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.3
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.1.3
```

```r
library(fmsb)
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```r
library(leaps)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:boot':
##
##     melanoma
```

```
## The following object is masked from 'package:ALSM':
##
##     oneway
```

```
## Registered S3 methods overwritten by 'pROC':
##   method    from
##   print.roc fmsb
##   plot.roc  fmsb
```

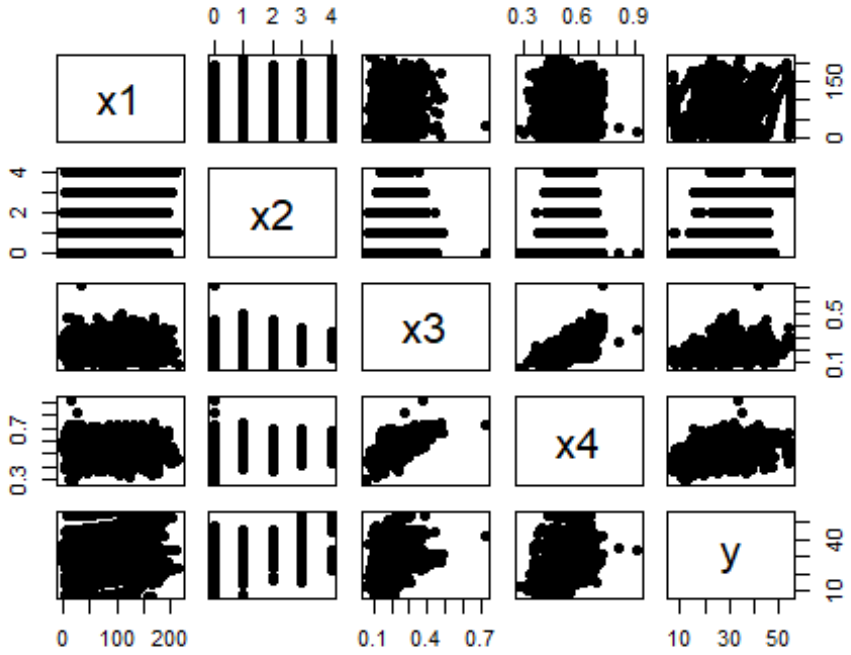
```
#import data
data = read.csv('C:/Users/Ankita Mishra/Desktop/parkinsons_data.csv')
```

Modify Data: Each subject was tested several times on a single day. To create a variable that accurately assesses the effect of passing time, we average the results that occur on the same day. We also turn the subject age into a categorical variable with 5 groups.

```
#Average the test times into days for each subject
data_group = data
data_group$test_time = as.integer(data_group$test_time)
data_group <- data_group %>%
  group_by(subject.,age,sex,test_time) %>%
  dplyr::summarise_all('mean') %>%
  as.data.frame()

# Group Age into a categorical variable
ageCut = cut(data_group$age, breaks=c(1,60,65,70,75,100))
ageCategory = factor(ageCut,
levels=c('(1,60]','(60,65]','(65,70]','(70,75]','(75,100]'),
labels=c(0,1,2,3,4))
ageCategory = as.integer(as.character(ageCategory))
```

Assign variables and visualize: This is our first glimpse at the data. Raw scatterplots indicate weak correlations of each variable with Y. The measures of vocal entropy (X3-PPE and X4-RPDE) are unsurprisingly trending together. Multicollinearity issues are addressed formally later. We also see potential influential points here.

```
y = data_group$total_UPDRS # Response variable, UPDRS diagnostic score,
continuous
x1 = data_group$test_time # Predictor variable, day of data collection,
integers
x2 = ageCategory # Predictor variable, age group of subject, categorical
x3 = data_group$PPE # Predictor variable, pitch period entropy, continuous
x4 = data_group$RPDE # Predictor variable, recurrence period density entropy,
continuous
# Create scatterplot matrix
frame = data.frame(x1, x2, x3, x4, y)

pairs(frame[,1:5], pch=19)
```



Initial MLR Creation: Insights from the initial MLR models show that all predictors contribute significant - if small - influence on UPDRS score. Examining X3-PPE, we find its sum of squares contribution largely varies between the Type I and Type II ANOVA tables, indicating potential multicollinearity issues. We also see that the coefficient of determination for this model will likely be around 20%.

```
model = lm(y~x1+x2+x3+x4)
anova(model)

## Analysis of Variance Table
##
## Response: y
##            Df Sum Sq Mean Sq  F value    Pr(>F)
## x1          1    661   660.5   7.1544  0.007602 **
## x2          1  18852 18852.0 204.1893 < 2.2e-16 ***
## x3          1   2171  2171.2  23.5167 1.438e-06 ***
## x4          1    631   631.3   6.8377  0.009062 **
## Residuals 985  90941    92.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Anova(model, type="II")

## Anova Table (Type II tests)
##
## Response: y
##           Sum Sq  Df  F value    Pr(>F)
## x1           683   1   7.4006  0.006635 **
```

```
## x2         16011   1 173.4196 < 2.2e-16 ***
## x3           430   1   4.6574  0.031160 *
## x4           631   1   6.8377  0.009062 **
## Residuals  90941 985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
m= summary(model)
m
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.738  -7.562  -1.358   7.557  24.808
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.67305    2.16371   6.319 3.97e-10 ***
## x1           0.01553    0.00571   2.720  0.00664 **
## x2           2.97579    0.22597  13.169  < 2e-16 ***
## x3          11.54379    5.34903   2.158  0.03116 *
## x4          12.45306    4.76234   2.615  0.00906 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.609 on 985 degrees of freedom
## Multiple R-squared:  0.197,  Adjusted R-squared:  0.1938
## F-statistic: 60.42 on 4 and 985 DF,  p-value: < 2.2e-16
```

```
b0 = m$coefficients[1,1] # Intercept
b1 = m$coefficients[2,1] # Impact of passing time
b2 = m$coefficients[3,1] # Impact of subject age
b3 = m$coefficients[4,1] # Impact of PPE
b4 = m$coefficients[5,1] # Impact of RPDE
yhat = b0 + b1*x1 + b2*x2 + b3*x3 + b4*x4
res = residuals(model)
n = length(y)
```

Diagnostics - Multicollinearity: Here we formally test for multicollinearity with the variance inflation factor. The full model has a VIF of 1.004, which does not come near our threshold of 10. However, because we suspect X3 and X4, we experiment with other models and find that the VIF between them is only 1.664. Since this is far less than 10, we can proceed with reasonable assurance. Ridge regression is not necessary. We also tried including the interaction X3*X4, but this worsened the R^2 of the model.

```
# Empirically test for multicollinearity. Problems if VIF > 10.
VIF(lm(x1~x2+x3+x4))
```

```
## [1] 1.003972
```

```
VIF(lm(x2~x1+x3+x4))
```

```
## [1] 1.029038
```

```
VIF(lm(x3~x2+x1+x4))
```

```
## [1] 1.678419
```

```
VIF(lm(x4~x2+x3+x1))
```

```
## [1] 1.677207
```

```
VIF(lm(x3~x4))
```

```
## [1] 1.664426
```

```
summary(lm(y~x3))
```

```
##
## Call:
## lm(formula = y ~ x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.270  -7.473  -1.492   7.957  25.242
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   22.574      1.045  21.593   <2e-16 ***
## x3            28.976      4.507   6.429    2e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.49 on 988 degrees of freedom
## Multiple R-squared:  0.04015,    Adjusted R-squared:  0.03918
## F-statistic: 41.33 on 1 and 988 DF,  p-value: 1.999e-10
```

```
summary(lm(y~x4))
```

```
##
## Call:
## lm(formula = y ~ x4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.712  -7.732  -1.449   7.369  28.336
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   15.072      2.200   6.851 1.29e-11 ***
```

```
## x4            25.616      4.016   6.379 2.73e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.49 on 988 degrees of freedom
## Multiple R-squared:  0.03956,    Adjusted R-squared:  0.03859
## F-statistic: 40.69 on 1 and 988 DF,  p-value: 2.732e-10
```

```
summary(lm(y~x3+x4))
```

```
##
## Call:
## lm(formula = y ~ x3 + x4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.369  -7.774  -1.596   7.983  26.942
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   16.599      2.245   7.394 3.04e-13 ***
## x3            17.983      5.792   3.105  0.00196 **
## x4            15.497      5.158   3.004  0.00273 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.45 on 987 degrees of freedom
## Multiple R-squared:  0.04885,    Adjusted R-squared:  0.04692
## F-statistic: 25.35 on 2 and 987 DF,  p-value: 1.845e-11
```

Diagnostics - Outliers / Influential Points: Using the criteria for DFFITS, DFBETAS, and Cook's Distance, no influential points were identified. Inspecting the diagonals of the Hat matrix revealed several X-outliers. These are removed from the model in the next step, and all diagnosis is performed with the new data. VIF analysis was re-run without the outliers, and the results barely changed at all. The residual plots do not reveal any obvious non-constant variance.

```
print("Searching for points influencing the model fit...")
```

```
## [1] "Searching for points influencing the model fit..."
```

```
for (p in dffits(model)) {
  if (p>1) {
    print(p)
  }
}
print("Searching for points influencing the impact parameters...")
```

```
## [1] "Searching for points influencing the impact parameters..."
```

```r
for (p in dfbetas(model)) {
  if (p>1) {
    print(p)
  }
}
print("Searching for Cook's Distance influential points...")
```

```
## [1] "Searching for Cook's Distance influential points..."
```

```r
CD = cooks.distance(model)
print(cat("Maximum Cook's Distance = ", max(CD)))
```

```
## Maximum Cook's Distance =  0.01560741NULL
```

```r
minor = qf(0.2, 5, n-5)
moderate = qf(0.5, 5, n-5)
print("Searching for X-Outliers using the Hat matrix")
```

```
## [1] "Searching for X-Outliers using the Hat matrix"
```

```r
print("Significance threshold is")
```

```
## [1] "Significance threshold is"
```

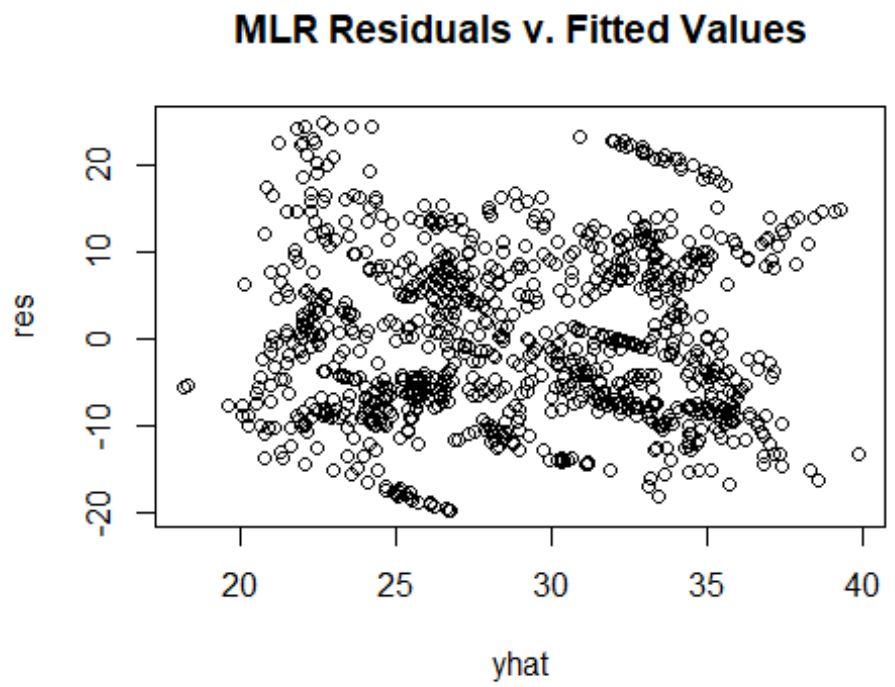
```r
3*(5/n) #Using a significance threshold of 3*(p/n) instead of 2*(p/n) to be
extra conservative with retaining
```

```
## [1] 0.01515152
```

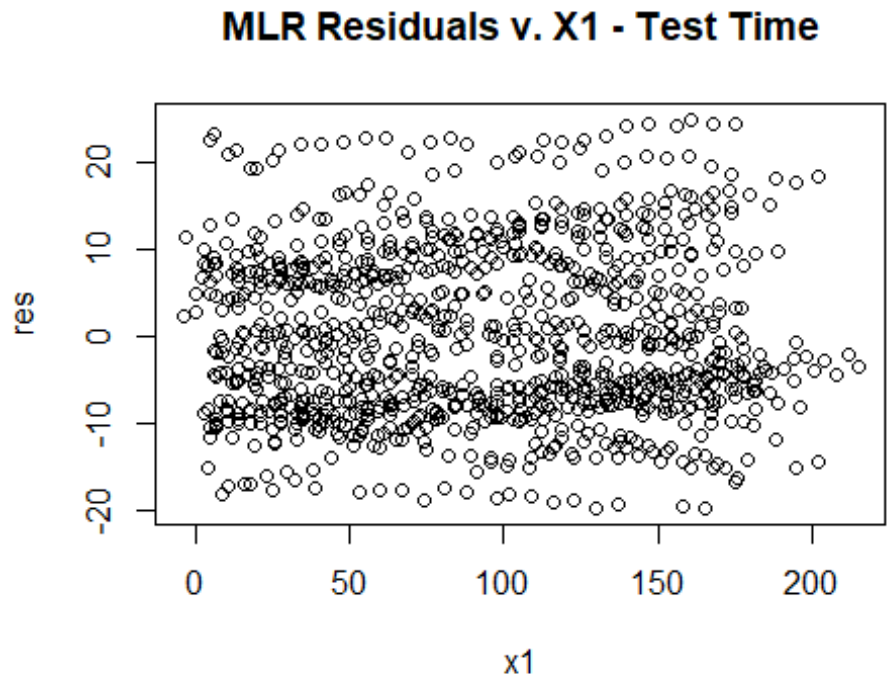
```r
hat = lm.influence(model)$hat # Computing the Hat matrix
for (p in hat) {
  if (abs(p)>(3*(5/n))) {
    print(which(hat == p))
  }
}
```

```
## 685
## 685
## 819
## 819
## 833
## 833
## 840
## 840
## 842
## 842
## 844
## 844
```

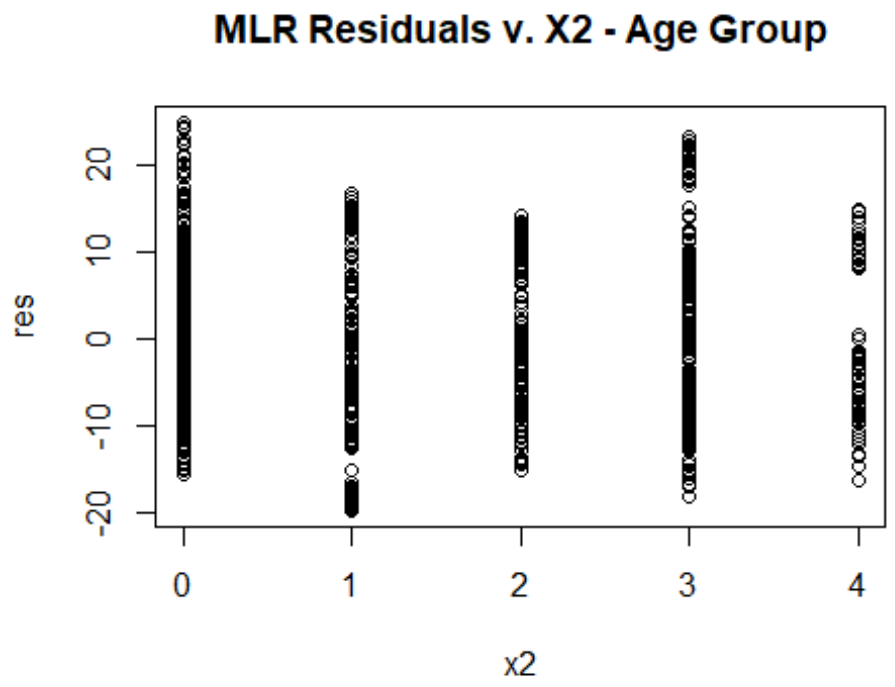
```r
# Residual Inspection
plot(yhat, res, main = "MLR Residuals v. Fitted Values")
```



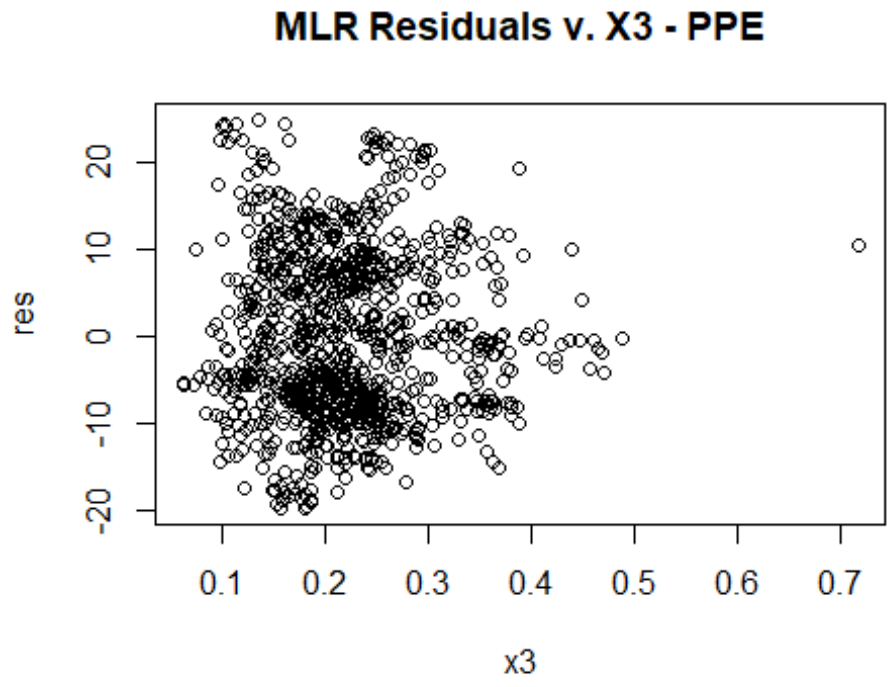

```
plot(x1, res, main = "MLR Residuals v. X1 - Test Time")
```



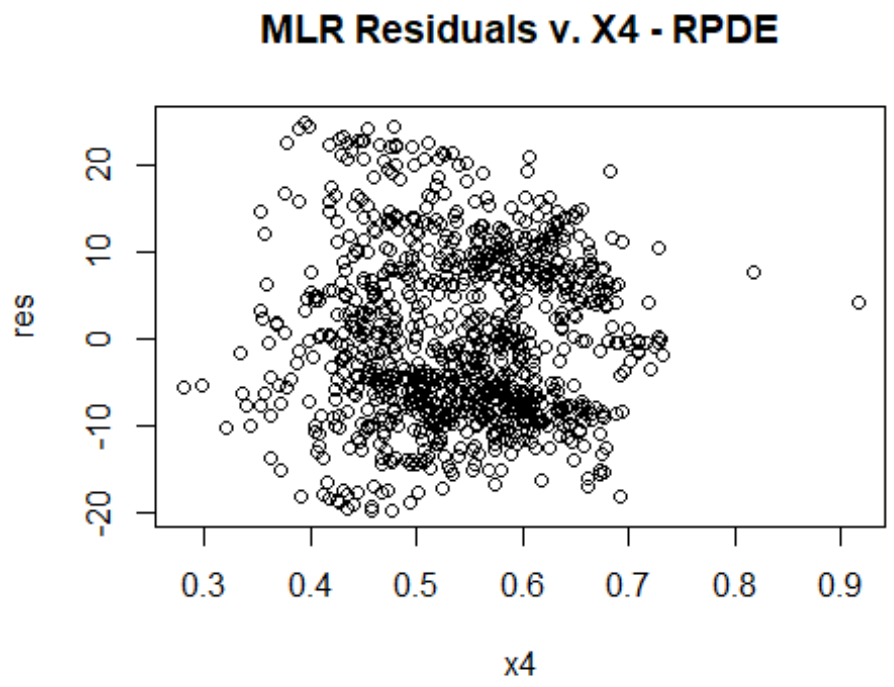

```
plot(x2, res, main = "MLR Residuals v. X2 - Age Group")
```




```
plot(x3, res, main = "MLR Residuals v. X3 - PPE")
```




```
plot(x4, res, main = "MLR Residuals v. X4 - RPDE")
```



Remake Model without Outliers: Following the exact same steps as Initial Model Creation, but data without outliers are stored in dataframe "outliers_gone". Interestingly, removing the outliers causes the added value of X3 to become insignificant.

```
# Remake Model without Outliers:
outliers = c(685, 819, 833, 840, 842, 844) # Indicies of outliers, discovered
in previous step
outliers_gone <- data_group[-outliers, ]
y = outliers_gone$total_UPDRS
x1 = outliers_gone$test_time
ageCut = cut(outliers_gone$age, breaks=c(1,60,65,70,75,100))
ageCategory = factor(ageCut,
levels=c('(1,60]','(60,65]','(65,70]','(70,75]','(75,100]'),
labels=c(0,1,2,3,4))
ageCategory = as.integer(as.character(ageCategory))
x2 = ageCategory
x3 = outliers_gone$PPE
x4 = outliers_gone$RPDE
model = lm(y~x1+x2+x3+x4)
anova(model)

## Analysis of Variance Table
##
## Response: y
##            Df Sum Sq Mean Sq F value    Pr(>F)
## x1          1    700   699.8   7.551  0.006108 **
## x2          1  19209 19208.6 207.268 < 2.2e-16 ***
```

```
## x3          1   1791  1791.2  19.327 1.222e-05 ***
## x4          1    573   573.5   6.188  0.013027 *
## Residuals 979  90729    92.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(model, type="II")
```

```
## Anova Table (Type II tests)
##
## Response: y
##           Sum Sq  Df  F value    Pr(>F)
## x1           714   1   7.7095  0.005598 **
## x2         16096   1 173.6836 < 2.2e-16 ***
## x3           342   1   3.6927  0.054941 .
## x4           573   1   6.1880  0.013027 *
## Residuals  90729 979
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
m= summary(model)
m
```

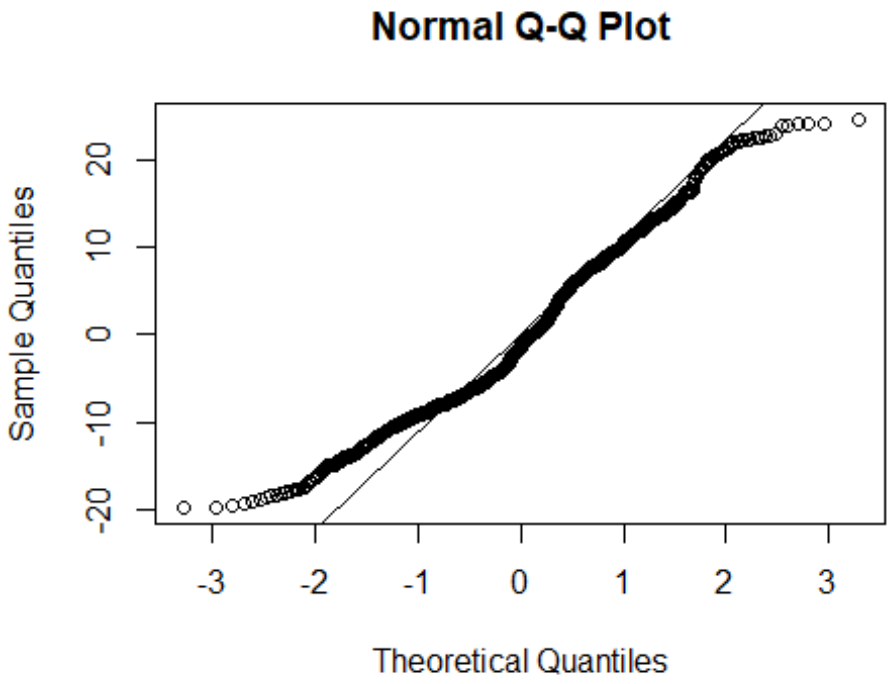
```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.814  -7.558  -1.373   7.597  24.722
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.932208   2.198871   6.336 3.59e-10 ***
## x1           0.015955   0.005746   2.777   0.0056 **
## x2           3.002871   0.227854  13.179  < 2e-16 ***
## x3          10.754516   5.596538   1.922   0.0549 .
## x4          12.113075   4.869427   2.488   0.0130 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.627 on 979 degrees of freedom
## Multiple R-squared:  0.1971, Adjusted R-squared:  0.1938
## F-statistic: 60.08 on 4 and 979 DF,  p-value: < 2.2e-16
```

```
b0 = m$coefficients[1,1]
b1 = m$coefficients[2,1]
b2 = m$coefficients[3,1]
b3 = m$coefficients[4,1]
b4 = m$coefficients[5,1]
yhat = b0 + b1*x1 + b2*x2 + b3*x3 + b4*x4
```

```
res = residuals(model)
n = length(y)
```

Diagnostics - Heteroscedacity and Non-normality: Heteroscedacity is tested with the Brown-Forsythe test, with 3 equal sections. Non-normality is tested with the Shapiro test. The model failed both test, but the most drastic failure was with non-normality, as easily viewed from the Q-Q plot. The BF result is seemingly in conflict with the residual plots, which are not obviously hereoscedastic. However, from these results we decided to apply remediation to correct the assumption violations.

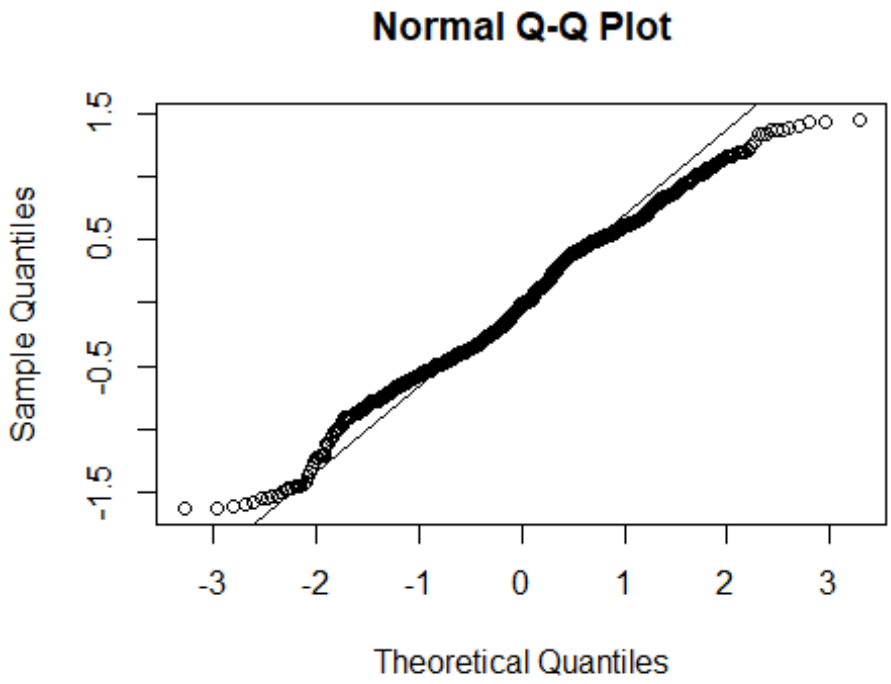
```
#Diagnostics
# Brown-Forsythe Test
g<-rep(1,n)
g[y<=(max(y)/3)]=0 # Divide Y data into 3 equal sections
g[y>=(2*(max(y)/3))]=2
bftest(model, g, alpha=0.05)

##       t.value     P.Value alpha  df
## [1,] 3.848719 0.000126411  0.05 982

# Shapiro Test
shapiro.test(res)

##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.97293, p-value = 1.42e-12

qqnorm(res)
qqline(res)
```



Remediation - Y Transformation: The Box Cox method was used to find an appropriate lambda for transforming Y. The boxcox R function was not working, so this was performed manually by trial and error until a lambda was selected that minimized the non-normality. As a result of the transformation, the p-value on the Shapiro test improved by 6 orders of magnitude. At the same time, the p-value on the BF test drastically worsened. However, the fitted value residual plot still looks fine, so we proceed with this transformation having improved the normality of the data.

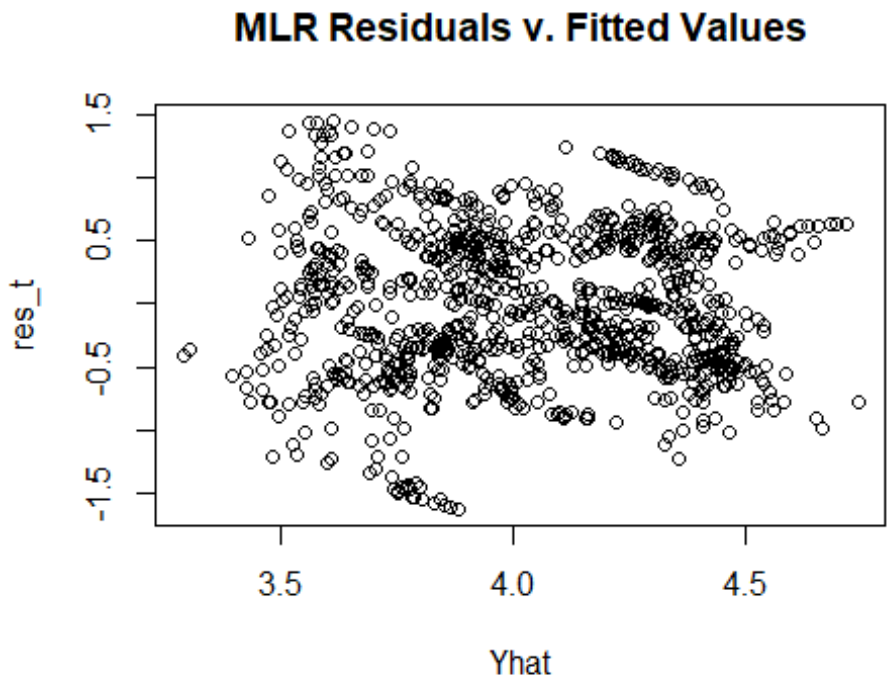
```
# Remediation: Y Transformation
Y = y**0.42 # Transforming UPDRS score with lambda = 0.42
model_t = lm(Y~x1+x2+x3+x4) # Model creation with transformed data
# Brown-Forsythe Test
g<-rep(1,n)
g[Y<=(max(Y)/3)]=0 # Divide Y data into 3 equal sections
g[Y>=(2*(max(Y)/3))]=2
bftest(model_t, g, alpha=0.05)

##       t.value      P.Value alpha  df
## [1,] 7.082151 2.702949e-12  0.05 982

# Shapiro Test
res_t = residuals(model_t)
shapiro.test(res_t)

##
##  Shapiro-Wilk normality test
##
```

```
## data:  res_t
## W = 0.98914, p-value = 1.15e-06

qqnorm(res_t)
qqline(res_t)
```




```
# Plot new residuals
m_t = summary(model_t)
m_t

##
## Call:
## lm(formula = Y ~ x1 + x2 + x3 + x4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.62076 -0.44214 -0.01405  0.47274  1.44593
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.926373   0.135828  21.545  < 2e-16 ***
## x1          0.001064   0.000355   2.997 0.002795 **
## x2          0.184299   0.014075  13.094  < 2e-16 ***
## x3          0.618544   0.345708   1.789 0.073890 .
## x4          1.093505   0.300793   3.635 0.000292 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.5947 on 979 degrees of freedom
## Multiple R-squared:  0.2091, Adjusted R-squared:  0.2059
## F-statistic: 64.72 on 4 and 979 DF,  p-value: < 2.2e-16

b0t = m_t$coefficients[1,1]
b1t = m_t$coefficients[2,1]
b2t = m_t$coefficients[3,1]
b3t = m_t$coefficients[4,1]
b4t = m_t$coefficients[5,1]
Yhat = b0t + b1t*x1 + b2t*x2 + b3t*x3 + b4t*x4
plot(Yhat, res_t, main = "MLR Residuals v. Fitted Values")
```



Remediation - Weighted Least Squares: As a last attempt to improve the heteroscedacity, we applied weighted least squares regression. This did not improve the result on the BF test, so WLS was abandoned. Multiple iterations of WLS only worsened the results.

```
# Building models with weighted residuals
wts1=1/fitted(lm(abs(res_t)~x1+x2+x3+x4,data=data))**2
model_w1 = lm(Y~x1+x2+x3+x4, weight = wts1, data) # Model with 1 iteration of
WLS
summary(model_w1)

##
## Call:
## lm(formula = Y ~ x1 + x2 + x3 + x4, data = data, weights = wts1)
##
## Weighted Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.98976 -0.92333 -0.03974  0.96415  2.55159
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.0009363  0.1376879  21.795  < 2e-16 ***
## x1          0.0008249  0.0003378   2.442  0.01479 *
## x2          0.1814592  0.0134057  13.536  < 2e-16 ***
## x3          0.6693033  0.2996697   2.233  0.02574 *
## x4          0.9849193  0.2915915   3.378  0.00076 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.184 on 979 degrees of freedom
## Multiple R-squared:  0.2035, Adjusted R-squared:  0.2002
## F-statistic: 62.52 on 4 and 979 DF,  p-value: < 2.2e-16
```

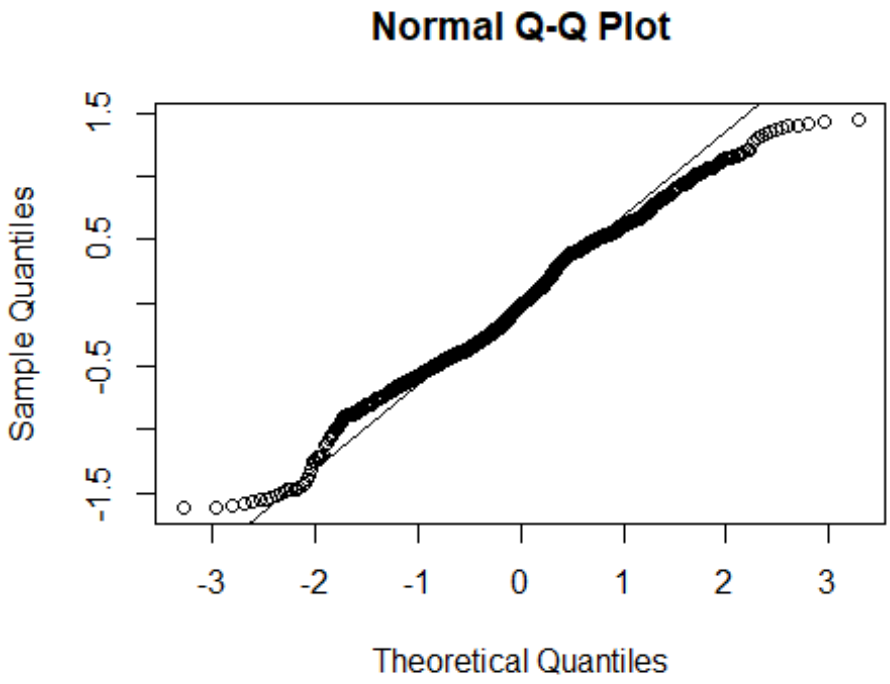
```r
# Brown-Forsythe Test
g<-rep(1,n)
g[Y<=(max(Y)/3)]=0 # Divide Y data into 3 equal sections
g[Y>=(2*(max(Y)/3))]=2
bftest(model_w1, g, alpha=0.05)
```

```
##       t.value      P.Value alpha  df
## [1,] 6.998348 4.791723e-12  0.05 982
```

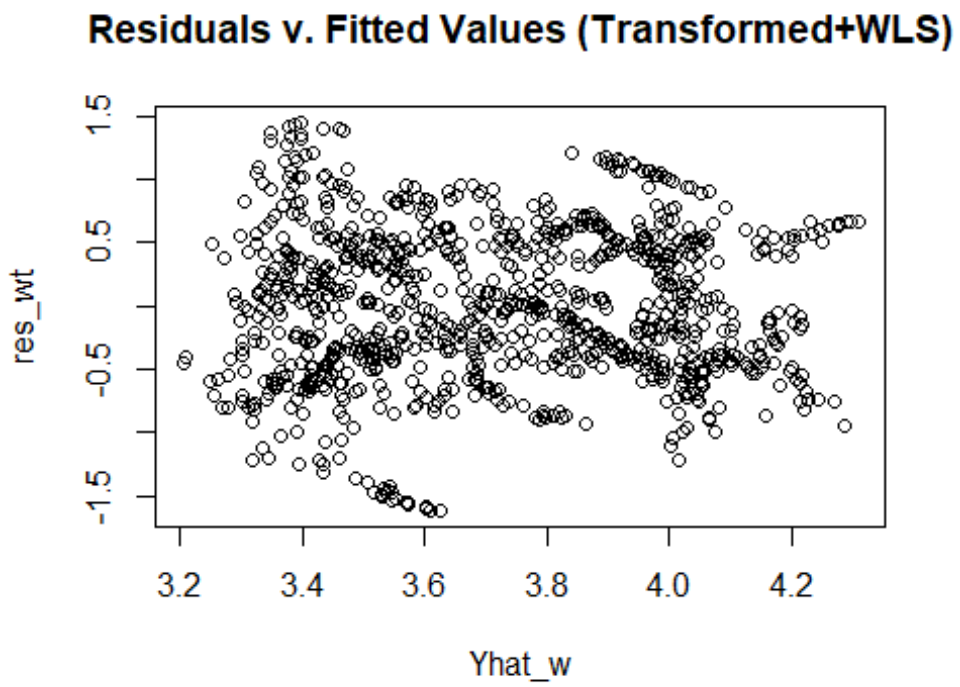
```r
# Shapiro Test
res_wt = residuals(model_w1)
shapiro.test(res_wt)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_wt
## W = 0.98936, p-value = 1.476e-06
```

```r
qqnorm(res_wt)
qqline(res_wt)
```




```
m_w = summary(model_w1)
b0wt = m_w$coefficients[1,1]
b1wt = m_w$coefficients[2,1]
b2wt = m_w$coefficients[3,1]
b3wt = m_w$coefficients[4,1]
Yhat_w = b0wt + b1wt*x1 + b2wt*x2 + b3wt*x4
#res_weighted = residuals(model_w2)
#res_weighted2 = residuals(model_w3)
plot(Yhat_w, res_wt, main = "Residuals v. Fitted Values (Transformed+WLS)")
```



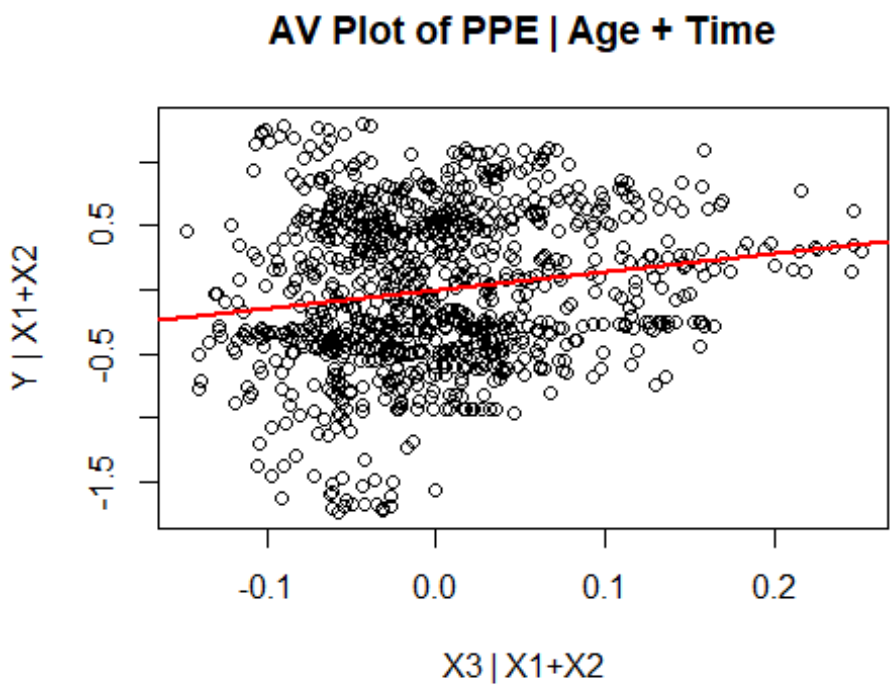
Partial Regression:

The research questions asks about the effects of vocal entropy on UPDRS score even when subject age and time are accounted for in the model. This calls for a comparison of the partial regression coefficients of X3 and X4, given X1 and X2. Here, we create and view the partial regression plots which do show a slight upward trend for both X3 and X4. We also perform an ANOVA F-test comparing a reduced model to the full model containing the partial impacts of X3 and X4. The resulting p-value is 3.7x10^-9. This provides an answer to the research question, that PPE and RPDE together have a significant impact on UPDRS score, regardless of subject age or time of data collection.

```
# Creating plots of X3 | X1,X2 and X4 | X1,X2
m1 <- lm(Y~x1+x2+x3)
rY <- residuals(lm(Y~x1+x2))
rX3 <- residuals(lm(x3~x1+x2))
m2 <- lm(rY~rX3)
summary(m2)

##
## Call:
## lm(formula = rY ~ rX3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.68376 -0.42942 -0.04423  0.49647  1.40696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.599e-17  1.906e-02   0.000        1
```

```
## rX3         1.395e+00  2.733e-01   5.103 4.02e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5977 on 982 degrees of freedom
## Multiple R-squared:  0.02583,    Adjusted R-squared:  0.02484
## F-statistic: 26.04 on 1 and 982 DF,  p-value: 4.019e-07
```

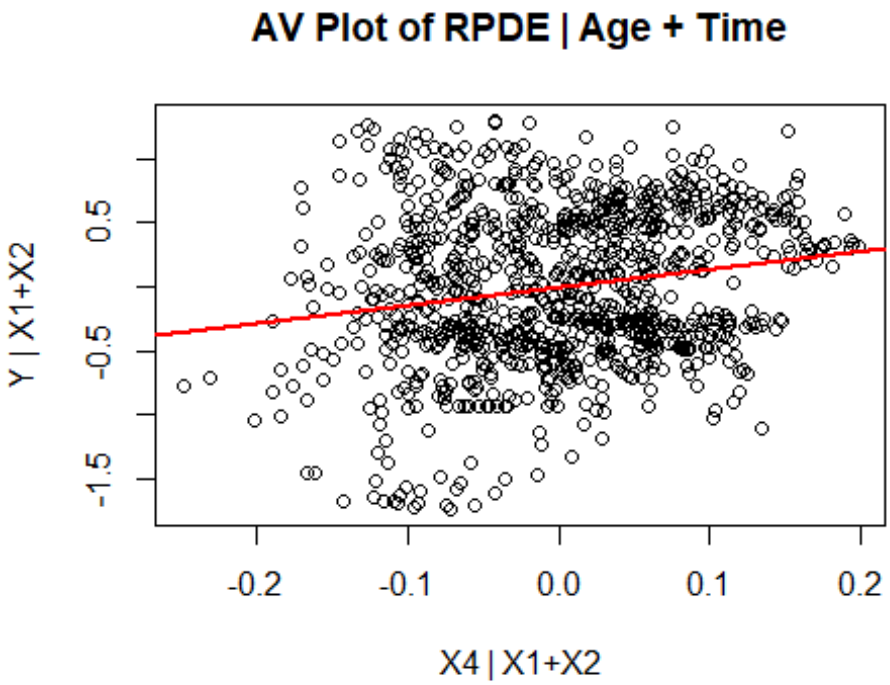
```
plot(rX3, rY, main="AV Plot of PPE | Age + Time", xlab="X3 | X1+X2", ylab="Y
| X1+X2")
abline(m2, col="red", lwd=2)
```




```
m3 <- lm(Y~x1+x2+x4)
rX4 <- residuals(lm(x4~x1+x2))
m4 <- lm(rY~rX4)
summary(m4)

##
## Call:
## lm(formula = rY ~ rX4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.63231 -0.43362  0.00103  0.45979  1.44502
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 7.998e-17  1.896e-02   0.000        1
## rX4         1.426e+00  2.366e-01   6.026 2.37e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5947 on 982 degrees of freedom
## Multiple R-squared:  0.03566,    Adjusted R-squared:  0.03468
## F-statistic: 36.32 on 1 and 982 DF,  p-value: 2.371e-09
```

```
plot(rX4, rY, main="AV Plot of RPDE | Age + Time", xlab="X4 | X1+X2", ylab="Y
| X1+X2")
abline(m4, col="red", lwd=2)
```



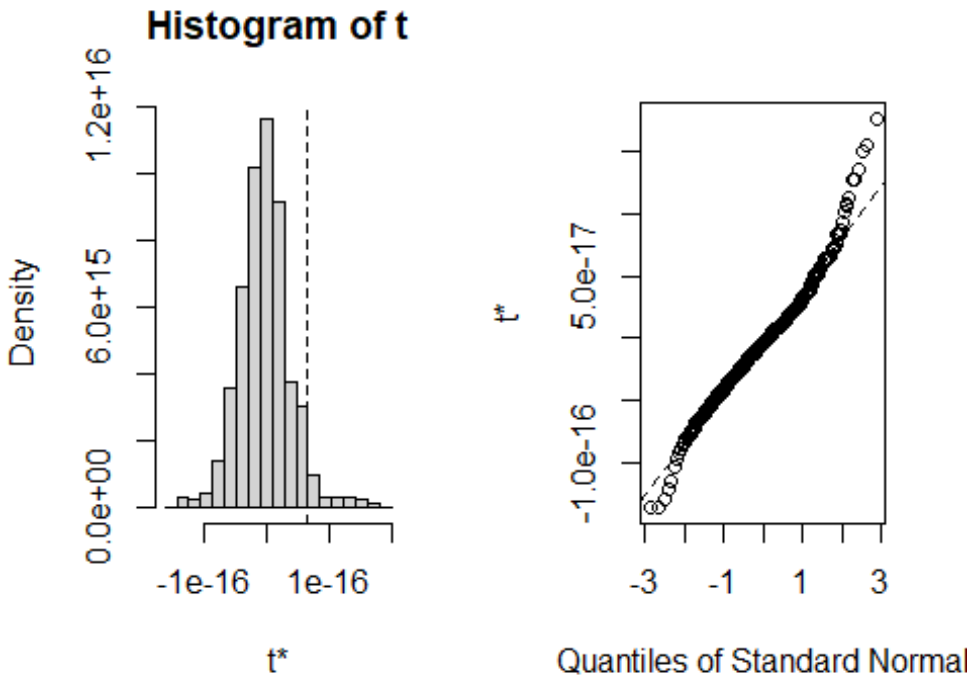

```
# Performing F-test with full and reduced models
reduced = lm(rY~1)
full = lm(rY~rX3+rX4)
anova(reduced, full)
```

```
## Analysis of Variance Table
##
## Model 1: rY ~ 1
## Model 2: rY ~ rX3 + rX4
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    983 360.18
## 2    981 346.20  2    13.977 19.802 3.706e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Bootstrapping to Find Confidence Intervals on Partial Regression Coefficients: Because the data failed the constant variance and normality assumptions, the above conclusions are dubious. For a more confidence answer, we perform bootstrap analysis on the partial regression coefficients. The resulting 95% confidence intervals do not contain zero, so the partial impacts of PPE and RPDE separately are significant, factoring out age and time. This satisfies the "or" condition of our alternative hypothesis. We accept the alternative hypothesis that at least one measure of vocal entropy is a significant predictor of UPDRS score, regardless of a subject's age or at what time within a 6-month window the measurement is taken.

```
data = outliers_gone
data$y = data$total_UPDRS
data$Y = y**0.42
data$x1 = data$test_time
ageCut = cut(data$age, breaks=c(1,60,65,70,75,100))
ageCategory = factor(ageCut,
levels=c('(1,60]','(60,65]','(65,70]','(70,75]','(75,100]'),
labels=c(0,1,2,3,4))
ageCategory = as.integer(as.character(ageCategory))
data$x2 = ageCategory
data$x3 = data$PPE
data$x4 = data$RPDE
# Partial regression coefficient for X3 on X1 and X2
partial_coef1 <- function(data, indices, maxit=20) {
  data <- data[indices,] # selecting sample with boot
  rY <- residuals(lm(Y~x1+x2, data=data))
  rX3 <- residuals(lm(x3~x1+x2, data=data))
  m2 <- lm(rY~rX3, data)
  return(coef(m2))
}
# Partial regression coefficient for X4 on X1 and X2
partial_coef2 <- function(data, indices, maxit=20) {
  data <- data[indices,] # selecting sample with boot
  rY <- residuals(lm(Y~x1+x2, data=data))
  rX4 <- residuals(lm(x4~x1+x2, data=data))
  m4 <- lm(rY~rX4, data)
  return(coef(m4))
}
#Partial regression for X3 - Performing 500 replications with boot
output1 <- boot(data=data, statistic=partial_coef1, R=500, maxit=20)
output1

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = data, statistic = partial_coef1, R = 500, maxit = 20)
##
##
```

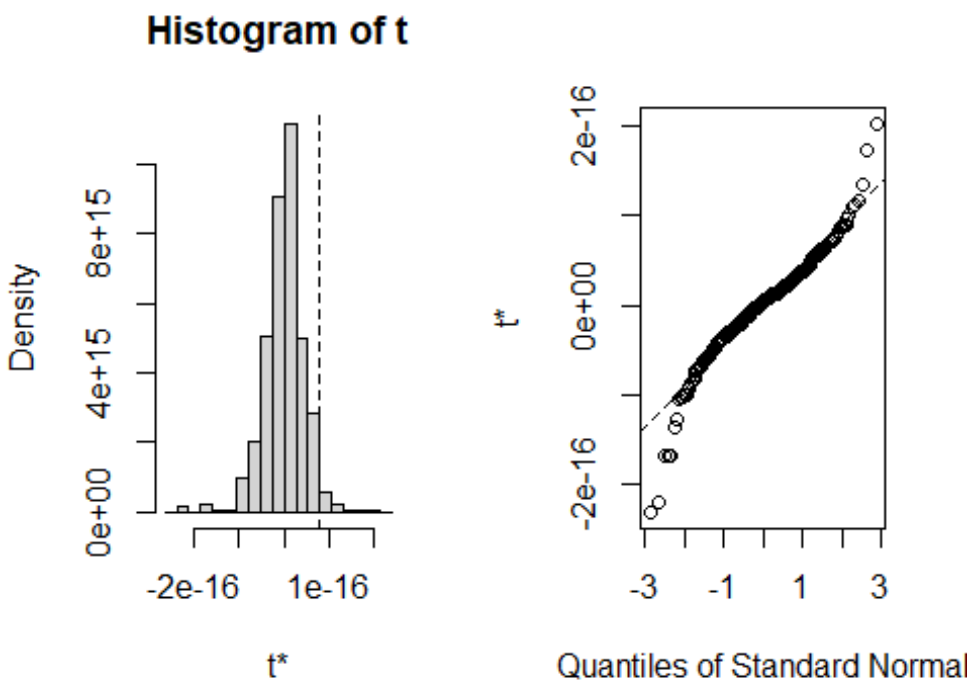
```
## Bootstrap Statistics :
##         original        bias     std. error
## t1* 6.598837e-17 -6.932586e-17 4.139466e-17
## t2* 1.394670e+00 -1.082358e-02 2.404001e-01

plot(output1)
```




```
bluh1 = boot.ci(output1, type="perc", index=2)
bluh1

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 500 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = output1, type = "perc", index = 2)
##
## Intervals :
## Level     Percentile
## 95%   ( 0.921,  1.838 )
## Calculations and Intervals on Original Scale

# Partial regression for X4 - Performing 500 replications with boot
output2 <- boot(data=data, statistic=partial_coef2, R=500, maxit=20)
output2

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
```

```
##
## Call:
## boot(data = data, statistic = partial_coef2, R = 500, maxit = 20)
##
##
## Bootstrap Statistics :
##         original        bias     std. error
## t1* 7.997657e-17 -7.934646e-17 4.554636e-17
## t2* 1.425857e+00  1.386316e-02 2.386045e-01
```

```
plot(output2)
```




```
bluh2 = boot.ci(output2, type="perc", index=2)
bluh2
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 500 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = output2, type = "perc", index = 2)
##
## Intervals :
## Level     Percentile
## 95%   ( 0.977,  1.895 )
## Calculations and Intervals on Original Scale
```

Back-transformation: For completeness, we backtransform the data in the 95% confidence intervals to estimate the range of likely partial linear impacts of PPE and RPDE.

```
# Partial regression coefficient, X3
ci1 = bluh1$percent[4:5]**(1/0.42)
ci1

## [1] 0.8226675 4.2581333

# Partial regression coefficient, X4
ci2 = bluh2$percent[4:5]**(1/0.42)
ci2

## [1] 0.9450181 4.5800437
```

QUESTION 2

Question 2 tests how the UPDRS score is affected by one's vocal qualities (Shimmer, PPE and HNR) and sex.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.1.3
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(fmsb)
```

```
## Warning: package 'fmsb' was built under R version 4.1.3
```

```
library(ALSM)
```

```
## Warning: package 'ALSM' was built under R version 4.1.3
```

```
## Loading required package: leaps
```

```
## Loading required package: SuppDists
```

```
library(onewaytests)
```

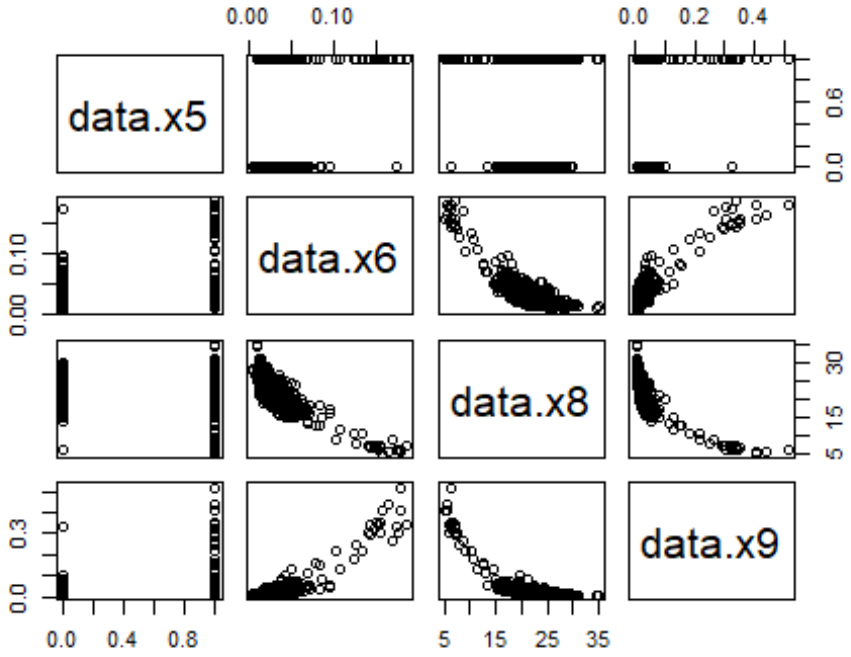
```
## Warning: package 'onewaytests' was built under R version 4.1.3
```

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.3
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

library(readxl)

## Warning: package 'readxl' was built under R version 4.1.3

library(fmsb)
library(boot)

##
## Attaching package: 'boot'

## The following object is masked from 'package:car':
##
##     logit

library(leaps)
library(caret)

## Warning: package 'caret' was built under R version 4.1.3

## Loading required package: ggplot2

## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:boot':
##
##     melanoma

## The following object is masked from 'package:ALSM':
##
##     oneway

## Registered S3 methods overwritten by 'pROC':
##   method    from
##   print.roc fmsb
##   plot.roc  fmsb

data<-read.csv('C:/Users/Ankita Mishra/Desktop/parkinsons_data.csv', header
=TRUE, sep=",")

#modify data
data_group= data
data_group$test_time= as.integer(data_group$test_time)
data_group <- data_group %>%
```

```
  group_by('subject#', age, sex, test_time) %>%
  dplyr::summarise_all("median")%>%
  as.data.frame()

data= data_group

#define variables
data$y = data$total_UPDRS
data$x5 = data$sex
data$x6= data$Shimmer
data$x8= data$HNR
data$x9 = data$NHR

#correlation between the variables
df= data.frame(data$x5, data$x6, data$x8, data$x9)
pairs(df)
```




```
cor(df)

##             data.x5     data.x6      data.x8    data.x9
## data.x5 1.000000000  0.06670335  0.008474488  0.1721074
## data.x6 0.066703347  1.00000000 -0.829548281  0.8880035
## data.x8 0.008474488 -0.82954828  1.000000000 -0.7199781
## data.x9 0.172107363  0.88800347 -0.719978074  1.0000000
```

The full model includes all the variables and the interaction terms. Running a Shapiro test and Brown-Forsythe test on the data tells us the data is non-normal and doesn't have constant variance as interpreted by the p-value.

```
#full model

fullmod =
lm(y~x9+x6+x8+as.factor(x5)+as.factor(x5)*x9+as.factor(x5)*x6+as.factor(x5)*x
8, data)

res= resid(fullmod)
shapiro.test(res)

##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.98786, p-value = 4.228e-07

library(onewaytests)
data$gr = cut(data$y, 5)
data$residual = fullmod$residuals
bf.test(residual~gr, data)

##
##   Brown-Forsythe Test (alpha = 0.05)
## ------------------------------------------------------------
##   data : residual and gr
##
##   statistic  : 1040.797
##   num df     : 4
##   denom df   : 376.9707
##   p.value    : 3.404589e-202
##
##   Result     : Difference is statistically significant.
## ------------------------------------------------------------
```

Checking for influential points, row 25 seems to be the outlier hence it is removed from the data.

```
#remove the influential point
data= data[-c(25), ]
fullmod =
lm(y~x9+x6+x8+as.factor(x5)+as.factor(x5)*x9+as.factor(x5)*x6+as.factor(x5)*x
8, data)

res = resid(fullmod)
shapiro.test(res)

##
##  Shapiro-Wilk normality test
```

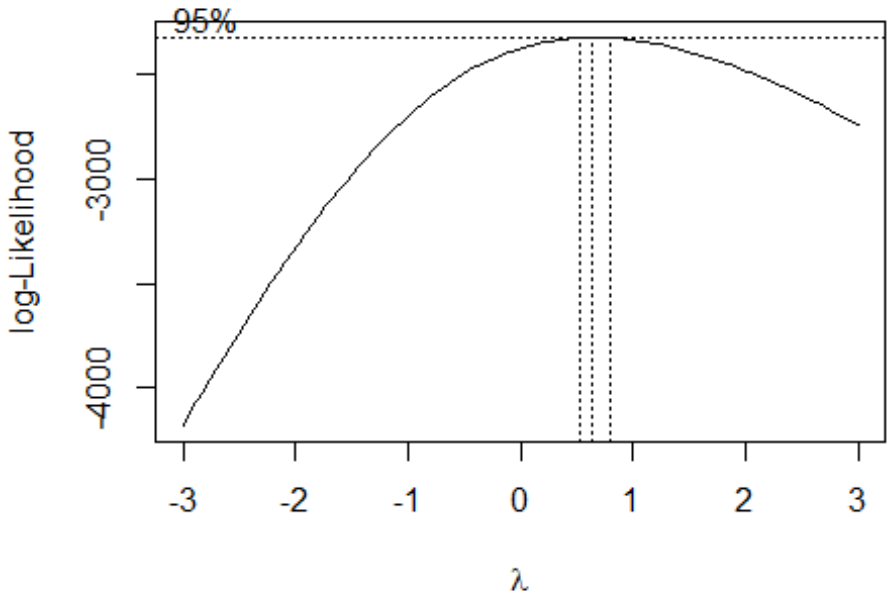
```
##
## data:  res
## W = 0.988, p-value = 4.923e-07
```

```
library(onewaytests)
data$gr = cut(data$y, 5)
data$residual = fullmod$residuals
bf.test(residual~gr, data)
```

```
##
##   Brown-Forsythe Test (alpha = 0.05)
## ------------------------------------------------------------
##   data : residual and gr
##
##   statistic  : 1048.33
##   num df     : 4
##   denom df   : 372.3247
##   p.value    : 3.826669e-201
##
##   Result     : Difference is statistically significant.
## ------------------------------------------------------------
```

Since our model is still non-normal and doesn't have constant variance, we transform Y. This gives us a lambda value= 1.060606

```
#find lambda and transform Y
library(MASS)
bcmle= boxcox(fullmod, lambda= seq(-3, 3, by=0.1))
```

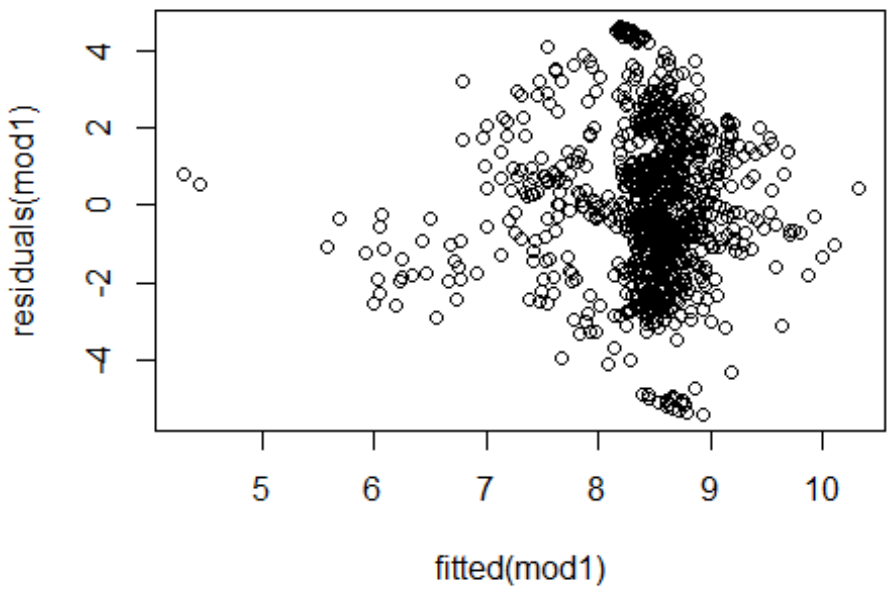



```
lambda= bcmle$x[which.max(bcmle$y)]
lambda

## [1] 0.6363636

data$newy= (data$y)**lambda

#modified model after transformation
modY=
lm(newy~x9+x6+x8+as.factor(x5)+as.factor(x5)*x9+as.factor(x5)*x6+as.factor(x5
)*x8, data)

res= resid(modY)
shapiro.test(res)

##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99113, p-value = 1.691e-05

data$gr= cut(data$newy, 5)
data$residual = modY$residuals
bf.test(residual~gr, data)

##
##   Brown-Forsythe Test (alpha = 0.05)
```

```
## ----------------------------------------------------------
##   data : residual and gr
##
##   statistic  : 658.0907
##   num df     : 4
##   denom df   : 149.5834
##   p.value    : 8.136837e-94
##
##   Result     : Difference is statistically significant.
## ----------------------------------------------------------
```

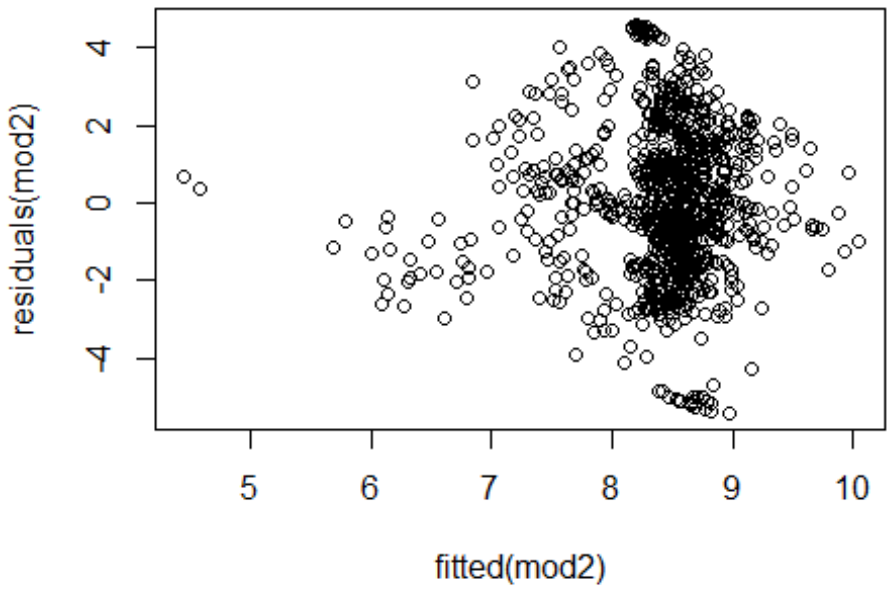
To reduce the non-normality and non-contant variance, we attempted the weighted least squares method

```
# weighted least squares method
wts1=1/fitted(lm(abs(residuals(modY))~x9+x6+x8+as.factor(x5)+as.factor(x5)*x9
+as.factor(x5)*x8+as.factor(x5)*x8, data))^2
mod1=lm(newy~x9+x6+x8+as.factor(x5)+as.factor(x5)*x9+as.factor(x5)*x6+as.fact
or(x5)*x8, weight=wts1, data=data)

res = resid(mod1)
shapiro.test(res)

##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99231, p-value = 7.439e-05

library(onewaytests)
data$gr = cut(data$newy, 5)
data$residual = mod1$residuals
bf.test(residual~gr, data)

##
##   Brown-Forsythe Test (alpha = 0.05)
## ----------------------------------------------------------
##   data : residual and gr
##
##   statistic  : 743.4287
##   num df     : 4
##   denom df   : 167.8219
##   p.value    : 1.399766e-105
##
##   Result     : Difference is statistically significant.
## ----------------------------------------------------------

plot(fitted(mod1), residuals(mod1))
```




```
wts2=1/fitted(lm(abs(residuals(mod1))~x9+x6+x8+as.factor(x5)+as.factor(x5)*x9
+as.factor(x5)*x8+as.factor(x5)*x8, data))^2
mod2=lm(newy~x9+x6+x8+as.factor(x5)+as.factor(x5)*x9+as.factor(x5)*x6+as.fact
or(x5)*x8, weight=wts2, data=data)

res = resid(mod2)
shapiro.test(res)

##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99192, p-value = 4.488e-05

data$gr = cut(data$newy, 5)
data$residual = mod2$residuals
bf.test(residual~gr, data)

##
##   Brown-Forsythe Test (alpha = 0.05)
## ------------------------------------------------------------
##   data : residual and gr
##
##   statistic  : 776.0966
##   num df     : 4
##   denom df   : 166.243
##   p.value    : 2.252147e-106
```

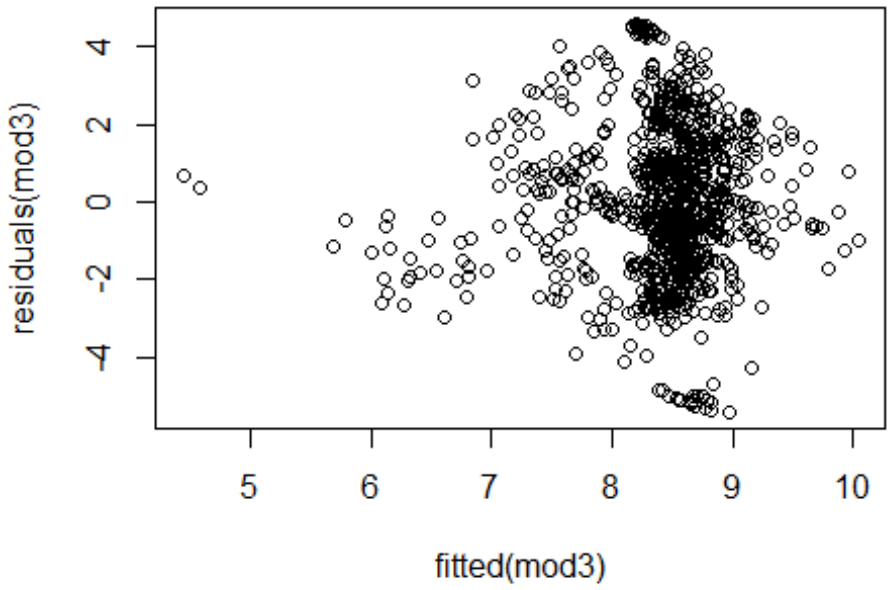
```
##
##   Result     : Difference is statistically significant.
## ------------------------------------------------------------

plot(fitted(mod2), residuals(mod2))
```




```
wts3=1/fitted(lm(abs(residuals(mod2))~x9+x6+x8+as.factor(x5)+as.factor(x5)*x9
+as.factor(x5)*x8+as.factor(x5)*x8, data))^2
mod3=lm(newy~x9+x6+x8+as.factor(x5)+as.factor(x5)*x9+as.factor(x5)*x6+as.fact
or(x5)*x8, weight=wts2, data=data)

res = resid(mod3)
shapiro.test(res)

##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99192, p-value = 4.488e-05

data$gr = cut(data$newy, 5)
data$residual = mod3$residuals
bf.test(residual~gr, data)

##
##   Brown-Forsythe Test (alpha = 0.05)
## ------------------------------------------------------------
##   data : residual and gr
```

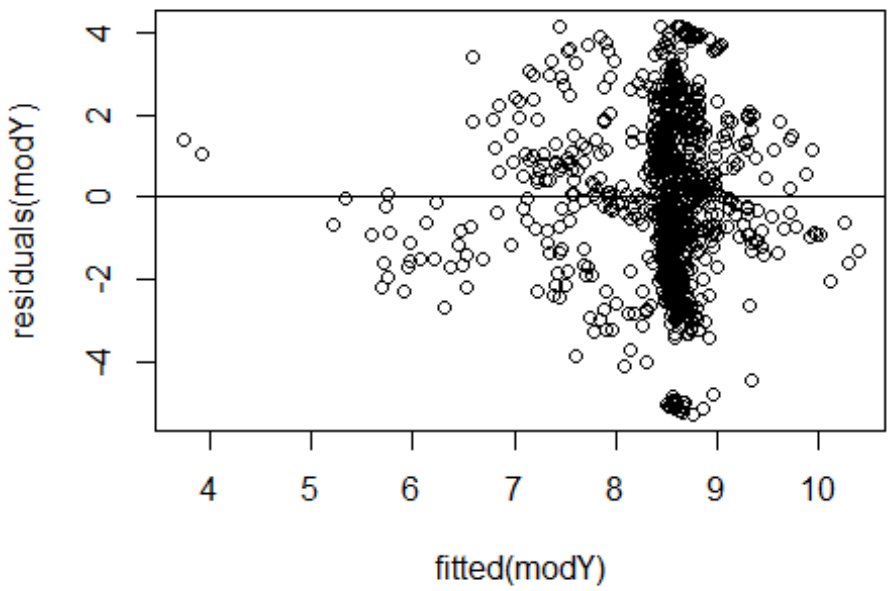
```
##
##   statistic  : 776.0966
##   num df     : 4
##   denom df   : 166.243
##   p.value    : 2.252147e-106
##
##   Result     : Difference is statistically significant.
## ----------------------------------------------------------------
```

```
plot(fitted(mod3), residuals(mod3))
```

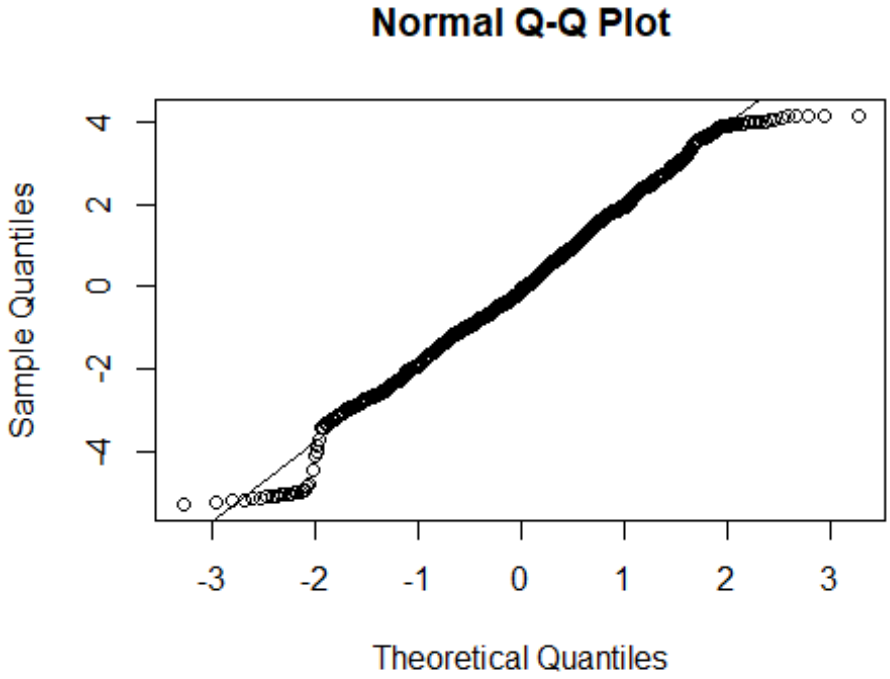


The WLS made the model worse, thus we continue with the transformed Y model.

```
plot(fitted(modY), residuals(modY))
abline(0,0)
```




```
res= resid(modY)
qqnorm(res)
qqline(res)
```



```r
#bootstrapping
library(boot)

boot.reg <- function(data, indices, maxit=1000) {
  data <- data[indices,]
  data.mod<-
lm(newy~x9+x6+x8+as.factor(x5)+as.factor(x5)*x9+as.factor(x5)*x6+as.factor(x5
)*x8, data=data)

  coef(data.mod)
}

data_model_reg <- boot(data=data, statistic = boot.reg, R=1000, maxit=100)
data_model_reg

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = data, statistic = boot.reg, R = 1000, maxit = 100)
##
##
## Bootstrap Statistics :
##         original        bias    std. error
## t1*   8.00906020  0.0109258024  1.33473673
## t2*  -8.35976073 -0.7319032358 11.35629173
## t3*  16.50951813  0.6121424954  9.30122174
## t4*   0.01229356 -0.0006481188  0.04955960
## t5*  10.43710904 -0.0003499511  1.64432355
## t6*   7.07129564  0.7996107679 11.76433561
## t7* -57.64145419 -0.9359035131 13.42136499
## t8*  -0.42003480  0.0005892862  0.06020885

summary(modY)

##
## Call:
## lm(formula = newy ~ x9 + x6 + x8 + as.factor(x5) + as.factor(x5) *
##     x9 + as.factor(x5) * x6 + as.factor(x5) * x8, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.278 -1.234 -0.101  1.378  4.155
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         8.00906    1.21757   6.578 7.89e-11 ***
## x9                 -8.35976    9.62612  -0.868 0.385371
## x6                 16.50952    9.41727   1.753 0.079907 .
```

```
## x8                  0.01229    0.04401   0.279 0.780028
## as.factor(x5)1     10.43711    1.75301   5.954 3.69e-09 ***
## x9:as.factor(x5)1   7.07130   10.75658   0.657 0.511089
## x6:as.factor(x5)1 -57.64145   16.67403  -3.457 0.000571 ***
## x8:as.factor(x5)1  -0.42003    0.06352  -6.613 6.29e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.928 on 945 degrees of freedom
## Multiple R-squared:  0.107,  Adjusted R-squared:  0.1003
## F-statistic: 16.17 on 7 and 945 DF,  p-value: < 2.2e-16

#cross-validation
library(MASS)
library(leaps)
library(caret)

set.seed(123)

train.control=trainControl(method="cv", number=10)
step.model=train(newy~x9+x6+x8+as.factor(x5)+as.factor(x5)*x9+as.factor(x5)*x
6+as.factor(x5)*x8, data, method="leapBackward",
tuneGrid=data.frame(nvmax=5), trControl=train.control)
step.model$results

##   nvmax   RMSE Rsquared      MAE    RMSESD RsquaredSD      MAESD
## 1     5 1.9288 0.103262 1.555715 0.0903344 0.04377808 0.06502607

summary(modY)

##
## Call:
## lm(formula = newy ~ x9 + x6 + x8 + as.factor(x5) + as.factor(x5) *
##     x9 + as.factor(x5) * x6 + as.factor(x5) * x8, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.278 -1.234 -0.101  1.378  4.155
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         8.00906    1.21757   6.578 7.89e-11 ***
## x9                 -8.35976    9.62612  -0.868 0.385371
## x6                 16.50952    9.41727   1.753 0.079907 .
## x8                  0.01229    0.04401   0.279 0.780028
## as.factor(x5)1     10.43711    1.75301   5.954 3.69e-09 ***
## x9:as.factor(x5)1   7.07130   10.75658   0.657 0.511089
## x6:as.factor(x5)1 -57.64145   16.67403  -3.457 0.000571 ***
## x8:as.factor(x5)1  -0.42003    0.06352  -6.613 6.29e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.928 on 945 degrees of freedom
## Multiple R-squared:  0.107,  Adjusted R-squared:  0.1003
## F-statistic: 16.17 on 7 and 945 DF,  p-value: < 2.2e-16

fullmod=
lm(newy~x9+x6+x8+as.factor(x5)+as.factor(x5)*x9+as.factor(x5)*x6+as.factor(x5
)*x8, data)
redmod= lm(newy~as.factor(x5), data)
anova(redmod, fullmod)

## Analysis of Variance Table
##
## Model 1: newy ~ as.factor(x5)
## Model 2: newy ~ x9 + x6 + x8 + as.factor(x5) + as.factor(x5) * x9 +
as.factor(x5) *
##     x6 + as.factor(x5) * x8
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    951 3878.8
## 2    945 3512.9  6    365.87 16.404 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

qf(0.95, 6, 945)

## [1] 2.108158
```

Since Fs= 16.404 > F(0.95, 6, 945) = 2.108158, we reject H0 and hence vocal qualities like Shimmer, HNR, NHR and their interaction terms with sex have a significant impact on UPDRS score

Research Question 3:

3. **Do age ranges (36,45.8], (45.8,55.6], (55.6,65.4], (65.4,75.2], (75.2,85] have the same total_UPDRS scores for any NHR, DFA, and HNR values?**

Reduced Model: $Y = B_0 + B_9X_9 + B_{10}X_{10} + B_8X_8$

Full Model: $Y = B_0 + B_9X_9 + B_{10}X_{10} + B_8X_8 + B_{21}X_{21} + B_{22}X_{22} + B_{23}X_{23} + B_{24}X_{24} + B_{921}X_9X_{21} + B_{1021}X_{10}X_{21} + B_{821}X_8X_{21} + B_{922}X_9X_{22} + B_{1022}X_{10}X_{22} + B_{822}X_8X_{22} + B_{923}X_9X_{23} + B_{1023}X_{10}X_{23} + B_{823}X_8X_{23} + B_{924}X_9X_{24} + B_{1024}X_{10}X_{24} + B_{824}X_8X_{24}$

$\boldsymbol{H_0}$: $B_{21} = B_{22} = B_{23} = B_{24} = B_{921} = B_{1021} = B_{821} = B_{922} = B_{1022} = B_{822} = B_{923} = B_{1023} = B_{823} = B_{924} = B_{1024} = B_{824} = 0$

$\boldsymbol{H_a}$: at least one of $B_{21}, B_{22}, B_{23}, B_{24}, B_{921}, B_{1021}, B_{821}, B_{922}, B_{1022}, B_{822}, B_{923}, B_{1023}, B_{823}, B_{924}, B_{1024}, or\ B_{824}$ is not equal to 0

First, we will modify the data, since there were repetitive measurements for every patient in the study to avoid multicollinearity issues:

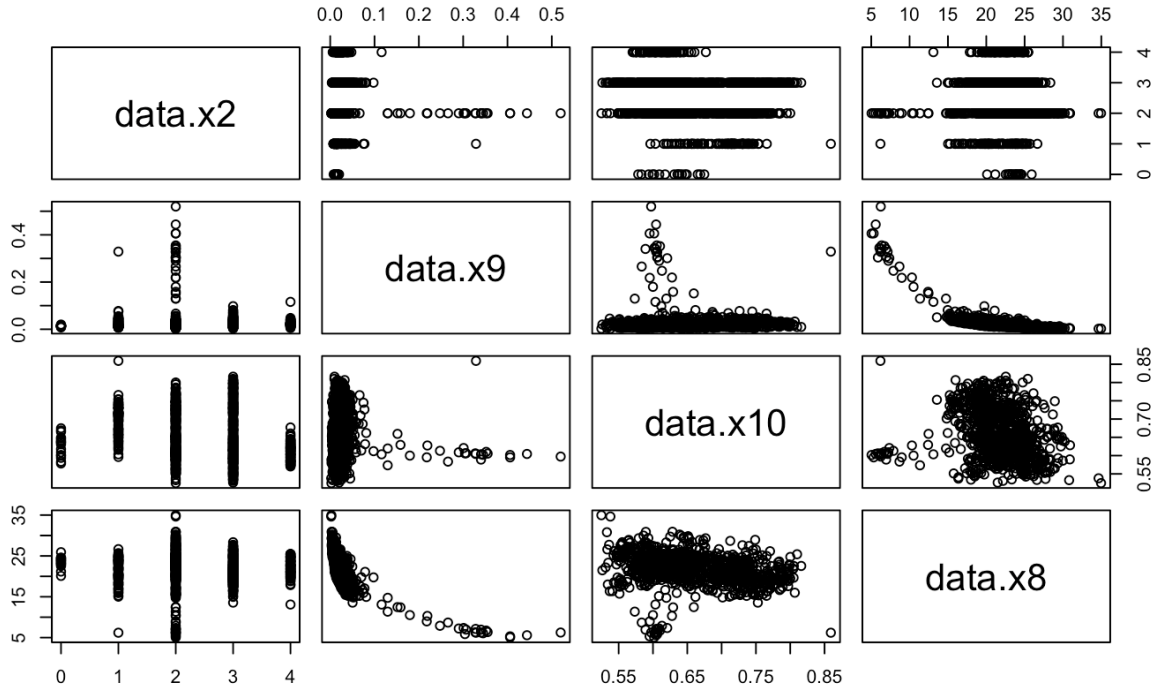
```
library(dplyr)

#modify data
data_group = data
data_group$test_time = as.integer(data_group$test_time)
data_group <- data_group %>%
  group_by(`subject#`,age,sex,test_time) %>%
  dplyr::summarise_all("median") %>%
  as.data.frame()
data_group = subset(data_group, select = -c(`subject#`))
data = data_group
```

We then define our variables and make it ready to be put in the model:

```
data$y=data$total_UPDRS

ageCut = cut(data$age,5)
temp = factor(ageCut, levels=c('(36,45.8]','(45.8,55.6]','(55.6,65.4]','(65.4,75.2]','(75.2,85]'), labels=c(0,1,2,3,4))
temp = as.integer(as.character(temp))

data$x2=temp
data$x9=data$NHR
data$x10=data$DFA
data$x8=data$HNR
```

We first make sure we don't have any multicollinearity issues:

```
df=data.frame(data$x2,data$x9,data$x10,data$x8)
pairs(df)
cor(df)
```




```
             data.x2     data.x9    data.x10     data.x8
data.x2   1.00000000 -0.04313492 -0.16611314 -0.04780366
data.x9  -0.04313492  1.00000000 -0.05705757 -0.71102192
data.x10 -0.16611314 -0.05705757  1.00000000 -0.25716980
data.x8  -0.04780366 -0.71102192 -0.25716980  1.00000000
```

$Corr(X_9, X_8) = -0.71102192$ so, we should use VIF to check out if it is an issue.

```
> VIF(lm(x9~x10+x8+as.factor(x2)))
[1] 2.58644
> VIF(lm(x10~x9+x8+as.factor(x2)))
[1] 1.37611
> VIF(lm(x8~x9+x10+as.factor(x2)))
[1] 2.802691
> VIF(lm(x9~x8))
[1] 2.05495
```

Thus, it doesn't seem like we have any multicollinearity issues.

```
full = lm(y~x9+x10+x8+as.factor(x2)+as.factor(x2)*x9+as.factor(x2)*x10+as.factor(x2)*x8,data)

res = resid(full)
shapiro.test(res)

library(onewaytests)
data$group=cut(data$y, 5)
data$residual=full$residuals
bf.test(residual~group,data)
```

```
        Shapiro-Wilk normality test

data:  res
W = 0.98326, p-value = 3.106e-09

  Brown-Forsythe Test (alpha = 0.05)
-------------------------------------------------------------
  data : residual and group

  statistic  : 408.1565
  num df     : 4
  denom df   : 441.5665
  p.value    : 8.10249e-147

  Result     : Difference is statistically significant.
-------------------------------------------------------------
```

It seems our model is non-normal and doesn't have constant variance.

```
> p=20
> n=length(data$y)
> pt2=qf(0.2,p,n-p)
> pt5=qf(0.5,p,n-p)
> s=0
> for(i in 1:n) {
+   val = pf(cooks.distance(full)[i],p,n-p)
+   if(val < pt2) {
+     s = s + 1
+   }
+ }
> print(s)
[1] 989
> s=0
> for(i in 1:n) {
+   val = pf(cooks.distance(full)[i],p,n-p)
+   if(val > pt5) {
+     s = s + 1
+   }
+ }
> print(s)
[1] 0
> sum(abs(dffits(full))>1)
[1] 2
> for(i in 1:n) {
+   if (abs(dffits(full)[i]) > 1) {
+     print(abs(dffits(full)[i]))
+   }
+ }
     685
5.494834
     928
1.402511
```

```
> for(i in 1:p) {
+   print(sum(abs(dfbetas(full)[,i])>1))
+ }
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
```

It seems row 685 seems to be a huge influential problem, the other row seems fine, so we will remove it.

```
data = data[-c(685), ]
```

```
full = lm(y~x9+x10+x8+as.factor(x2)+as.factor(x2)*x9+as.factor(x2)*x10+as.factor(x2)*x8,data)

res = resid(full)
shapiro.test(res)

library(onewaytests)
data$group=cut(data$y, 5)
data$residual=full$residuals
bf.test(residual~group,data)
```

```
        Shapiro-Wilk normality test

data:  res
W = 0.98371, p-value = 4.675e-09
```

```
  Brown-Forsythe Test (alpha = 0.05)
-------------------------------------------------------------
  data : residual and group

  statistic  : 405.1789
  num df     : 4
  denom df   : 442.6622
  p.value    : 1.911767e-146

  Result     : Difference is statistically significant.
-------------------------------------------------------------
```

```
> p=20
> n=length(data$y)
> pt2=qf(0.2,p,n-p)
> pt5=qf(0.5,p,n-p)
> s=0
> for(i in 1:n) {
+   val = pf(cooks.distance(full)[i],p,n-p)
+   if(val < pt2) {
+     s = s + 1
+   }
+ }
> print(s)
[1] 989
> s=0
> for(i in 1:n) {
+   val = pf(cooks.distance(full)[i],p,n-p)
+   if(val > pt5) {
+     s = s + 1
+   }
+ }
> print(s)
[1] 0
> sum(abs(dffits(full))>1)
[1] 1
> for(i in 1:n) {
+   if (abs(dffits(full)[i]) > 1) {
+     print(abs(dffits(full)[i]))
+   }
+ }
     928
1.404024
```

```
> for(i in 1:p) {
+   print(sum(abs(dfbetas(full)[,i])>1))
+ }
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
```

It seems our model is non-normal and doesn't have constant variance, and we are left with one influential point near 1, though it seems best if we transform Y.

```
library(MASS)
bcmle=boxcox(full,lambda=seq(-3,3,by=0.1))
lambda=bcmle$x[which.max(bcmle$y)]
lambda
```

Since $\lambda = 0.5151515$ we will make $Y' = Y^{0.5151515}$.

```
data$yNew = (data$y)**lambda
modY=lm(yNew~x9+x10+x8+as.factor(x2)+as.factor(x2)*x9+as.factor(x2)*x10+as.factor(x2)*x8,data)

res = resid(modY)
shapiro.test(res)

data$group=cut(data$yNew, 5)
data$residual=modY$residuals
bf.test(residual~group,data)
```

```
        Shapiro-Wilk normality test

data:  res
W = 0.98824, p-value = 3.992e-07
```

```
  Brown-Forsythe Test (alpha = 0.05)
------------------------------------------------------------
  data : residual and group

  statistic  : 333.929
  num df     : 4
  denom df   : 214.4986
  p.value    : 7.026453e-91

  Result     : Difference is statistically significant.
------------------------------------------------------------
```

Our normality and variance p-values have improved significantly, but we still have non-normality and non-constant variance in our model. We will now try WLS.

```
wts1=1/fitted(lm(abs(residuals(modY))~x9+x10+x8+as.factor(x2)+as.factor(x2)*x9+as.factor(x2)*x10+as.factor(x2)*x8, data))^2
mod1=lm(yNew~x9+x10+x8+as.factor(x2)+as.factor(x2)*x9+as.factor(x2)*x10+as.factor(x2)*x8, weight=wts1, data=data)

res = resid(mod1)
shapiro.test(res)

data$group=cut(data$yNew, 5)
data$residual=mod1$residuals
bf.test(residual~group,data)
```

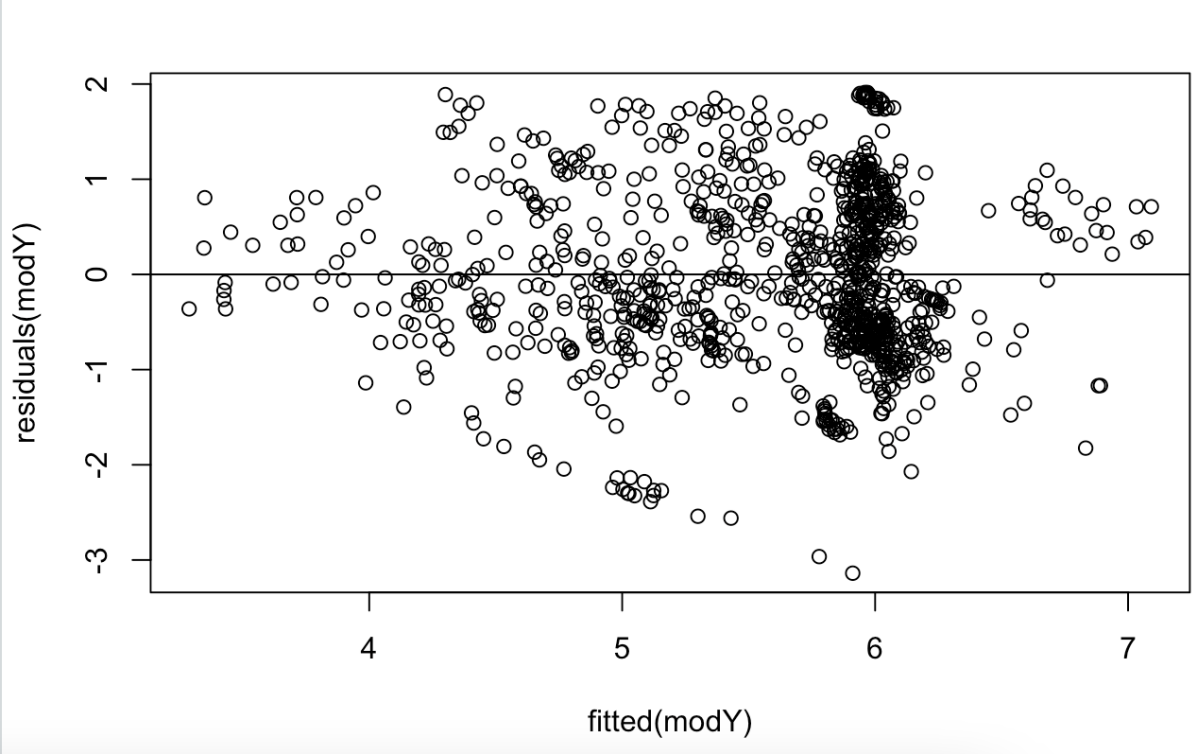
```
        Shapiro-Wilk normality test

data:  res
W = 0.98416, p-value = 7.049e-09
```

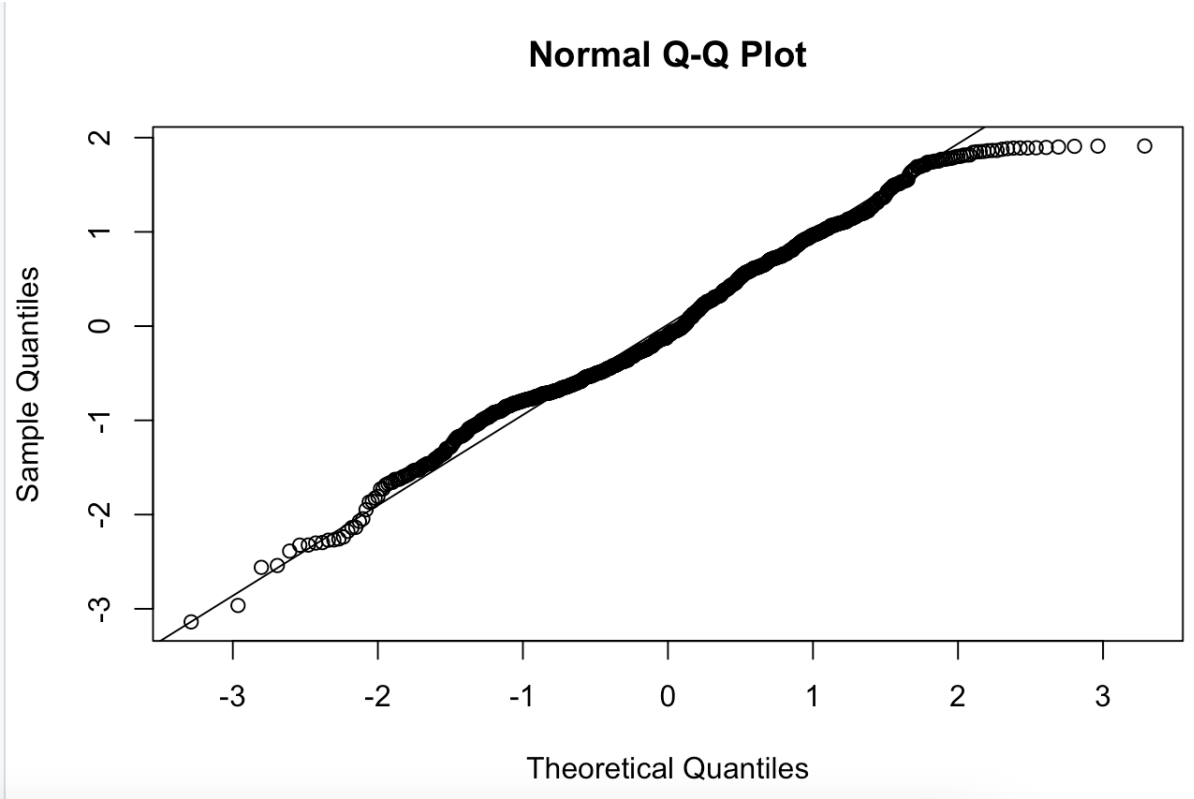
```
  Brown-Forsythe Test (alpha = 0.05)
------------------------------------------------------------
  data : residual and group

  statistic  : 410.8333
  num df     : 4
  denom df   : 248.7908
  p.value    : 2.697124e-108

  Result     : Difference is statistically significant.
------------------------------------------------------------
```

Since our p-values got smaller we have become more non-normal and non-constant variance, thus we will proceed with the transformed Y model and check out the rest of our diagnostics.

```
plot(fitted(modY),residuals(modY))
abline(0,0)
```

```
res = resid(modY)
qqnorm(res)
qqline(res)
```


residuals(modY)
fitted(modY)
2
1
0
-1
-2
-3
4
5
6
7


Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3

The residual plot looks fine, while we now know the reason why our normality condition is not met, it seems some of our data points after the 2nd theoretical quantiles are not following the normal distribution.

```
> p=20
> n=length(data$yNew)
> pt2=qf(0.2,p,n-p)
> pt5=qf(0.5,p,n-p)
> s=0
> for(i in 1:n) {
+   val = pf(cooks.distance(modY)[i],p,n-p)
+   if(val < pt2) {
+     s = s + 1
+   }
+ }
> print(s)
[1] 989
> s=0
> for(i in 1:n) {
+   val = pf(cooks.distance(modY)[i],p,n-p)
+   if(val > pt5) {
+     s = s + 1
+   }
+ }
> print(s)
[1] 0
```

989 and 0 are printed, which means that none of our points are influential when using cooks distance.

```
> sum(abs(dffits(modY))>1)
[1] 1
> for(i in 1:n) {
+   if (abs(dffits(modY)[i]) > 1) {
+     print(abs(dffits(modY)[i]))
+   }
+ }
     928
1.321544
```

Dffits on the other hand returns an influential point which is 1.321544, but since it isn't that far away from 1, we will continue without fixing the point.

```
> for(i in 1:p) {
+   print(sum(abs(dfbetas(modY)[,i])>1))
+ }
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
```

Dfbetas returns all 0's as well, so we won't perform remedial measures on the one data point we found which was influential in Dffits.

```r
library(boot)

boot.reg <- function(data, indices, maxit=1000) {
  data <- data[indices,]
  data.mod<-lm(yNew~x9+x10+x8+as.factor(x2)+as.factor(x2)*x9+as.factor(x2)*x10+as.factor(x2)*x8,data=data)
  coef(data.mod)
}

data_model_reg <- boot(data=data, statistic = boot.reg, R=1000, maxit=100)
data_model_reg
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = data, statistic = boot.reg, R = 1000, maxit = 100)


Bootstrap Statistics :
        original       bias    std. error
t1*    8.3367600 -0.003542145   4.1173913
t2*  -74.3927574 -0.941926601  53.9961865
t3*  -10.0121156 -0.488618678   4.2048526
t4*    0.1154472  0.013626328   0.1174599
t5*   -6.0663422  0.042808421   4.3241673
t6*    6.6797414  0.025314871   4.2481656
t7*   -0.6548691 -0.011866268   4.2068292
t8*   12.3874769  0.379188961   4.6743556
t9*   54.5001922  0.124941538  54.2785622
t10*  67.4649363  0.843469202  53.9804509
t11*  70.7921721  0.846909249  54.2933247
t12*  50.7363501 -2.433874297  55.0025757
t13*  17.8524436  0.512734399   4.5235415
t14*   0.4637564  0.485268012   4.3191898
t15*   8.8333086  0.495539418   4.2749213
t16*  -9.9385508  0.071260200   5.7743923
t17*  -0.1909925 -0.015567332   0.1197447
t18*  -0.2704454 -0.014404076   0.1189532
t19*  -0.1546157 -0.012983539   0.1195554
t20*  -0.1954853 -0.015802921   0.1304849
```

```
> summary(modY)

Call:
lm(formula = yNew ~ x9 + x10 + x8 + as.factor(x2) + as.factor(x2) *
    x9 + as.factor(x2) * x10 + as.factor(x2) * x8, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-3.13862 -0.63015 -0.09322  0.66387  1.91114

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          8.3368     7.4427   1.120   0.2629
x9                 -74.3928   103.2641  -0.720   0.4714
x10                -10.0121     8.0913  -1.237   0.2162
x8                   0.1154     0.2071   0.557   0.5774
as.factor(x2)1      -6.0663     7.9372  -0.764   0.4449
as.factor(x2)2       6.6797     7.4942   0.891   0.3730
as.factor(x2)3      -0.6549     7.4913  -0.087   0.9304
as.factor(x2)4      12.3875     8.0021   1.548   0.1219
x9:as.factor(x2)1   54.5002   103.9142   0.524   0.6001
x9:as.factor(x2)2   67.4649   103.2712   0.653   0.5137
x9:as.factor(x2)3   70.7922   103.3665   0.685   0.4936
x9:as.factor(x2)4   50.7364   103.9022   0.488   0.6254
x10:as.factor(x2)1  17.8524     8.5660   2.084   0.0374 *
x10:as.factor(x2)2   0.4638     8.1380   0.057   0.9546
x10:as.factor(x2)3   8.8333     8.1176   1.088   0.2768
x10:as.factor(x2)4  -9.9386     9.3055  -1.068   0.2858
x8:as.factor(x2)1   -0.1910     0.2164  -0.882   0.3778
x8:as.factor(x2)2   -0.2704     0.2078  -1.301   0.1935
x8:as.factor(x2)3   -0.1546     0.2084  -0.742   0.4584
x8:as.factor(x2)4   -0.1955     0.2199  -0.889   0.3741
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9045 on 969 degrees of freedom
Multiple R-squared:  0.3504,    Adjusted R-squared:  0.3376
F-statistic:  27.5 on 19 and 969 DF,  p-value: < 2.2e-16
```

Bootstrapping produces around the same results as our model for transformed Y though for x9, and x9 vs. all the age groups the standard errors bootstrapping produces are half of what the lm function produces.

```
library(MASS)
library(leaps)
library(caret)

set.seed(123)

train.control=trainControl(method="cv", number=10)
step.model=train(yNew~x9+x10+x8+as.factor(x2)+as.factor(x2)*x9+as.factor(x2)*x10+as.factor(x2)*x8, data=data, method="leapBackward",
                 tuneGrid=data.frame(nvmax=5), trControl=train.control)
step.model$results
```

```
  nvmax      RMSE Rsquared       MAE      RMSESD RsquaredSD       MAESD
1     5 0.956432 0.261161 0.7824142 0.05718571 0.06419157 0.04468827
```

Root MSE (RMSE) = 0.956432, which is around the same residual standard error our transformed Y model produces which is 0.9045.

Since we have now decided on the transformed Y model, we will now perform our hypothesis testing:

```
> red = lm(yNew~x9+x10+x8, data=data)
> full = lm(yNew~x9+x10+x8+as.factor(x2)+as.factor(x2)*x9+as.factor(x2)*x10+as.factor(x2)*x8, data=data)
> anova(red,full)
Analysis of Variance Table

Model 1: yNew ~ x9 + x10 + x8
Model 2: yNew ~ x9 + x10 + x8 + as.factor(x2) + as.factor(x2) * x9 + as.factor(x2) *
    x10 + as.factor(x2) * x8
  Res.Df     RSS Df Sum of Sq     F    Pr(>F)
1    985 1082.39
2    969  792.82 16    289.58 22.12 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> qf(0.95,16,969)
[1] 1.653953
```

Since $F_s = 22.12 > F(0.95,16,969) = 1.653953$, we reject H_0, and conclude that the age ranges (36,45.8], (45.8,55.6], (55.6,65.4], (65.4,75.2], (75.2,85] produce different total_UPDRS scores for any NHR, DFA, and HNR values.

- After performing the hypothesis testing we can conclude that age as a categorical predictor is significant in our model, along with its interactions with NHR, DFA, and HNR. Further testing may also allow normality and variance conditions to be met on the research model to strengthen the conclusion we have reached.
- The idea for this research question came from the fact that as an individual becomes older they develop or worsen their health conditions, and thus we wanted to see if age caused an individual's Parkinson's total UPDRS score to change.

Full Model

Overall R code for the final model selection section

```
install.packages('ALSM')
install.packages('onewaytests')
install.packages('MASS')
install.packages('fmsb')
library(ALSM)
library(onewaytests)
library(MASS)
library(readxl)
library(fmsb)

#import data
data <- read_excel("Desktop/parkinsons_data.xlsx")

#the initial model
model <- lm(total_UPDRS~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA, data)
summary(model)

#check for assumptions
#constant variance
data$group <- cut(data$total_UPDRS, 3)
data$residual <- model$residuals
bf.test(residual ~ group, data)

#normality
shapiro.test(residuals(model)[1:5000])
qqnorm(residuals(model))
qqline(residuals(model))

#linearity
plot(data$test_time+data$PPE+data$RPDE+data$Shimmer+data$`Jitter(Abs)`+data$HNR+data
$NHR+data$DFA, residuals(model), ylab="residuals", xlab="fitted values")
abline(0,0)

#multicolinearity
muldata <-
data.frame(data$test_time,data$PPE,data$RPDE,data$Shimmer,data$`Jitter(Abs)`,data$HNR,d
ata$NHR,data$DFA,data$total_UPDRS)
pairs(muldata)
cor(muldata)

VIF(lm(test_time~PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA,data))
VIF(lm(PPE~test_time+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA,data))
```

```
VIF(lm(RPDE~PPE+test_time+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA,data))
VIF(lm(Shimmer~PPE+RPDE+test_time+`Jitter(Abs)`+HNR+NHR+DFA,data))
VIF(lm(`Jitter(Abs)`~PPE+RPDE+Shimmer+test_time+HNR+NHR+DFA,data))
VIF(lm(HNR~PPE+RPDE+Shimmer+`Jitter(Abs)`+test_time+NHR+DFA,data))
VIF(lm(NHR~PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+test_time+DFA,data))
VIF(lm(DFA~PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+test_time,data))

#WLS to try to improve model
wts1=1/fitted(lm(abs(residuals(model))~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NH
R+DFA, data))^2
model2 <- lm(total_UPDRS~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA,
weight = wts1, data)
summary(model2)

wts2=1/fitted(lm(abs(residuals(model2))~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+N
HR+DFA, data))^2
model3 <- lm(total_UPDRS~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA,
weight = wts2, data)
summary(model3)

#transform Y
bcmle=boxcox(model,lambda=seq(-3,3,by=0.1))
lambda=bcmle$x[which.max(bcmle$y)]
lambda
data$yt <- data$total_UPDRS^lambda
model <- lm(yt~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA, data)

#assumption testing again for constant variance, normality and linearity
data$group <- cut(data$total_UPDRS, 5)
data$residual <- model3$residuals
bf.test(residual ~ group, data)

shapiro.test(residuals(model3)[1:5000])
qqnorm(residuals(model))
qqline(residuals(model))

plot(data$test_time+data$PPE+data$RPDE+data$Shimmer+data$`Jitter(Abs)`+data$HNR+data
$NHR+data$DFA, residuals(model), ylab="residuals", xlab="fitted values")
abline(0,0)


#Selecting the best model
plotmodel.s(data[,2:9], data$total_UPDRS)
BestSub(data[,2:9], data$total_UPDRS, num = 1)
```

```
anova(model)

#check for influential points
x <- dffits(lm(total_UPDRS~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA, data))
x[which(x>1)]

cooks.distance(lm(total_UPDRS~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA,
data))
qf(0.2,8,5875)
qf(0.5,8,5875)

d<-dfbetas(lm(total_UPDRS~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA,
data))
d
which(abs(d[, 1])>1 & abs(d[, 2])>1 & abs(d[,3])>1& abs(d[,4])>1& abs(d[,5])>1& abs(d[,6])>1&
abs(d[,7])>1& abs(d[,8])>1)
d[which(abs(d[, 2])>1 & abs(d[,3])>1& abs(d[,4])>1& abs(d[,5])>1& abs(d[,6])>1& abs(d[,7])>1&
abs(d[,8])>1), ]
```

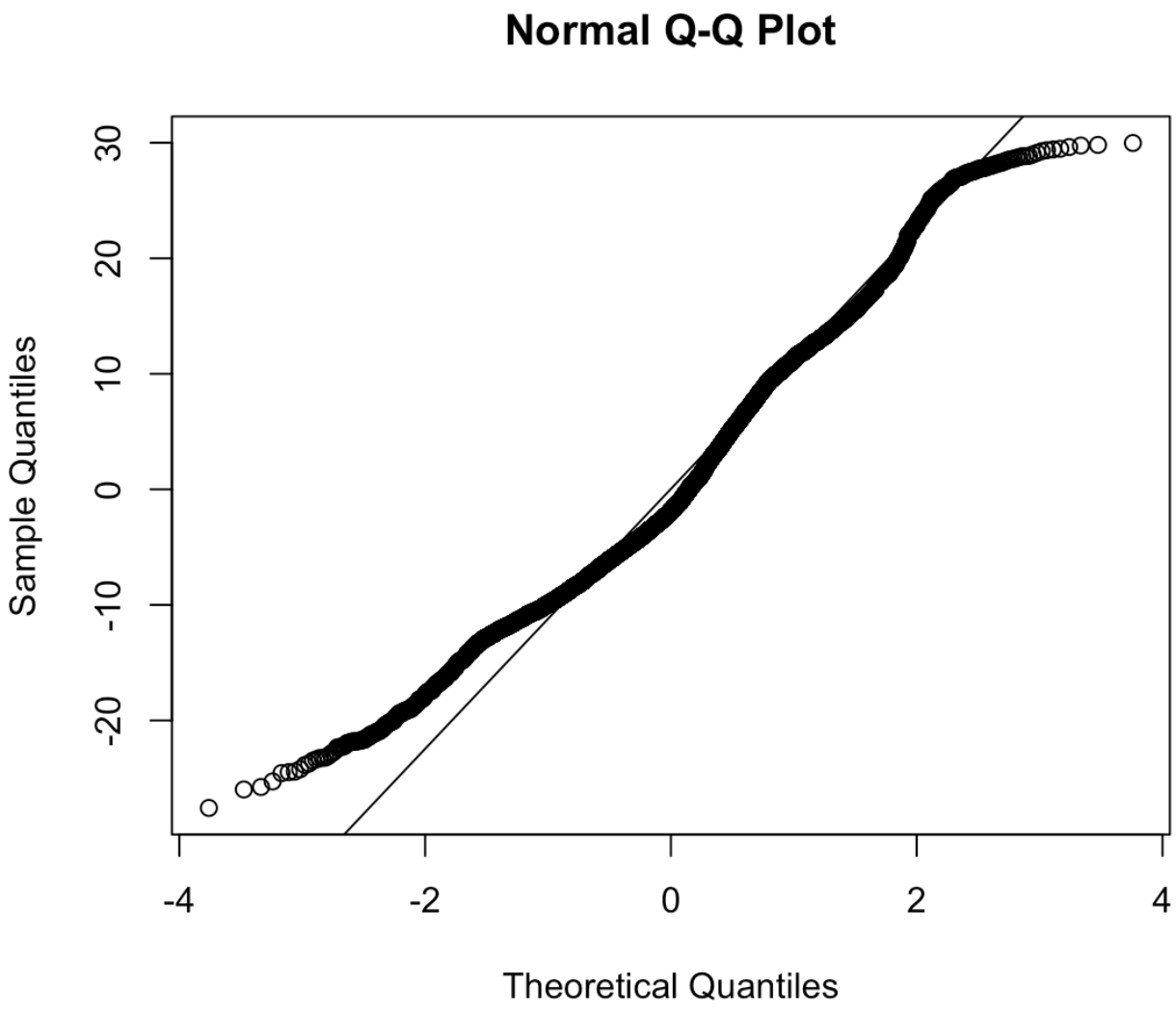
code outputs

Build the initial model with all the continuous predictors included:

```
> model <- lm(total_UPDRS~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA, data)
> summary(model)

Call:
lm(formula = total_UPDRS ~ test_time + PPE + RPDE + Shimmer + 
    `Jitter(Abs)` + HNR + NHR + DFA, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-27.574  -7.546  -1.889   7.628  29.977 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    5.790e+01  2.982e+00  19.415  < 2e-16 ***
test_time      1.686e-02  2.488e-03   6.777 1.34e-11 ***
PPE            2.651e+01  2.730e+00   9.712  < 2e-16 ***
RPDE           6.301e+00  1.811e+00   3.479 0.000506 ***
Shimmer       -1.561e+01  1.050e+01  -1.487 0.136942    
`Jitter(Abs)` -1.944e+04  7.340e+03  -2.648 0.008112 ** 
HNR           -5.547e-01  6.965e-02  -7.964 1.99e-15 ***
NHR           -3.079e+01  4.423e+00  -6.963 3.70e-12 ***
DFA           -3.869e+01  2.294e+00 -16.868  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.17 on 5866 degrees of freedom
Multiple R-squared:  0.09791,   Adjusted R-squared:  0.09668 
F-statistic: 79.59 on 8 and 5866 DF,  p-value: < 2.2e-16
```

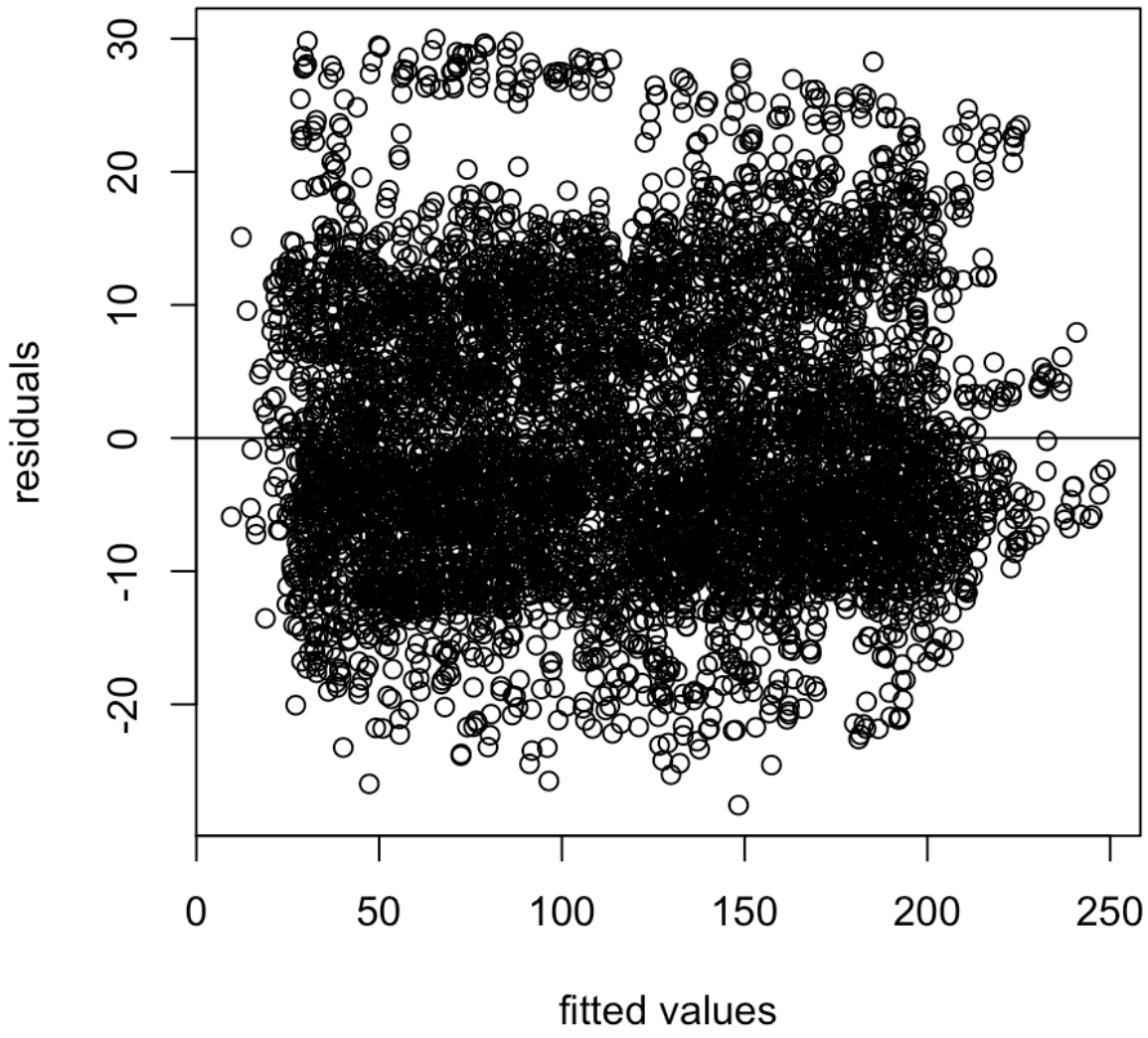
Check for constant variance using the bf test, check for the normality using the Shapiro wilk test:

```
> #constant variance
> data$group <- cut(data$total_UPDRS, 3)
> data$residual <- model$residuals
> bf.test(residual ~ group, data)

  Brown-Forsythe Test (alpha = 0.05)
------------------------------------------------------------
  data : residual and group

  statistic  : 8540.979
  num df     : 2
  denom df   : 4238.972
  p.value    : 0

  Result     : Difference is statistically significant.
------------------------------------------------------------
> #normality
> shapiro.test(residuals(model)[1:5000])

        Shapiro-Wilk normality test

data:  residuals(model)[1:5000]
W = 0.97272, p-value < 2.2e-16

> qqnorm(residuals(model))
> qqline(residuals(model))
```


Normal Q-Q Plot
Sample Quantiles
30
20
10
0
-10
-20
-4
-2
0
2
4
Theoretical Quantiles

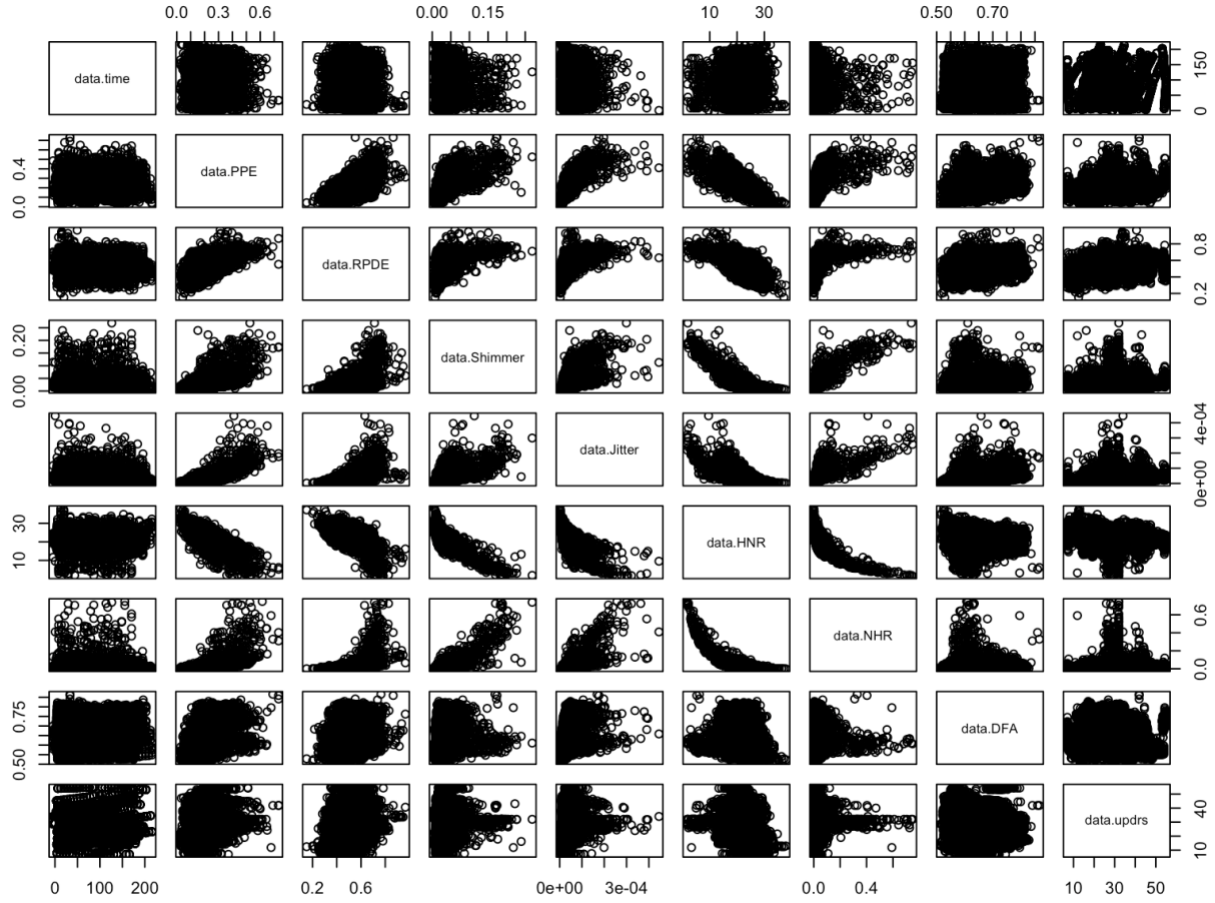
Plot the residuals to check for linearity:

```
> plot(data$test_time+data$PPE+data$RPDE+data$Shimmer+data$`Jitter(Abs)`+data$HNR+
+data$DFA, residuals(model), ylab="residuals", xlab="fitted values")
> abline(0,0)
```



Check for multicollinearity violations using the correlation plots and the variance inflation values.

```
> muldata <- data.frame(data$test_time,data$PPE,data$RPDE,data$Shimmer,data$`Jitter(Abs)`,
data$HNR,data$NHR,data$DFA,data$total_UPDRS)
> pairs(muldata)
> cor(muldata)
```



```
> VIF(lm(test_time~PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA,data))
[1] 1.004216
> VIF(lm(PPE~test_time+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA,data))
[1] 3.542713
> VIF(lm(RPDE~PPE+test_time+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA,data))
[1] 1.899211
> VIF(lm(Shimmer~PPE+RPDE+test_time+`Jitter(Abs)`+HNR+NHR+DFA,data))
[1] 4.176083
> VIF(lm(`Jitter(Abs)`~PPE+RPDE+Shimmer+test_time+HNR+NHR+DFA,data))
[1] 3.962207
> VIF(lm(HNR~PPE+RPDE+Shimmer+`Jitter(Abs)`+test_time+NHR+DFA,data))
[1] 5.073821
> VIF(lm(NHR~PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+test_time+DFA,data))
[1] 3.958299
> VIF(lm(DFA~PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+test_time,data))
[1] 1.502271
>
```

Since there is constant variance, do the weighted least squares method with two iterations to see if the assumptions have improved:

```
> wts1=1/fitted(lm(abs(residuals(model))~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+
DFA, data))^2
> model2 <- lm(total_UPDRS~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA, weight =
 wts1, data)
> summary(model2)

Call:
lm(formula = total_UPDRS ~ test_time + PPE + RPDE + Shimmer +
    `Jitter(Abs)` + HNR + NHR + DFA, data = data, weights = wts1)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-3.6007 -0.9072 -0.2394  0.9083  3.5069

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.927e+01  2.833e+00  20.922  < 2e-16 ***
test_time      1.523e-02  2.418e-03   6.298 3.24e-10 ***
PPE            1.719e+01  2.175e+00   7.904 3.20e-15 ***
RPDE           6.195e+00  1.761e+00   3.518 0.000439 ***
Shimmer       -2.133e+01  9.002e+00  -2.369 0.017849 *
`Jitter(Abs)` -4.701e+03  5.107e+03  -0.920 0.357417
HNR           -5.939e-01  6.599e-02  -9.000  < 2e-16 ***
NHR           -2.258e+01  2.685e+00  -8.411  < 2e-16 ***
DFA           -3.717e+01  2.208e+00 -16.835  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.207 on 5866 degrees of freedom
Multiple R-squared:  0.09647,   Adjusted R-squared:  0.09524
F-statistic: 78.29 on 8 and 5866 DF,  p-value: < 2.2e-16
```

```
> wts2=1/fitted(lm(abs(residuals(model2))~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR
+DFA, data))^2
> model3 <- lm(total_UPDRS~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA, weight =
 wts2, data)
> summary(model3)

Call:
lm(formula = total_UPDRS ~ test_time + PPE + RPDE + Shimmer +
    `Jitter(Abs)` + HNR + NHR + DFA, data = data, weights = wts2)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-4.5494 -0.9192 -0.2316  0.9136  3.4977

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.922e+01  2.513e+00  23.566  < 2e-16 ***
test_time      1.455e-02  2.205e-03   6.595 4.62e-11 ***
PPE            8.502e+00  1.502e+00   5.660 1.59e-08 ***
RPDE           5.473e+00  1.627e+00   3.365 0.000771 ***
Shimmer       -8.533e+00  5.639e+00  -1.513 0.130261
`Jitter(Abs)`  4.086e+03  3.496e+03   1.169 0.242587
HNR           -5.963e-01  5.495e-02 -10.851  < 2e-16 ***
NHR           -2.067e+01  1.738e+00 -11.892  < 2e-16 ***
DFA           -3.478e+01  2.101e+00 -16.555  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.215 on 5866 degrees of freedom
Multiple R-squared:  0.1024,    Adjusted R-squared:  0.1012
F-statistic: 83.66 on 8 and 5866 DF,  p-value: < 2.2e-16
```

Transform the Y to try to improve the assumptions:

```
> #transform Y
> bcmle=boxcox(model,lambda=seq(-3,3,by=0.1))
> lambda=bcmle$x[which.max(bcmle$y)]
> lambda
[1] 0.6363636
> data$yt <- data$total_UPDRS^lambda
> model <- lm(yt~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA, data)
```

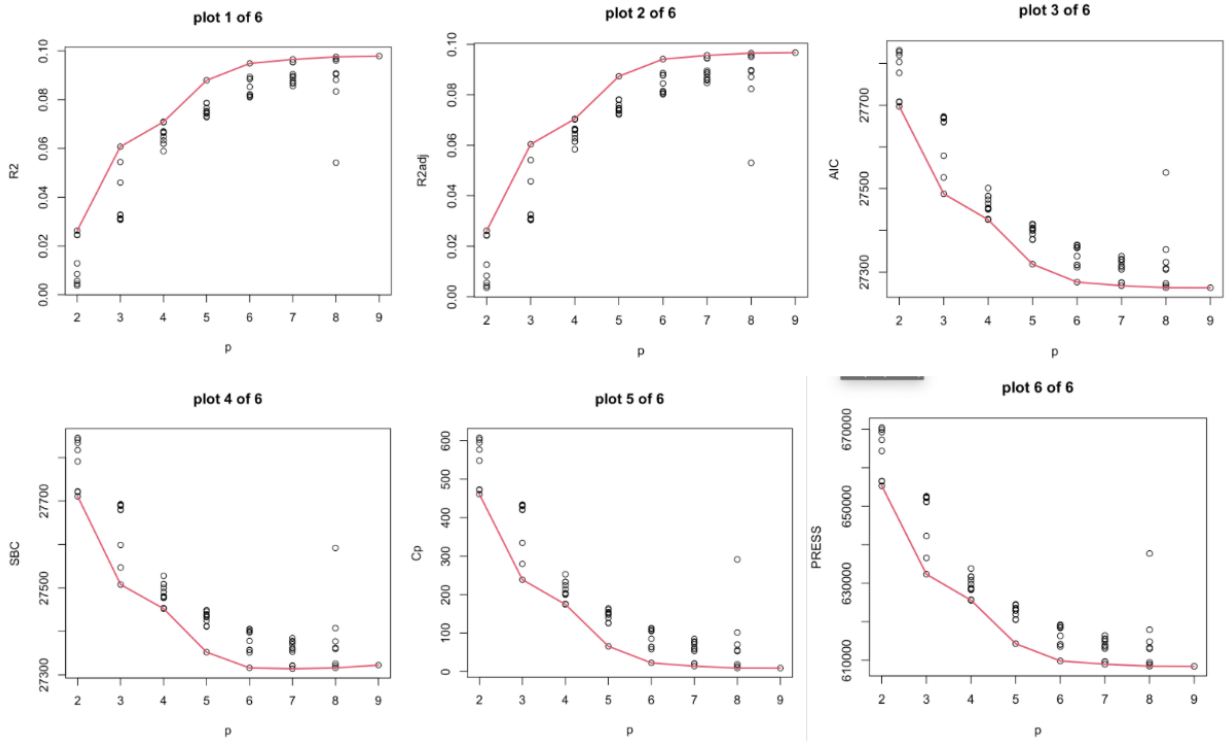
Do the assumption testing again for the constant variance and normality to see if there is any changes:

```
> #assumption testing again for constant variance, normality and linearity
> data$group <- cut(data$total_UPDRS, 5)
> data$residual <- model3$residuals
> bf.test(residual ~ group, data)

  Brown-Forsythe Test (alpha = 0.05)
------------------------------------------------------------
  data : residual and group

  statistic  : 7402.528
  num df     : 4
  denom df   : 2103.066
  p.value    : 0

  Result     : Difference is statistically significant.
------------------------------------------------------------
> #assumption testing again for constant variance, normality and linearity
> data$group <- cut(data$total_UPDRS, 5)
> data$residual <- model3$residuals
> bf.test(residual ~ group, data)

  Brown-Forsythe Test (alpha = 0.05)
------------------------------------------------------------
  data : residual and group

  statistic  : 7402.528
  num df     : 4
  denom df   : 2103.066
  p.value    : 0

  Result     : Difference is statistically significant.
```

```
> shapiro.test(residuals(model3)[1:5000])

        Shapiro-Wilk normality test

data:  residuals(model3)[1:5000]
W = 0.97106, p-value < 2.2e-16

> qqnorm(residuals(model))
> qqline(residuals(model))
```

Use the 6 different model selection criteria and the best subset function to select the best model:

```
> plotmodel.s(data[,2:9], data$total_UPDRS)
```




```
> BestSub(data[,2:9], data$total_UPDRS, num = 1)
  p 1 2 3 4 5 6 7 8         SSEp r2 r2.adj           Cp      AICp      SBCp       PRESSp
1 2 0 0 0 0 1 0 0 0 3.413266e-24  1      1 28647.058442 -368433.9 -368420.5 3.414781e-24
2 3 0 0 0 1 1 0 0 0 2.046389e-24  1      1  1870.230703 -371437.5 -371417.5 2.048148e-24
3 4 0 1 0 1 1 0 0 0 2.027574e-25  1      1   464.129453 -385017.5 -384990.7 2.029785e-25
4 5 1 1 0 1 1 0 0 0 6.157335e-25  1      1    90.712651 -378489.5 -378456.1 6.164968e-25
5 6 1 1 0 1 1 0 1 0 6.156681e-25  1      1    44.535571 -378488.1 -378448.0 6.165692e-25
6 7 1 1 1 1 1 0 1 0 3.268703e-25  1      1    17.102944 -382205.8 -382159.1 3.274491e-25
7 8 1 1 1 1 1 1 1 0 3.268670e-25  1      1     7.000712 -382203.9 -382150.5 3.274522e-25
8 9 1 1 1 1 1 1 1 1 3.268570e-25  1      1     9.000000 -382202.1 -382142.0 3.274681e-25
```

Check for influential points in the model:

```
> #check for influential points
> x <- dffits(lm(total_UPDRS~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA, data))
> x[which(x>1)]
named numeric(0)
> cook <- cooks.distance(lm(total_UPDRS~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA, data))
> twenty <- qf(0.2,8,5875)
> fifty <- qf(0.5,8,5875)
> cook[which(cook>fifty)]
named numeric(0)
> d<-dfbetas(lm(total_UPDRS~test_time+PPE+RPDE+Shimmer+`Jitter(Abs)`+HNR+NHR+DFA, data))
> which(abs(d[, 1])>1 & abs(d[, 2])>1 & abs(d[,3])>1& abs(d[,4])>1& abs(d[,5])>1& abs(d[,6])>1& abs(d[,7])>1&
 abs(d[,8])>1)
named integer(0)
> d[which(abs(d[, 2])>1 & abs(d[,3])>1& abs(d[,4])>1& abs(d[,5])>1& abs(d[,6])>1& abs(d[,7])>1& abs(d[,8])>1),
 ]
     (Intercept) test_time PPE RPDE Shimmer `Jitter(Abs)` HNR NHR DFA
```