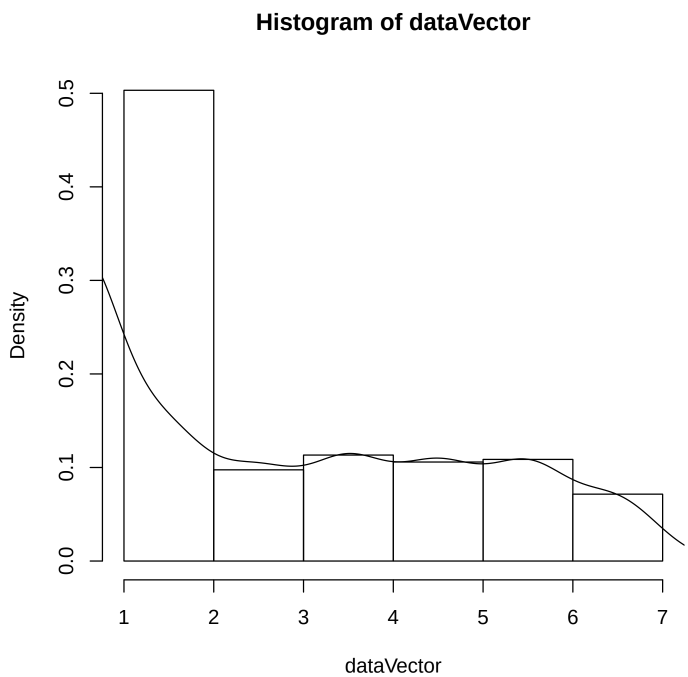
# 1 Raw data acquisition

```
> data = read.csv("study3-emotions-data.csv")
> dataSelf<-data[,c(1,2:19)]
> dataSelf1<-data[,c(1,20:37)]
> dataSelf2<-data[,c(1,38:55)]
```

Density distribution of the scores:

```
> dataVector <- na.omit(as.vector(t(dataSelf1[,-1])))
> hist(dataVector,probability = T, breaks = "FD")
> lines(density(dataVector - 0.5))
```




This is used to simulate random scoring with a similar distribution. This simulation has been performed with the Python script `random-scores-simulator.py` using the following formula for the distribution:
`random() < 0.5 ?  1 :  randint(2,7)`.

```
> # 8999 samples gradings, generated by random-scores-simulator.pp
> randomdata = read.csv("random-data.csv")
> randomDataSelf<-randomdata[,c(1,2:19)]
> randomDataSelf1<-randomdata[,c(1,20:37)]
> randomDataSelf2<-randomdata[,c(1,38:55)]
>
```

```
> # check the density distribution of the random data
> #randomdataVector <- na.omit(as.vector(t(randomDataSelf1[,-1])))
> #hist(randomdataVector,probability = T, breaks = "FD")
> #lines(density(randomdataVector - 0.5))
```

# 2 Computation of the degree of accuracy of partners' models

$\mathcal{M}^{\circ}(B, A)$ and $\mathcal{M}^{\circ}(C, A)$.

```
> d1<-abs(dataSelf-dataSelf1)
> d1$meanDiff<-rowMeans(d1[,c(2:19)],na.rm=T)
> d2<-abs(dataSelf-dataSelf2)
> d2$meanDiff<-rowMeans(d2[,c(2:19)],na.rm=T)
> d<-as.data.frame(cbind(d1$meanDiff,d2$meanDiff))
> names(d)<-c("scoreB","scoreC")
> # same thing for the random data
> randomd1<-abs(randomDataSelf-randomDataSelf1)
> randomd1$meanDiff<-rowMeans(randomd1[,c(2:19)],na.rm=T)
> randomd2<-abs(randomDataSelf-randomDataSelf2)
> randomd2$meanDiff<-rowMeans(randomd2[,c(2:19)],na.rm=T)
> randomd<-as.data.frame(cbind(randomd1$meanDiff,randomd2$meanDiff))
>
```

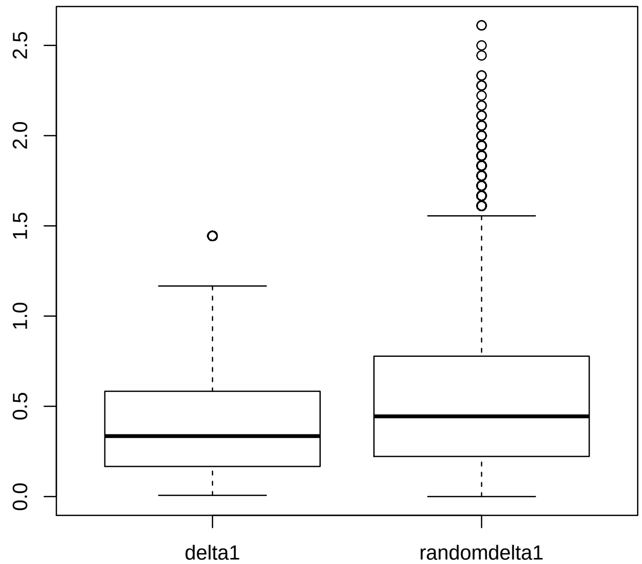
# 3 Computation of $\Delta_1$

```
> delta1<-c(1:60)
> for(i in seq(1, 60, by = 3)) {
+   delta1[i]<-abs(d[i,1]-d[i+1,1])
+   delta1[i+1]<-abs(d[i+2,1]-d[i,2])
+   delta1[i+2]<-abs(d[i+1,2]-d[i+2,2])
+ }
>
```

We test $\Delta_1$ against randomly attributed scores (following the same distribution as in the actual data) to check if we observe any symmetrical modelling effects (that would translate into significantly lower $\Delta_1$).

```
> randomdelta1<-c(1:8999)
> for(i in seq(1, 8999, by = 3)) {
+   randomdelta1[i]<-abs(randomd[i,1]-randomd[i+1,1])
+   randomdelta1[i+1]<-abs(randomd[i+2,1]-randomd[i,2])
+   randomdelta1[i+2]<-abs(randomd[i+1,2]-randomd[i+2,2])
+ }
> boxplot(cbind(delta1, randomdelta1))
```

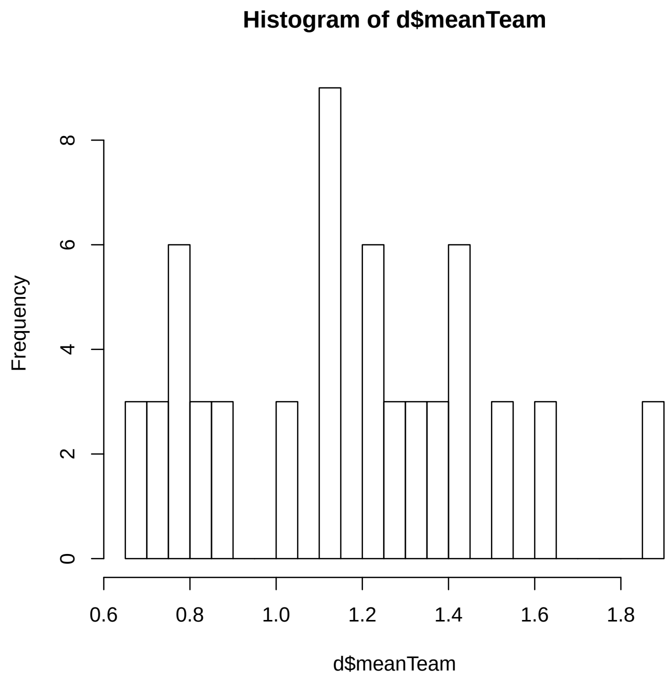
```
> # are variances equal?
> var.test(as.vector(delta1), as.vector(randomdelta1))

        F test to compare two variances

data:  as.vector(delta1) and as.vector(randomdelta1)
F = 0.5863, num df = 59, denom df = 8997, p-value = 0.009486
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.4206348 0.8729522
sample estimates:
ratio of variances
         0.5862648

> t.test(delta1, randomdelta1,var.equal = F)

        Welch Two Sample t-test

data:  delta1 and randomdelta1
t = -3.3032, df = 60.35, p-value = 0.00161
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.21396938 -0.05257898
sample estimates:
mean of x mean of y
0.4032339 0.5365081
```




The symmetry hypothesis tells that *good* modelling should be symmetrical, but does not say much about *bad* modelling.

hence, the effect should be even stronger if we check the average $\Delta_1$ of the best modellers against random data.

To do so, we first compute the average grading score per triad.




A dip test confirms that the distribution is not uni-modal:

```
> diptest::dip.test(d$meanTeam)

        Hartigans' dip test for unimodality / multimodality

data:  d$meanTeam
D = 0.0656, p-value = 0.0444
alternative hypothesis: non-unimodal, i.e., at least bimodal
```

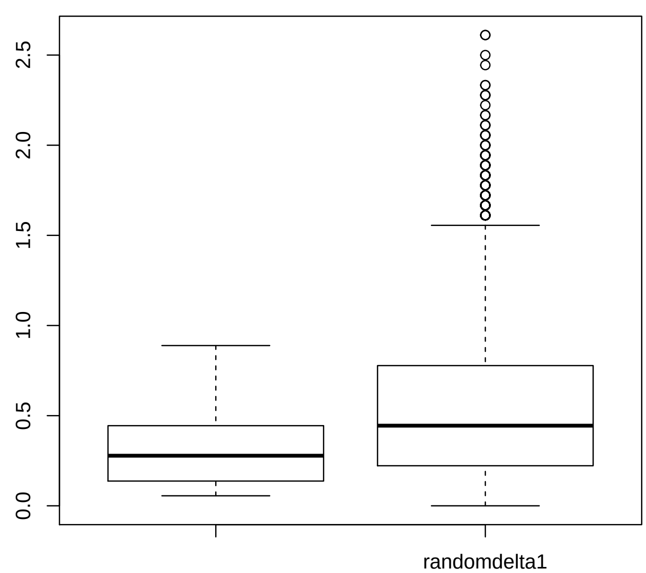
We use a K-Mean clustering to know where to cut our group of modellers, provinding an initial estimate of the clusters' centers:

```
> centers <- c(1:2)
> centers[1] = 0.8
> centers[2] = 1.2
> clusters<-kmeans(d$meanTeam,centers)
> # alternatively, let kmeans find the centers. But this may change between runs due to the
> #clusters<-kmeans(d$meanTeam,2)
>
> clusters$centers

       [,1]
1 0.8270308
2 1.3575446
```

```
> top <- max(d$meanTeam[clusters$cluster == 1])
> bottom <- min(d$meanTeam[clusters$cluster == 2])
> d$teamRating<-ifelse(d$meanTeam<=(top+bottom)/2,"good","bad")
```

We finally re-run a t-test on this subset of the modellers:

```
> boxplot(cbind(delta1[d$teamRating=="good"], randomdelta1))
> t.test(delta1[d$teamRating=="good"], randomdelta1,var.equal = F)

        Welch Two Sample t-test

data:  delta1[d$teamRating == "good"] and randomdelta1
t = -4.554, df = 20.296, p-value = 0.0001866
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.3322192 -0.1236169
sample estimates:
mean of x mean of y
0.3085901 0.5365081
```



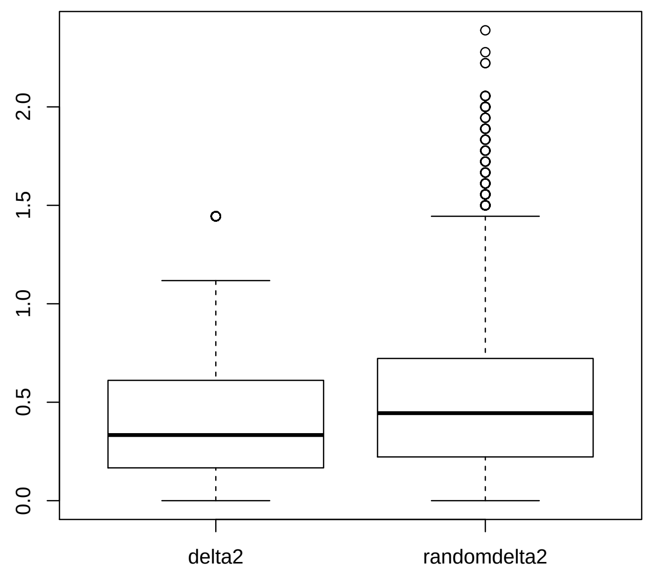

This confirms the symmetry hypothesis: the difference of scores for good modellers with random data is even stronger.

## 4 Computation of $\Delta_2$

```
> delta2<-c(1:60)
> for(i in seq(1, 60, by = 3)) {
+   delta2[i]<-abs(d[i,1]-d[i,2])
+   delta2[i+1]<-abs(d[i+1,1]-d[i+1,2])
+   delta2[i+2]<-abs(d[i+2,1]-d[i+2,2])
+ }
```

As previously, we test $\Delta_2$ against randomly attributed scores to check if we observe any similar modelling behaviours between two persons judging the same third one (which would translate into significantly lower $\Delta_2$ compared to chance):

```
> boxplot(cbind(delta2, randomdelta2))
> t.test(delta2, randomdelta2)

        Welch Two Sample t-test

data:  delta2 and randomdelta2
t = -1.6119, df = 59.957, p-value = 0.1122
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.15956495  0.01716017
sample estimates:
mean of x mean of y
0.4226920 0.4938944
```




In average, $\Delta_2$ is not significantly smaller than chance: this does not support the hypothesis of $\mathcal{M}^{\circ}(A, C)$ and $\mathcal{M}^{\circ}(B, C)$ being correlated, ie modelling accuracy would either comes for the overall quality of the interaction, or C fosters good models by making his emotional state easily readable.

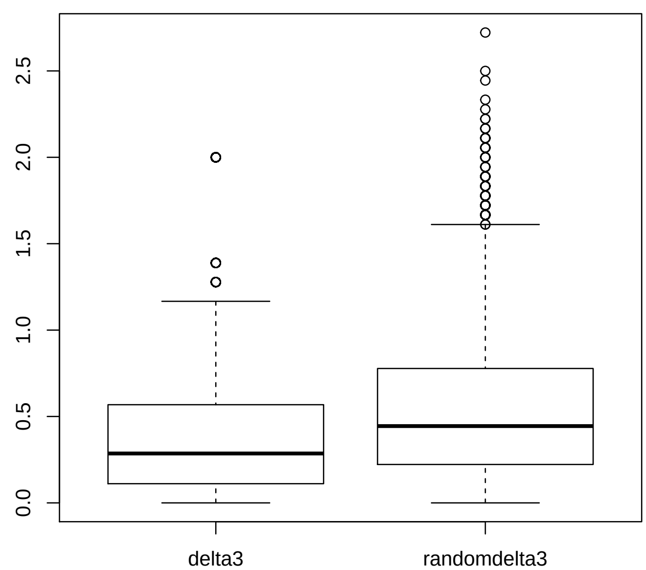
# 5 Computation of $\Delta_3$

```
> delta3<-c(1:60)
> for(i in seq(1, 60, by = 3)) {
+   delta3[i]<-abs(d[i+1,1]-d[i+2,1])
+   delta3[i+1]<-abs(d[i,1]-d[i+2,2])
+   delta3[i+2]<-abs(d[i,2]-d[i+1,2])
+ }
```

As previously, we test $\Delta_3$ against randomly attributed scores to check if we observe any similar modelling behaviours between two persons judging the same third one (which would translate into significantly lower $\Delta_3$ compared to chance):

```
> boxplot(cbind(delta3, randomdelta3))
> t.test(delta3, randomdelta3)
```

```
        Welch Two Sample t-test

data:  delta3 and randomdelta3
t = -2.625, df = 59.788, p-value = 0.01098
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.24410153 -0.03296222
sample estimates:
mean of x mean of y
0.3982911 0.5368230
```




We obtain a significantly lower $\Delta_3$, which supports the hypothesis of $\mathcal{M}^\circ(C, A)$ and $\mathcal{M}^\circ(C, B)$ being correlated: it supports the hypothesis that some persons are better modellers than others.