

# Portfolio 4, Study Group 10

Jesper Fischer, Kristian Severin, Lasse Hansen, Lærke Brædder, Sarah Hvid

03/04/2020

## Importing packages

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse

## v ggplot2 3.2.1      v purrr   0.3.3
## v tibble  2.1.3      v dplyr  0.8.4
## v tidyr   1.0.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0

## -- Conflicts ----- tidyverse
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

**1.a. How much does it have to be moved (indicate 3 translations and 3 rotations)?**

new mm = 47.5 55.1 31.3 and rot = 0.1, 0, -0.07

**1.b. Align the AC of the structural image to [0,0,0] with PC at [0,y,0]. How much does that have to be moved?**

we moved it to mm = 132.1 147.2 89.3, and no rotation.

**1.c. How far are the two images from being in register with one another?**

mm and vx subtracted from each other. 47.5 - 132.1 = -84.6 55.1 - 147.2 = -92.1 31.3 - 89.3 = -58

0.1 - 0 = 0.1 0 - 0 = 0 -0.07 - 0 = -0.07

**2.a-2.f Pictures of Realignment, Coregistration, Segmentation, Normalization, Smoothing:**

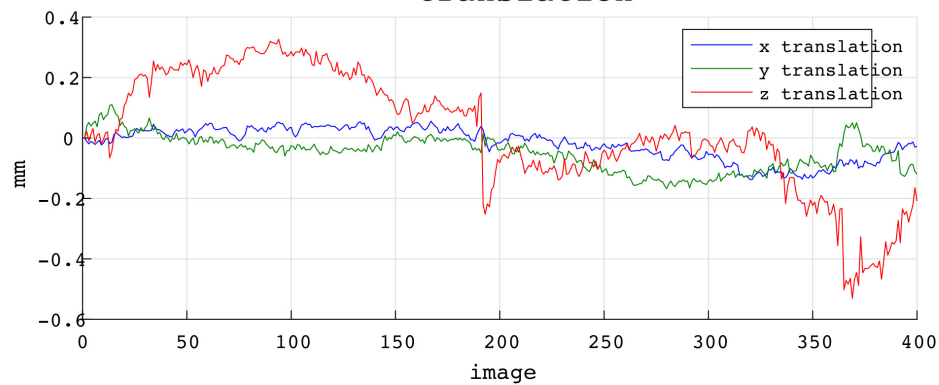
**Import the realignment parameters into R.**

```
Real <- read_table("rp_fSubjectNo0001-0005-00001-000001-01.txt", col_names = F)
```

## Image realignment

```
1 /Users/kristianmengel-niemann/Documents/2. Semester/ExpMeth2/Classes/fMRI
2 /Users/kristianmengel-niemann/Documents/2. Semester/ExpMeth2/Classes/fMRI
3 /Users/kristianmengel-niemann/Documents/2. Semester/ExpMeth2/Classes/fMRI
4 /Users/kristianmengel-niemann/Documents/2. Semester/ExpMeth2/Classes/fMRI
5 /Users/kristianmengel-niemann/Documents/2. Semester/ExpMeth2/Classes/fMRI
6 /Users/kristianmengel-niemann/Documents/2. Semester/ExpMeth2/Classes/fMRI
7 /Users/kristianmengel-niemann/Documents/2. Semester/ExpMeth2/Classes/fMRI
8 /Users/kristianmengel-niemann/Documents/2. Semester/ExpMeth2/Classes/fMRI
9 /Users/kristianmengel-niemann/Documents/2. Semester/ExpMeth2/Classes/fMRI
10 /Users/kristianmengel-niemann/Documents/2. Semester/ExpMeth2/Classes/fMRI
11 /Users/kristianmengel-niemann/Documents/2. Semester/ExpMeth2/Classes/fMRI
12 /Users/kristianmengel-niemann/Documents/2. Semester/ExpMeth2/Classes/fMRI
..... etc
```

### translation



### rotation

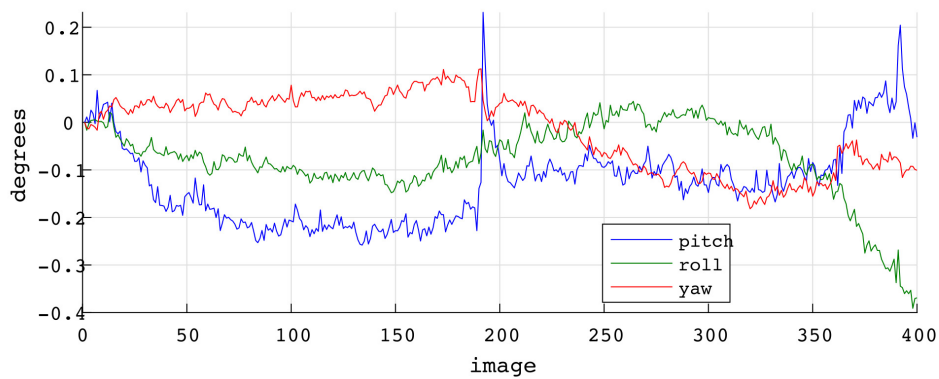


Figure 1: Realignment  
2

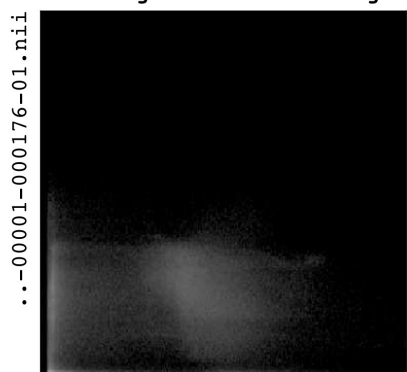
## Normalised Mutual Information Coregistration

$$X1 = 0.000*X + 0.000*Y - 0.500*Z + 92.876$$

$$Y1 = -0.488*X - 0.004*Y - 0.000*Z + 117.188$$

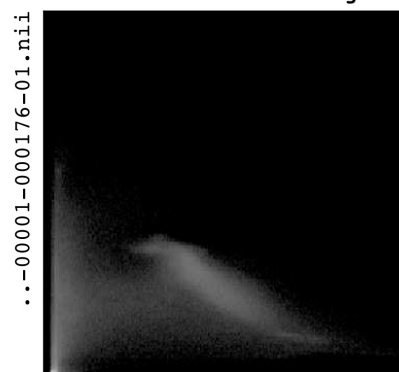
$$Z1 = -0.004*X + 0.488*Y + 0.000*Z - 40.261$$

Original Joint Histogram



..-00001-000001-01.nii

Final Joint Histogram



..-00001-000001-01.nii

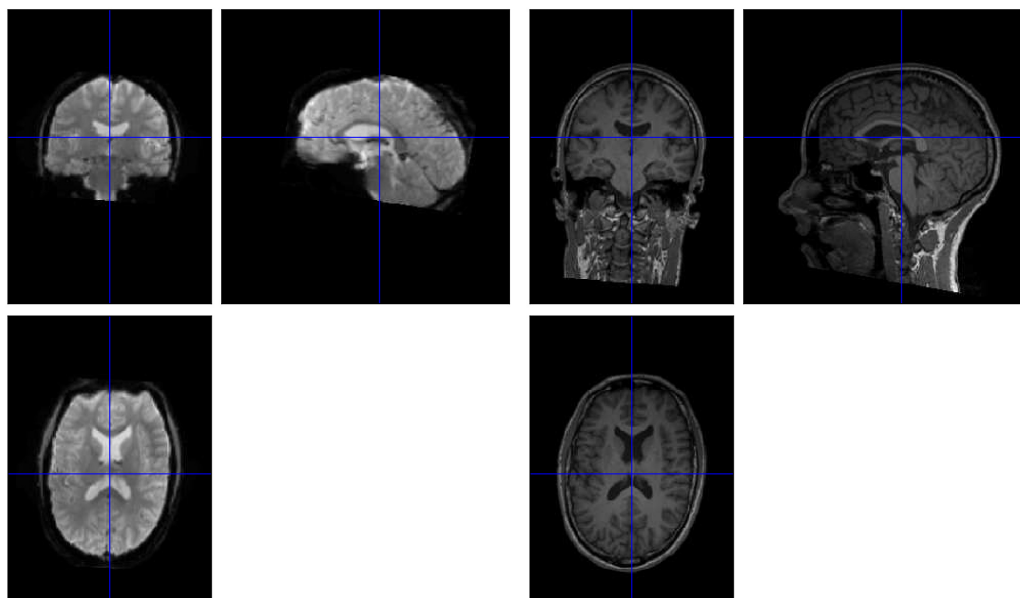


Figure 2: Realignment2

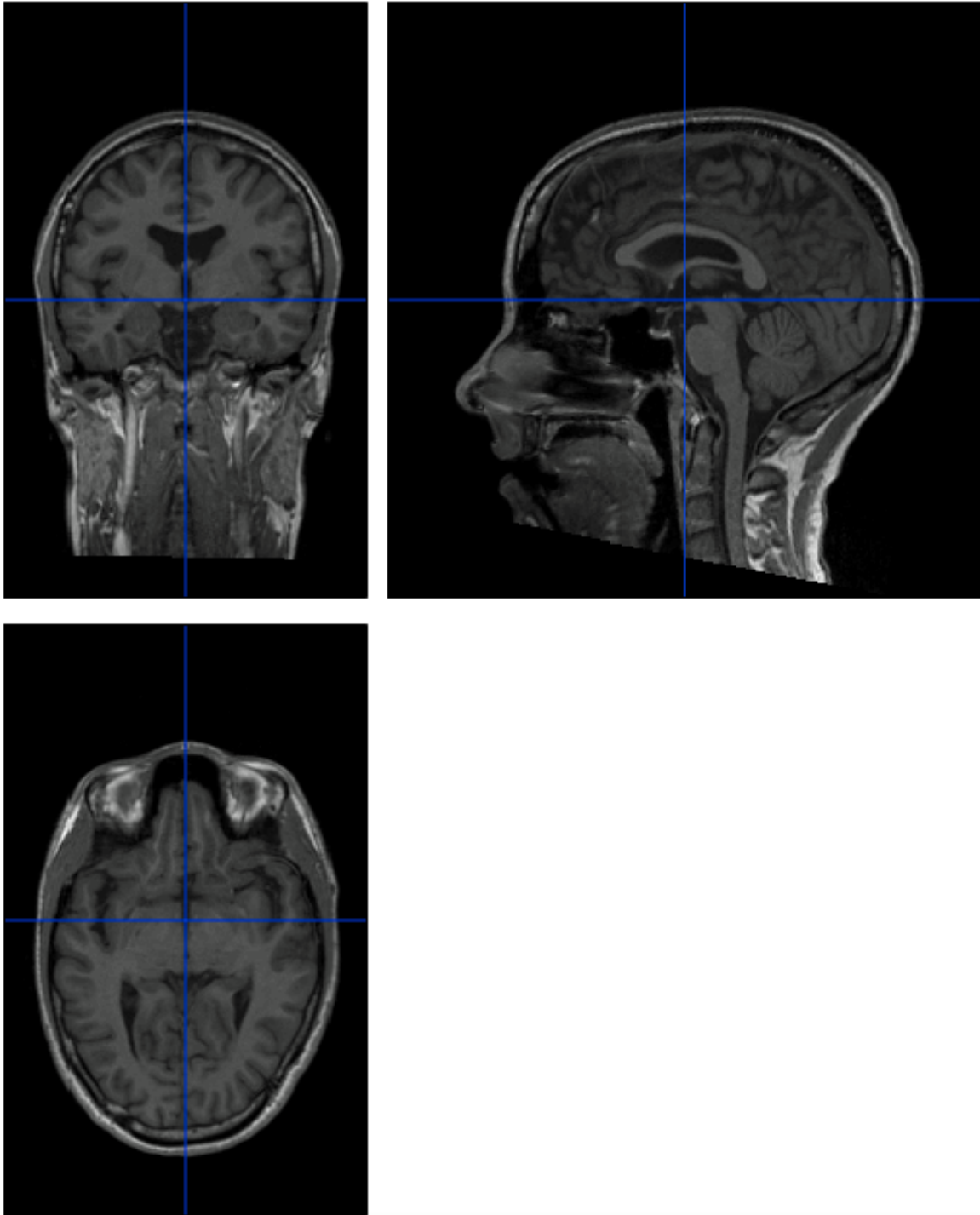


Figure 3: Functional Coregistration

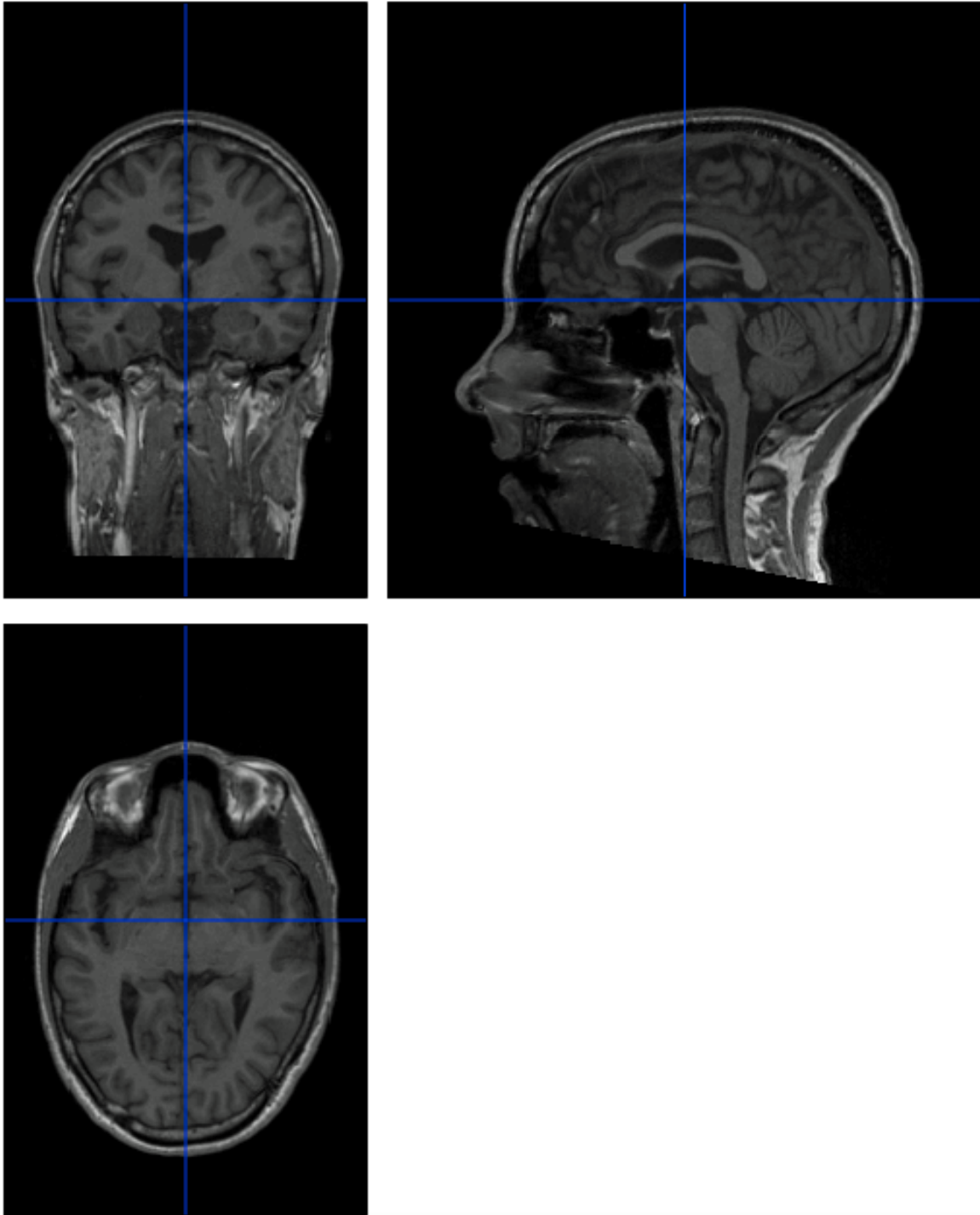


Figure 4: Structural Coregistration

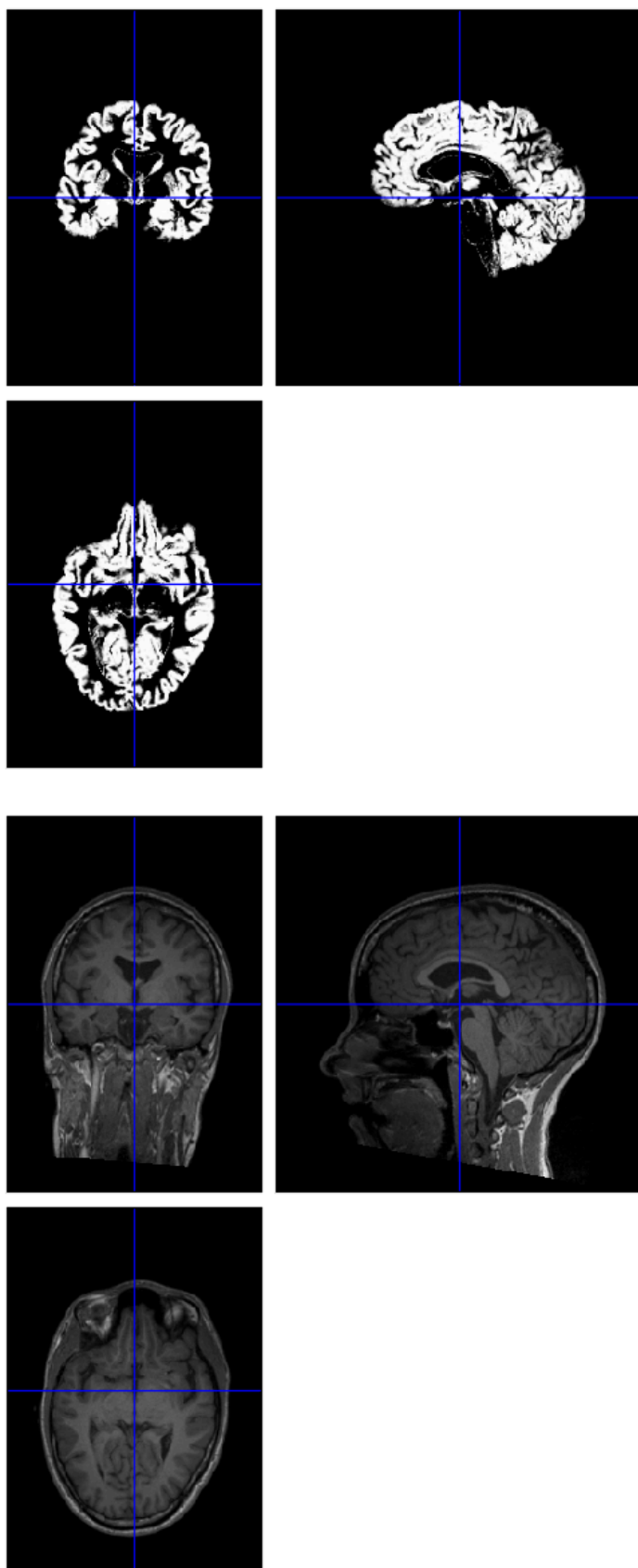


Figure 5: Segmentation

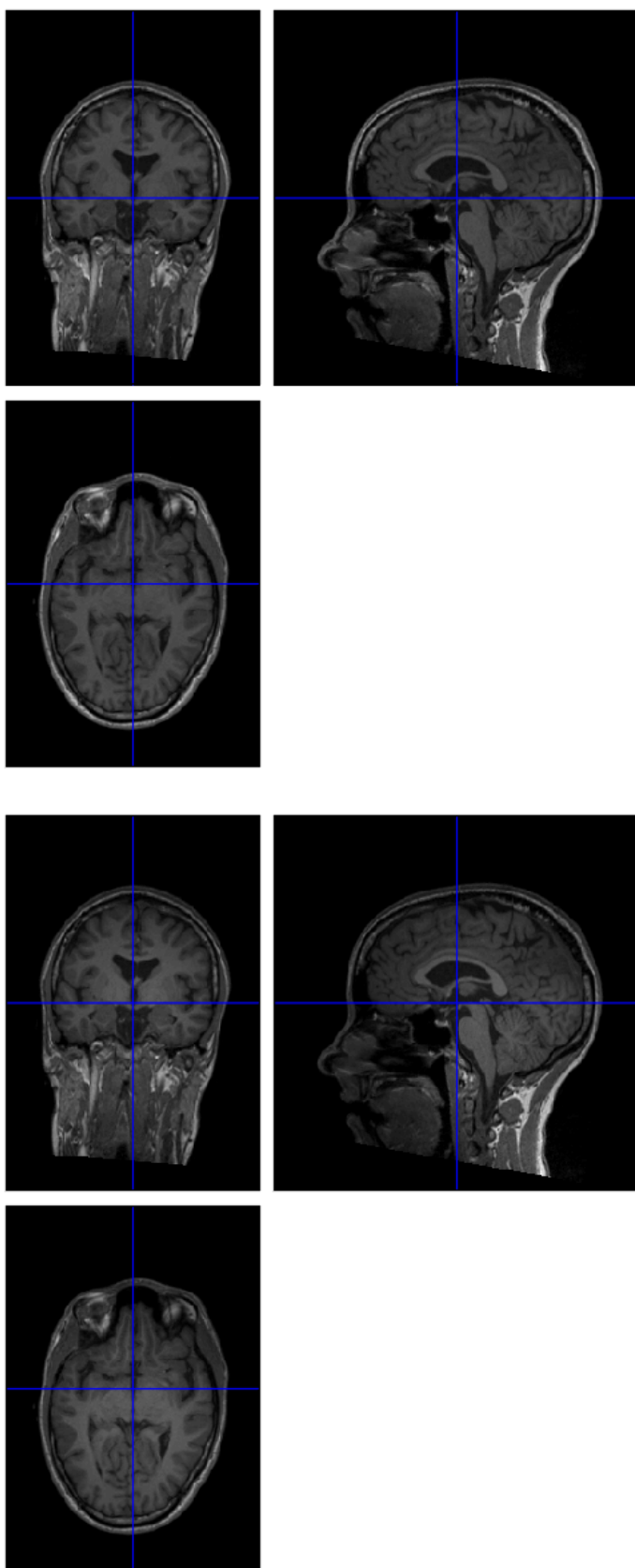


Figure 6: Bias-Correct Segmentation

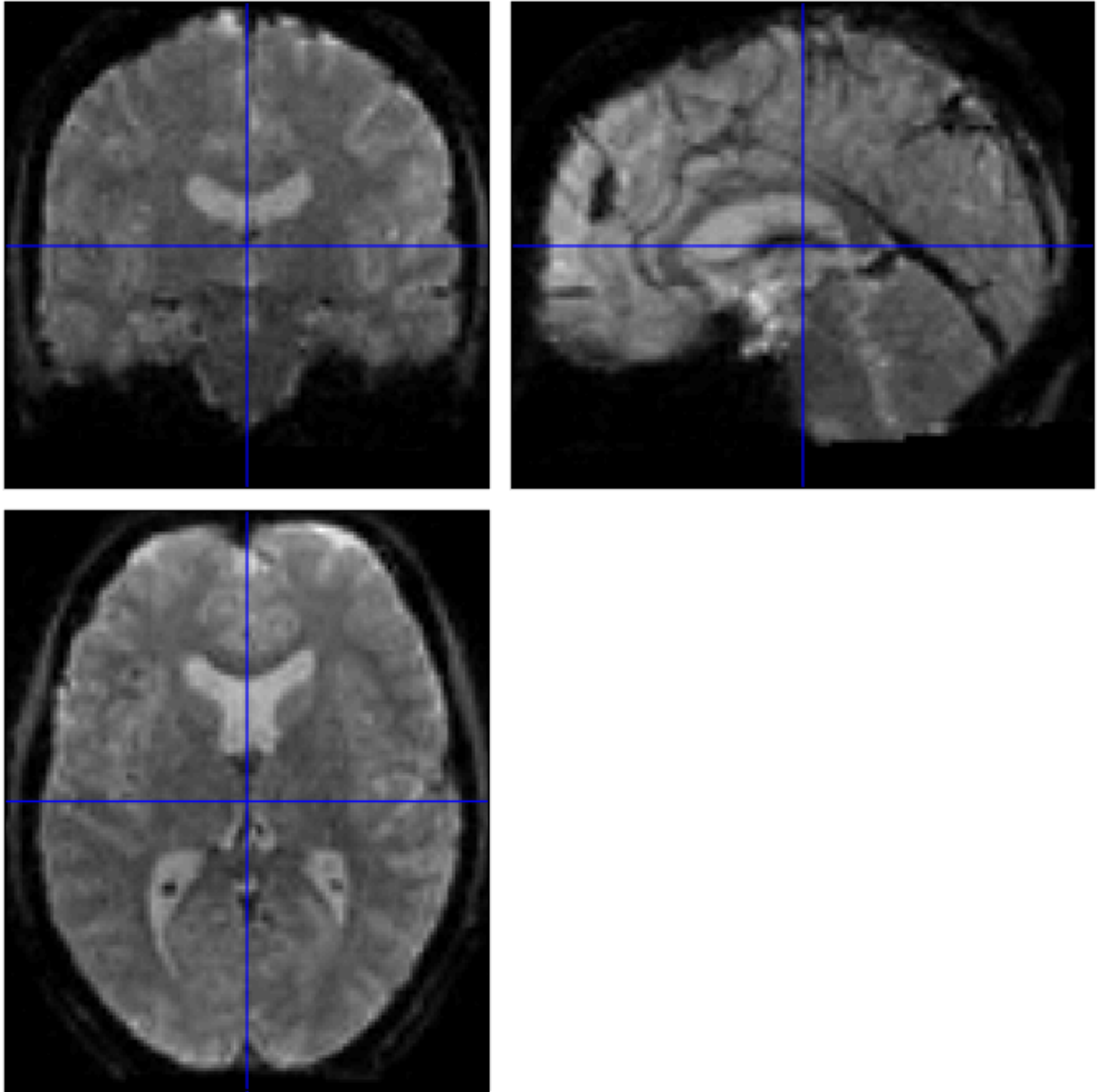


Figure 7: Normalization



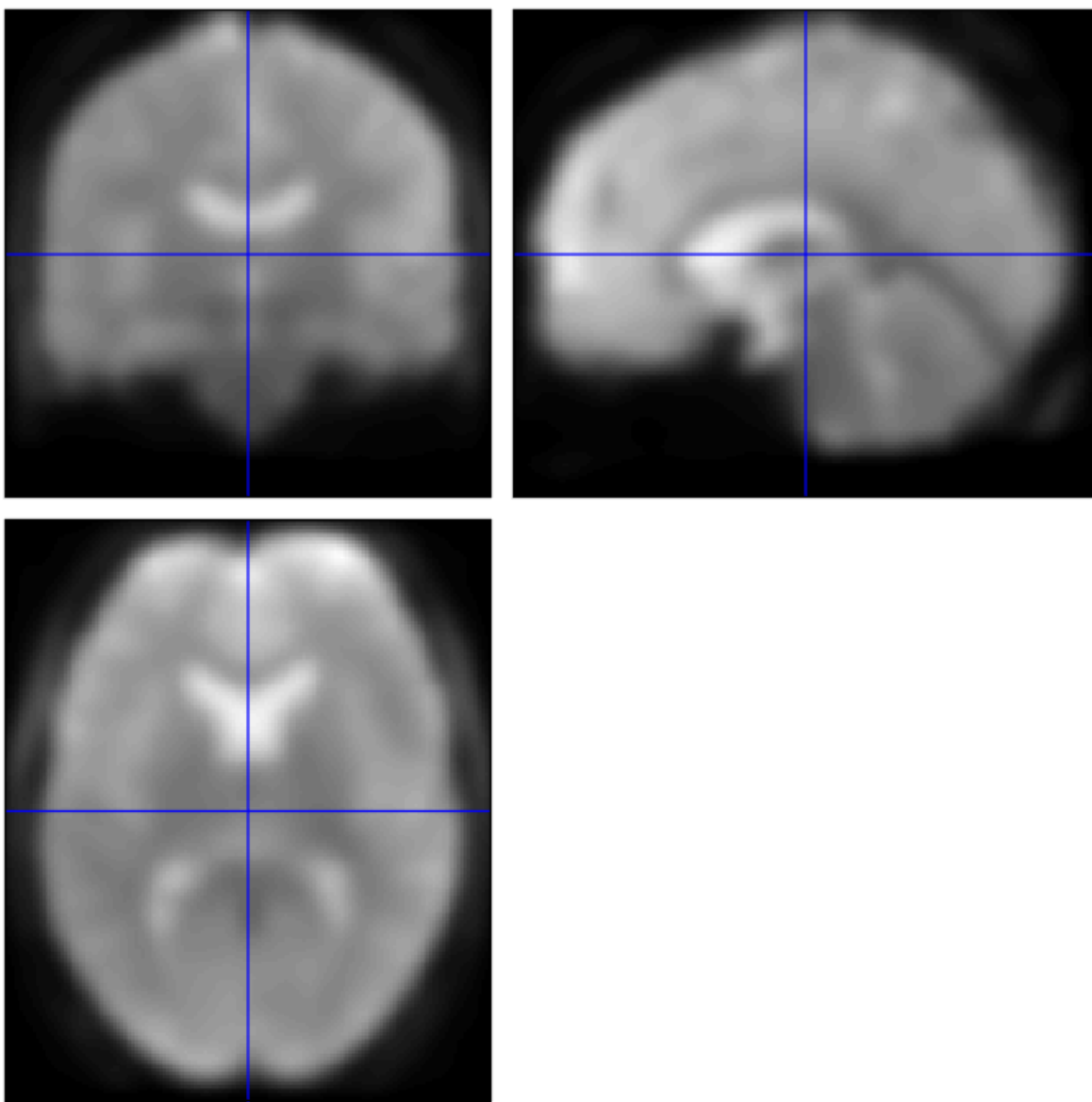
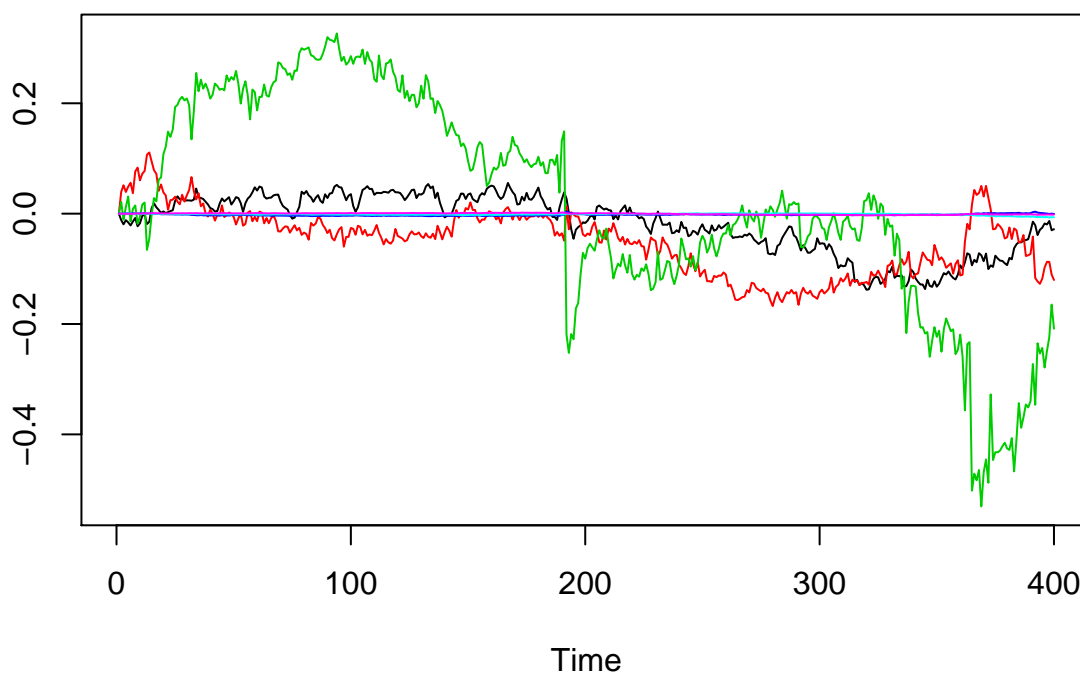


Figure 8: Smoothing

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   X2 = col_double(),
##   X3 = col_double(),
##   X4 = col_double(),
##   X5 = col_double(),
##   X6 = col_double()
## )
```

3.a. Make a lineplot of the realignment parameters in R.

```
ts.plot(Real, col = 1:6)
```



3.b. How far has the participant moved for each dimension during the experiment (Hint: use “apply()” to run a function across columns)?

Using the function apply to apply the function range across all columns. This gives us the minimum and maximum values of the realignment parameters. Afterwards the difference between these values are calculated, which gives the total of how far the participant has moved across all axis.

```
ran <- apply(Real, 2, range)
apply(ran, 2, diff)
```

```
##           X1           X2           X3           X4           X5           X6
## 0.193561140 0.278387520 0.856998800 0.008542486 0.007595161 0.005134256
```

The output shows that the participants moved most across the z-axis, and least across the x-axis. The rotational values are all very low, less than 0.5 degrees (converted from radians).

**3.c. Are any of the realignment parameters significantly correlated with the fMRI model (same model as used in exercise 3)?**

```
fmrides <- as.matrix(read.csv("fmri.design.csv", header=FALSE))
real <- as.matrix(Real)
mod <- lm(real ~ fmrides)
summary.lm(mod)
```

```
##
## Call:
## lm(formula = real ~ fmrides)
##
## Residuals:
##          X1          X2          X3          X4          X5          X6
## Min      -0.11596 -0.120299 -0.556003 -2.552e-03 -0.0050446 -0.0028180
## 1Q       -0.03487 -0.046767 -0.111789 -1.331e-03 -0.0006062 -0.0013354
## Median   0.01317  0.009864 -0.009521 -6.261e-05  0.0001511  0.0006317
## 3Q        0.04600  0.040498  0.179215  7.259e-04  0.0009998  0.0011231
## Max       0.07500  0.162744  0.331940  5.964e-03  0.0023060  0.0022622
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## [1,] -0.018576    0.010521  -1.766   0.0782 .
## [2,] -0.006568    0.012543  -0.524   0.6008
## [3,] -0.004150    0.012543  -0.331   0.7409
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2104 on 397 degrees of freedom
## Multiple R-squared:  0.06054,    Adjusted R-squared:  0.05581
## F-statistic: 12.79 on 2 and 397 DF,  p-value: 4.13e-06
```

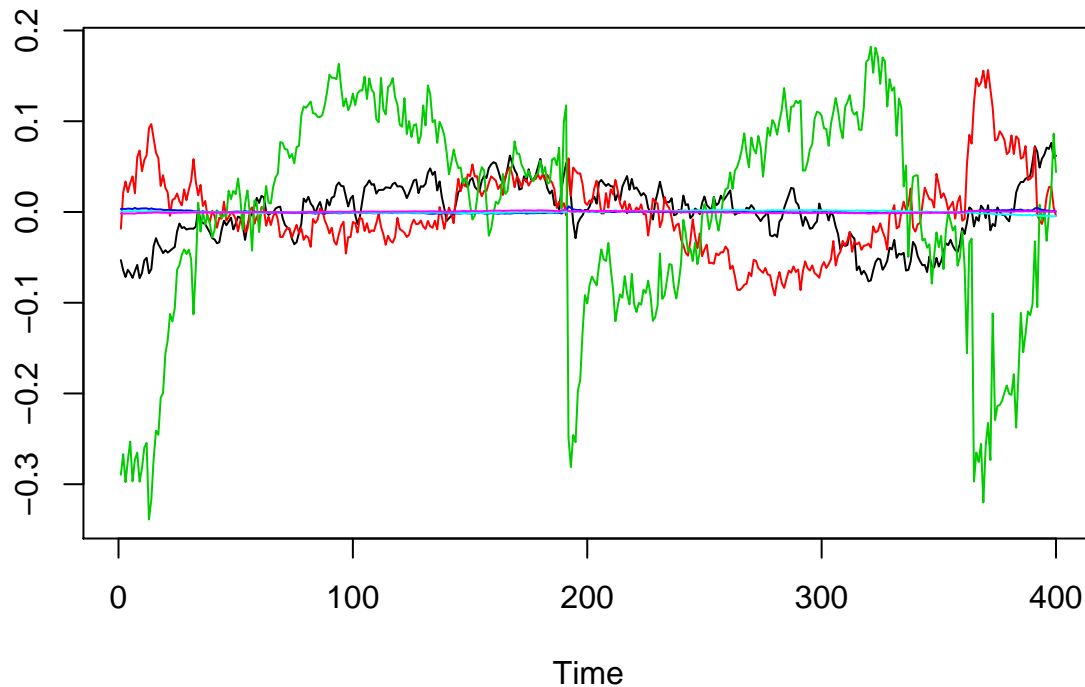
The first model between V1 and the realignment parameters shows significant correlation between data4 and data5 which is the rotation parameters. the second model between v2 and the realignment parameters shows significant correlation between data2, data3, data4, and data6. so in this case it is significant for both the y- and z-axis and rotational parameters.

Remove linear effects of time from the realignment parameters (hint: 1:400, fit a line and use residuals).

**3.d. Make a lineplot of the realignment parameters with time removed.**

Creating a vector containing the numbers 1-400 representing the passage of time. this is then used as a predictor for our data, in order to remove any variation explained by this fact. Thereafter the residuals are extracted from the model and two plots are created.

```
constant <- 1:400
modd <- lm(real ~ constant)
res <- modd$residuals
res <- as.data.frame(modd$residuals)
ts.plot(res, col = 1:6)
```



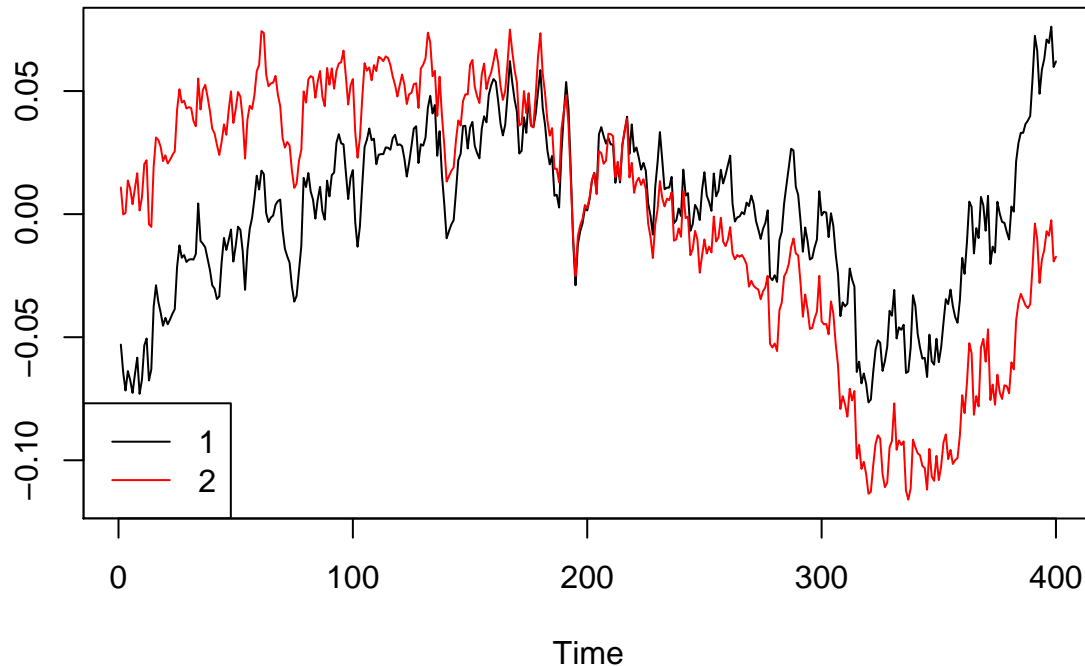
This correction has improved the variance in our data caused by the drift, as can be seen if the first plot is compared to this one.

### 3.e. Make a lineplot including only the first realignment parameter before and after removal.

We are removing the first parameters from the timecorrected residuals and from the residuals that are not time corrected. Then we are binding them into a dataframe and creating two plottypes to illustrate the correction of the data.

```
res1 <- as.data.frame(mod$residuals)
combined <- as.matrix(cbind(res$X1, res1$X1))

ts.plot(combined, col = 1:2)
legend('bottomleft', legend = 1:2, col = 1:2, lty = 1)
```



In our plot, 1 = time corrected and 2 = not time corrected. It can be discerned that the time corrected line has less variance overall.

### 3.f. Are the realignment parameters (corrected for effects of time) now correlated with the fMRI model?

Creating a model that uses the time corrected residuals to predict the fmrides model.

```
realparam <- as.matrix(res)
summary(lm(fmrides ~ realparam))
```

```
## Response V1 :
##
## Call:
## lm(formula = V1 ~ realparam)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.3286 -0.7889 -0.5749  1.1487  1.7751
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.349e-06  4.862e-02   0.000   1.000
## realparamX1 -4.342e+00  3.938e+00  -1.103   0.271
## realparamX2  2.858e+00  2.789e+00   1.025   0.306
## realparamX3 -5.346e-01  1.066e+00  -0.501   0.616
## realparamX4 -1.444e+02  7.531e+01  -1.917   0.056 .
## realparamX5  1.272e+02  5.403e+01   2.355   0.019 *
## realparamX6  1.832e+01  1.682e+02   0.109   0.913
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```

## Residual standard error: 0.9725 on 393 degrees of freedom
## Multiple R-squared:  0.06852,    Adjusted R-squared:  0.0543
## F-statistic: 4.818 on 6 and 393 DF,  p-value: 9.256e-05
##
##
## Response V2 :
##
## Call:
## lm(formula = V2 ~ realparam)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4721 -0.7655 -0.5104  1.1115  1.9904
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.045e-06  4.905e-02   0.000  0.99997
## realparamX1 -1.781e+01  3.973e+00 -4.484 9.64e-06 ***
## realparamX2 -8.510e+00  2.813e+00 -3.025  0.00265 **
## realparamX3  9.567e-01  1.076e+00  0.889  0.37429
## realparamX4  1.756e+02  7.597e+01  2.312  0.02132 *
## realparamX5 -1.115e+02  5.451e+01 -2.045  0.04148 *
## realparamX6  6.871e+02  1.697e+02  4.049 6.20e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.981 on 393 degrees of freedom
## Multiple R-squared:  0.0521, Adjusted R-squared:  0.03762
## F-statistic: 3.6 on 6 and 393 DF,  p-value: 0.001737

```

According to the new model in V1 only the res5 (rotation) show significant results. In v2 res1 (distance), res2 (distance), res4 (rotations), res5 (rotations), res6 (rotations) are significant.