Predictive Analyses of Reaction Time for EEG

Lab Report - Summer 2019

Thanasi Bakis

Retrieving the Data

All subjects' data are processed with signal filters, averaged across channels using principal component analysis (twice, to examine the N200 and P300), and given to us in a Matlab matrix form. To faciliate the use of R's plotting and modeling functions, we convert each subject's data to R data frame form. From each subject data file, three data frames are created:

1. `N200.Data`, which contains the samples of all trials for the subject, averaged across channels with PCA (to examine for N200 activity). Each row is a sample, with columns `Trial`, `Time.ms`, and `Sample.Val`.
2. `P300.Data`, which contains the samples of all trials for the subject, averaged across channels with PCA (to examine for P300 activity). Each row is a sample, with columns `Trial`, `Time.ms`, and `Sample.Val`.
3. `Info`, which contains non-sampled data about all trials for the subject. Each row is a trial, with columns `Reaction.Time.ms`, `Condition`, and `Correct`. This information is stored here in a separate data frame, instead of the sample data frame, since it is constant across all samples in the trial and would be repeated many times in that data frame.

Once each subject's data is converted, the data is merged across subjects 112 (both sessions), 116 (session 1), and 143 (session 1), to provide sufficient data to the model. Other subjects and sessions may be used, if desired. The resulting `eeg` data structure is a list containing the three data frames (`N200.Data`, `P300.Data`, and `Info`), each merged across all the given subjects and sessions. Trial numbers are incremented across subjects to ensure uniqueness.

Note this procedure can take time initially. Upon the first run, an intermediate form of the data will be saved (the conversion of the Matlab matrix to an R matrix), as well as the final output (the desired R data frame). This allows subsequent function calls to avoid part or all of the conversion. If the original data is changed and the procedure needs to be re-run from scratch, these intermediate files should be deleted.

```
root = "/share/data/michael/exp7data/subjects"

eeg = read.sessions("/share/data/michael/exp7data/subjects/s112/ses1/s112_ses1_sfinal.mat",
                    "/share/data/michael/exp7data/subjects/s112/ses2/s112_ses2_sfinal.mat",
                    "/share/data/michael/exp7data/subjects/s116/ses1/s116_ses1_sfinal.mat",
                    "/share/data/michael/exp7data/subjects/s143/ses1/s143_ses1_sfinal.mat")
```

Locating the Peaks

Given the data frames, we attempt to select trials that are suitable to include in the model predicting reaction time from peak locations. To be included in this `features` data frame, a trial must include at least one of the two peaks, as well as have a reaction time of at least 350 ms. Notable columns in this data frame include:

1. `Trial`, the unique trial number,
2. `Time.ms.N200`, the time (in ms) of the located N200 peak,
3. `Sample.Val.N200`, the sample value at the time of the N200,
4. `Time.ms.P300`, the time (in ms) of the located P300 peak,
5. `Sample.Val.P300`, the sample value at the time of the P300,

6. `Reaction.Time.ms`, the subject's reaction time in the given trial,
7. `Condition`, the condition ID number of the given trial, and
8. `Correct`, a binary indicator of the subject's correctness in the given trial.

Other columns were used in the peak locating process and may be of interest. These include:

9. `Range.Left.N200`, the time (in ms) of the "start" of the N200 peak (determined by the inflection point left of the curve's peak),
10. `Range.Right.N200`, the time (in ms) of the "end" of the N200 peak (determined by the inflection point right of the curve's peak),
11. `Derivative.Left.N200`, the average derivative between the start of the N200 and the N200 peak,
12. `Derivative.Right.N200`, the average derivative between the N200 peak and the end of the N200,
13. `Range.Left.P300`, the time (in ms) of the "start" of the P300 peak (determined by the inflection point left of the curve's peak),
14. `Range.Right.P300`, the time (in ms) of the "end" of the P300 peak (determined by the inflection point right of the curve's peak),
15. `Derivative.Left.P300`, the average derivative between the start of the P300 and the P300 peak, and
16. `Derivative.Right.P300`, the average derivative between the P300 peak and the end of the P300.

Note that this peak location process can take time.

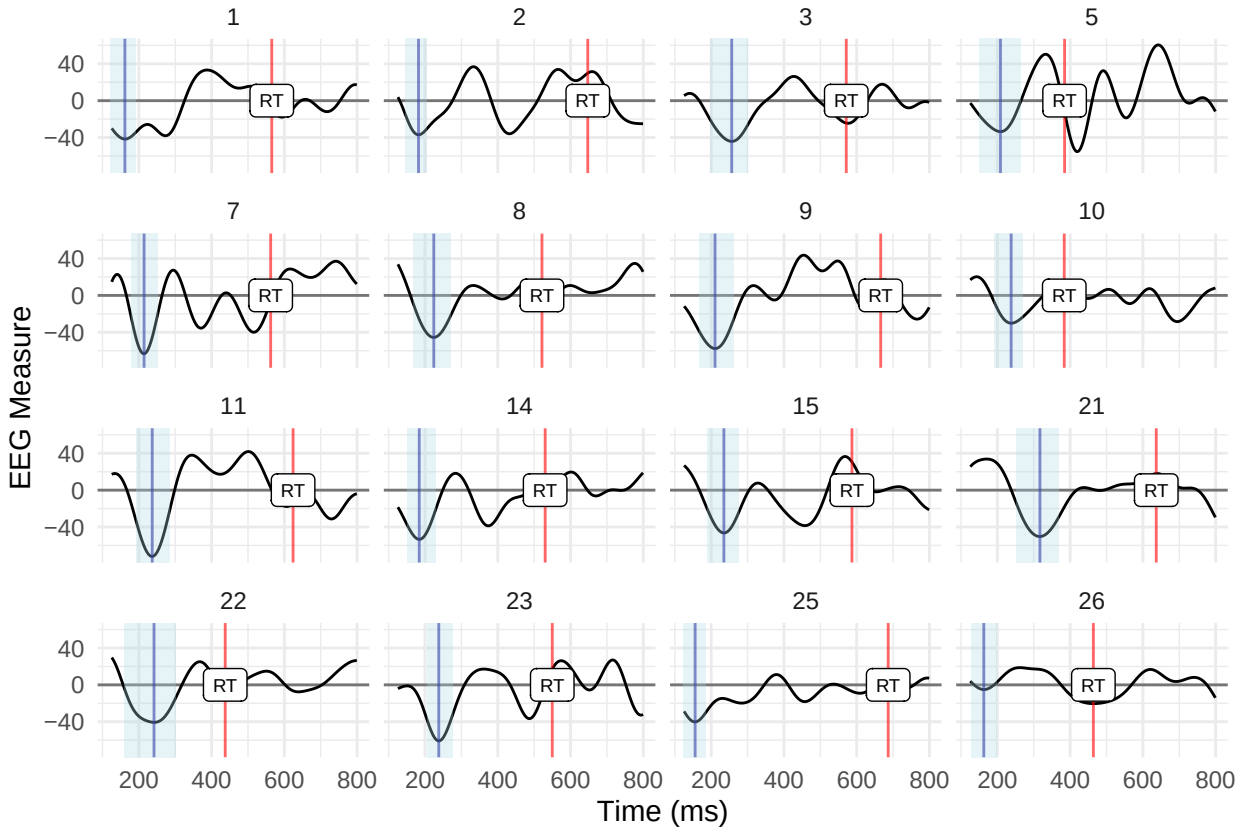
```
features = generate.features(eeg) %>%
  filter(Reaction.Time.ms >= 350)
```

Visualizing the Peaks

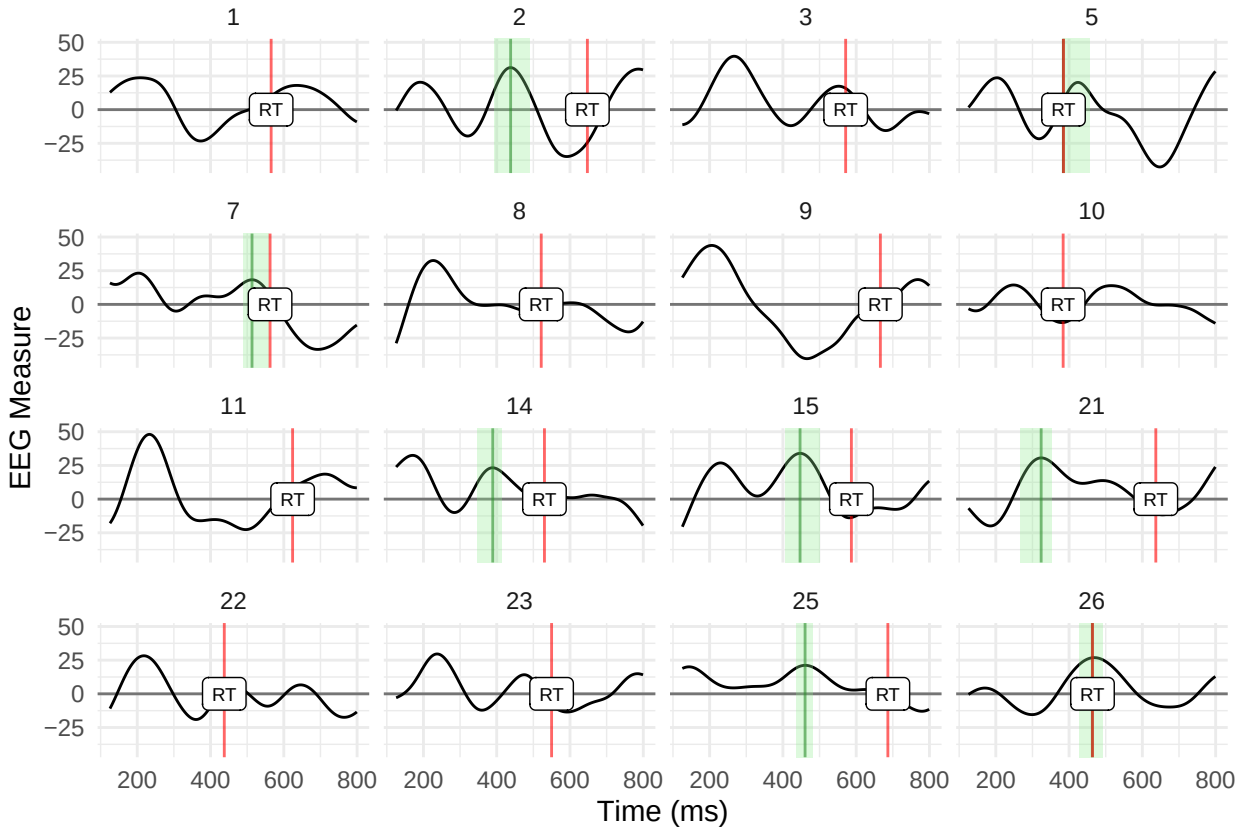
We draw a few trials to see where the N200 and P300 peaks were detected, to verify the algorithm works as desired. The N200 peaks are highlighted blue, and the P300 peaks are green.

```
some.trials = features %>%
  pull(Trial) %>%
  head(16)

eeg$N200.Data %>%
  visualize.trials(some.trials, features, N200 = T, RT = T)
```

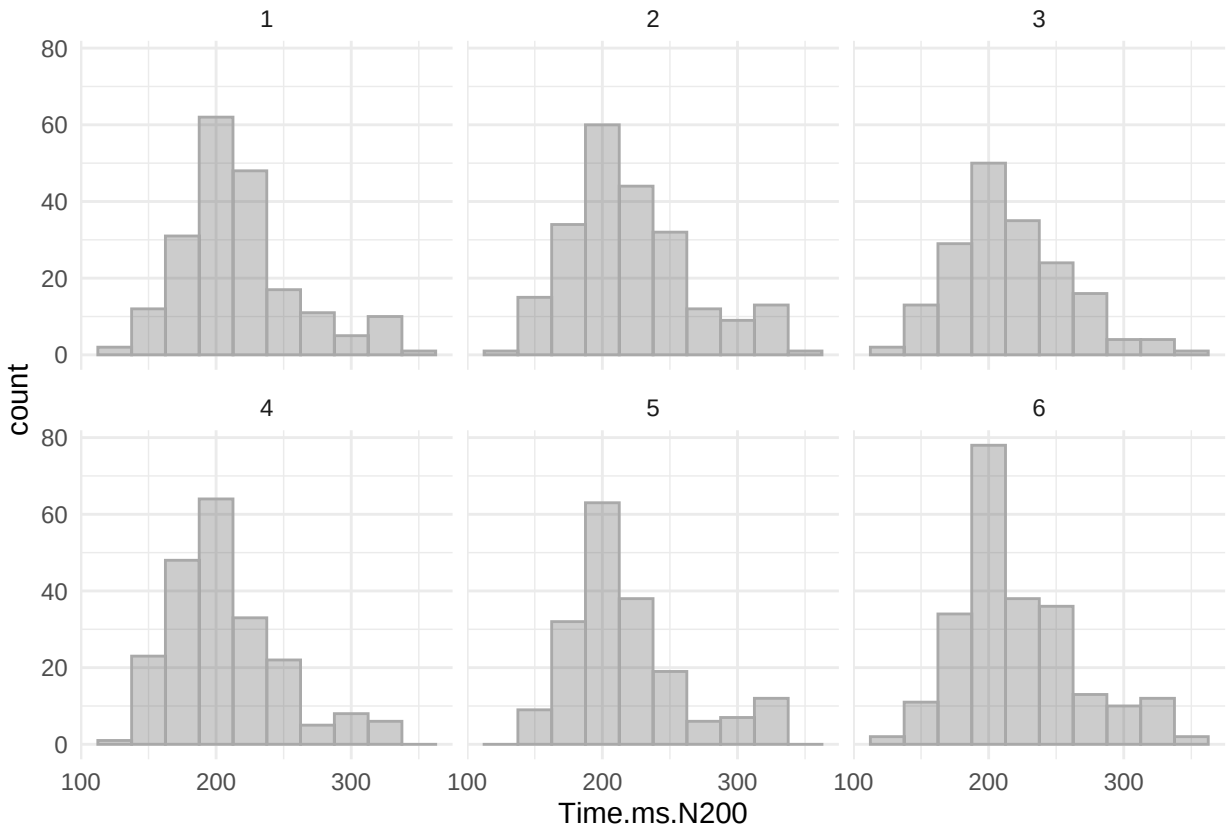



```
eeg$P300.Data %>%
  visualize.trials(some.trials, features, P300 = T, RT = T)
```

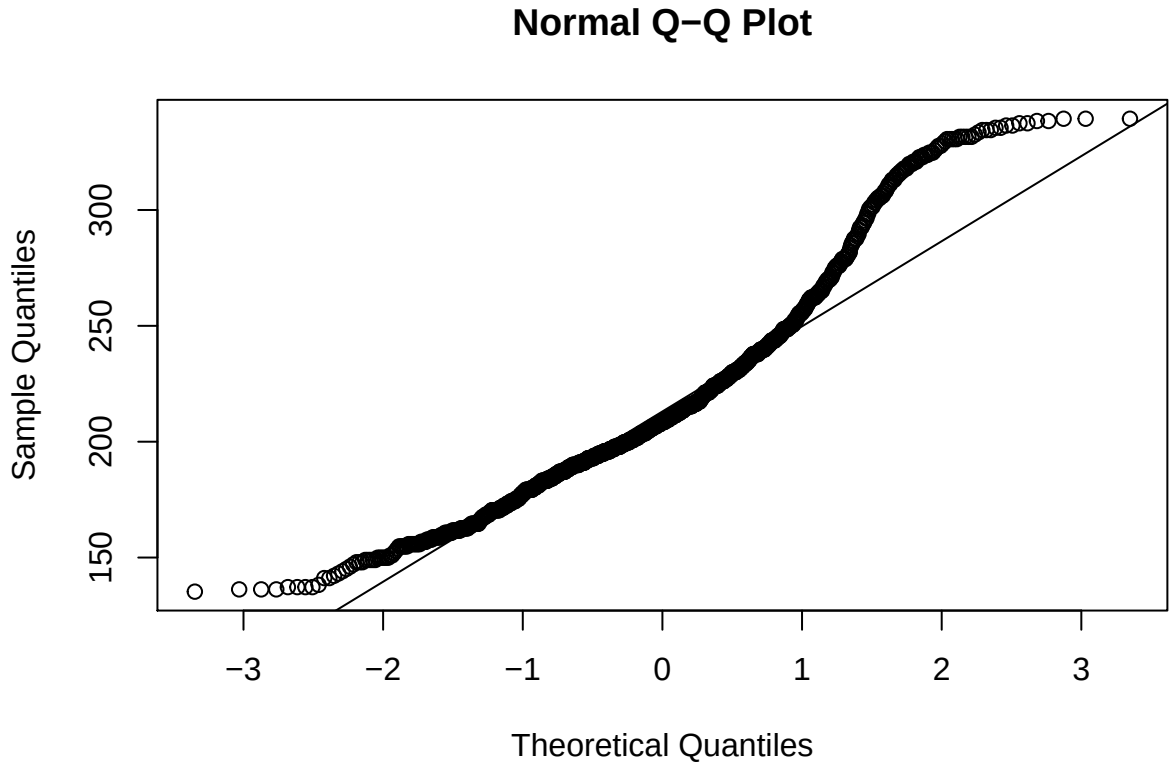


We also plot histograms of the peak times to check that they are distributed approximately around 200 and 300 ms.

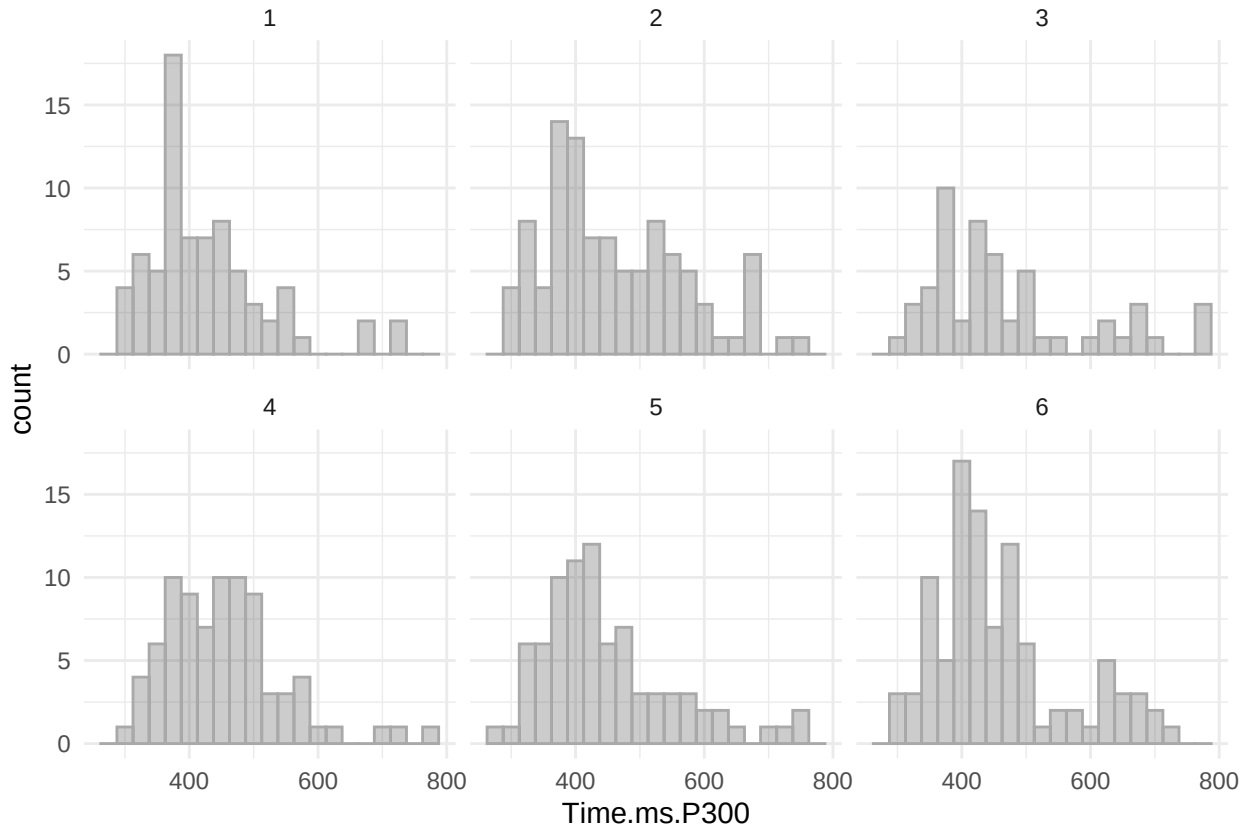
```
features.hist(features, Time.ms.N200)
```



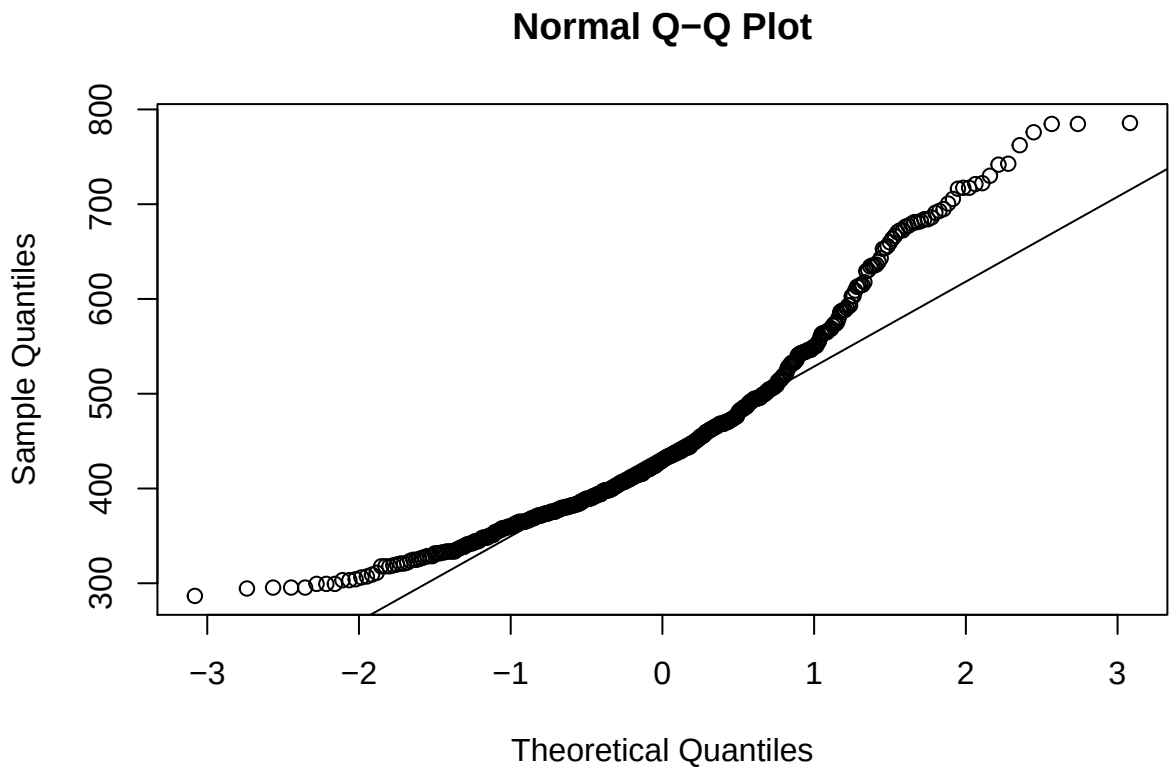

```
qqnorm(features$Time.ms.N200); qqline(features$Time.ms.N200)
```



```
features.hist(features, Time.ms.P300)
```




```
qqnorm(features$Time.ms.P300); qqline(features$Time.ms.P300)
```



Constructing the Models

Our primary interest is to examine the relationship between the times of the N200 and P300 peaks, and the subjects' reaction times to the trials. In order to remove the correlation between the two peaks, the absolute P300 time is replaced with the offset of the P300 from the N200 time.

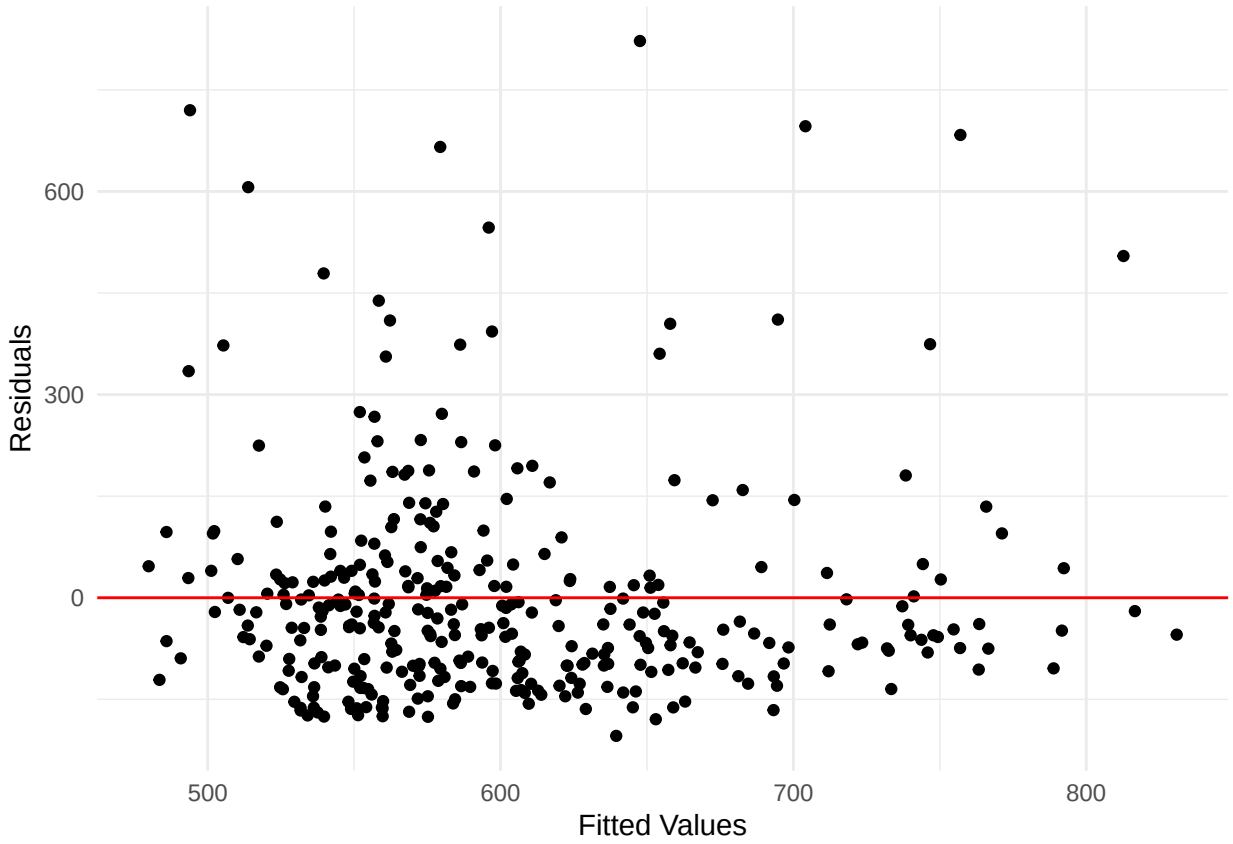
We first directly model the peaks with the reaction times. We find that the times of both peaks are highly significant predictors, although the R-squared is found to be 0.1601.

```
features = mutate(features, P300.Offset.ms = Time.ms.P300 - Time.ms.N200)
# eliminates correlation between N200 and P300 values

model = lm(Reaction.Time.ms ~ Time.ms.N200 + P300.Offset.ms, data = features)
summary(model)
```

```
##
## Call:
## lm(formula = Reaction.Time.ms ~ Time.ms.N200 + P300.Offset.ms,
##     data = features)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -203.95 -100.45  -40.51   37.17  822.13
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    202.92134   60.78575   3.338 0.000936 ***
## Time.ms.N200     1.10271    0.22237   4.959 1.12e-06 ***
## P300.Offset.ms   0.66041    0.08412   7.851 5.40e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 161.2 on 341 degrees of freedom
##   (1028 observations deleted due to missingness)
## Multiple R-squared:  0.1601, Adjusted R-squared:  0.1552
## F-statistic: 32.51 on 2 and 341 DF,  p-value: 1.193e-13
```

```
ggplot(model, aes(.fitted, .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "red") +
  theme_minimal() +
  xlab("Fitted Values") +
  ylab("Residuals")
```



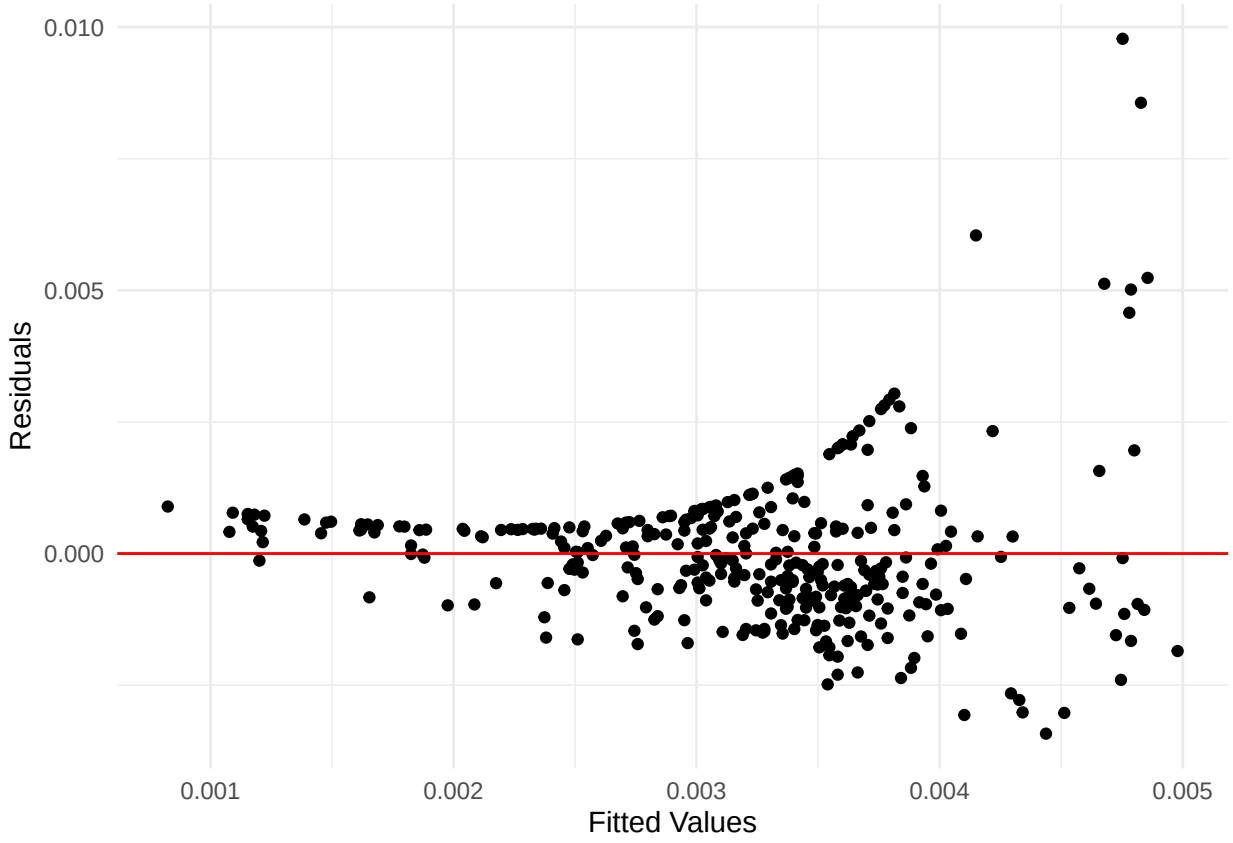
We also are interested in seeing if the P300 contains any information about the speed of processing, which is inversely proportional to the reaction time. We once again find the P300 to be highly significant, though with a R-squared of 0.2351.

```
features = mutate(features, Inverse.RT.ms = 1 / (Reaction.Time.ms - Time.ms.N200))

model = lm(Inverse.RT.ms ~ P300.Offset.ms, data = features)
summary(model)
```

```
##
## Call:
## lm(formula = Inverse.RT.ms ~ P300.Offset.ms, data = features)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0034211 -0.0008510 -0.0000870  0.0005436  0.0097777
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.835e-03  1.766e-04   27.37   <2e-16 ***
## P300.Offset.ms -7.021e-06  6.847e-07  -10.25   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001445 on 342 degrees of freedom
##   (1028 observations deleted due to missingness)
## Multiple R-squared:  0.2351, Adjusted R-squared:  0.2329
## F-statistic: 105.1 on 1 and 342 DF,  p-value: < 2.2e-16
```

```
ggplot(model, aes(.fitted, .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "red") +
  theme_minimal() +
  xlab("Fitted Values") +
  ylab("Residuals")
```



Lastly, we are curious to see if the P300 information about speed is related to the subject's accuracy, although we fail to conclude evidence for this, seeing a large p-value for the P300.

```
glm(Correct ~ P300.Offset.ms, family = "binomial", data = features) %>%
  summary()
```

```
##
## Call:
## glm(formula = Correct ~ P300.Offset.ms, family = "binomial",
##     data = features)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5647  -1.5273   0.8549   0.8615   0.8762
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.7632974  0.2644221   2.887  0.00389 **
## P300.Offset.ms 0.0001971  0.0010289   0.192  0.84811
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 424.91  on 343  degrees of freedom
## Residual deviance: 424.87  on 342  degrees of freedom
##   (1028 observations deleted due to missingness)
## AIC: 428.87
##
## Number of Fisher Scoring iterations: 4
```