Statistics Examen_NeuroBIM

Maxime Houtekamer

16 november 2015

Loading the data file into R

First of all, the data was loaded into R from the textfile. A summary of the data was obtained as an indication whether the data was correctly loaded into R.

```
data<-read.table("lesionsBIM.txt",header=TRUE)
data1<-read.table("lesionsBIM.txt",header=TRUE)
```

Let's add a collumn for the difference between time needed

```
diff<-vector()
for(i in 1:(length(data[,1]))){
  if(as.character(data[i,1]) == "D5 "){
    diff<-c(diff,(data[i,22] - data[i,26]))
      } else if(as.character(data[i,1]) == "D3 "){
    diff<-c(diff,(data[i,22] - data[i,24]))
    }
}
data<-cbind(data,diff)
```

Creating seperate files for the 4 conditions

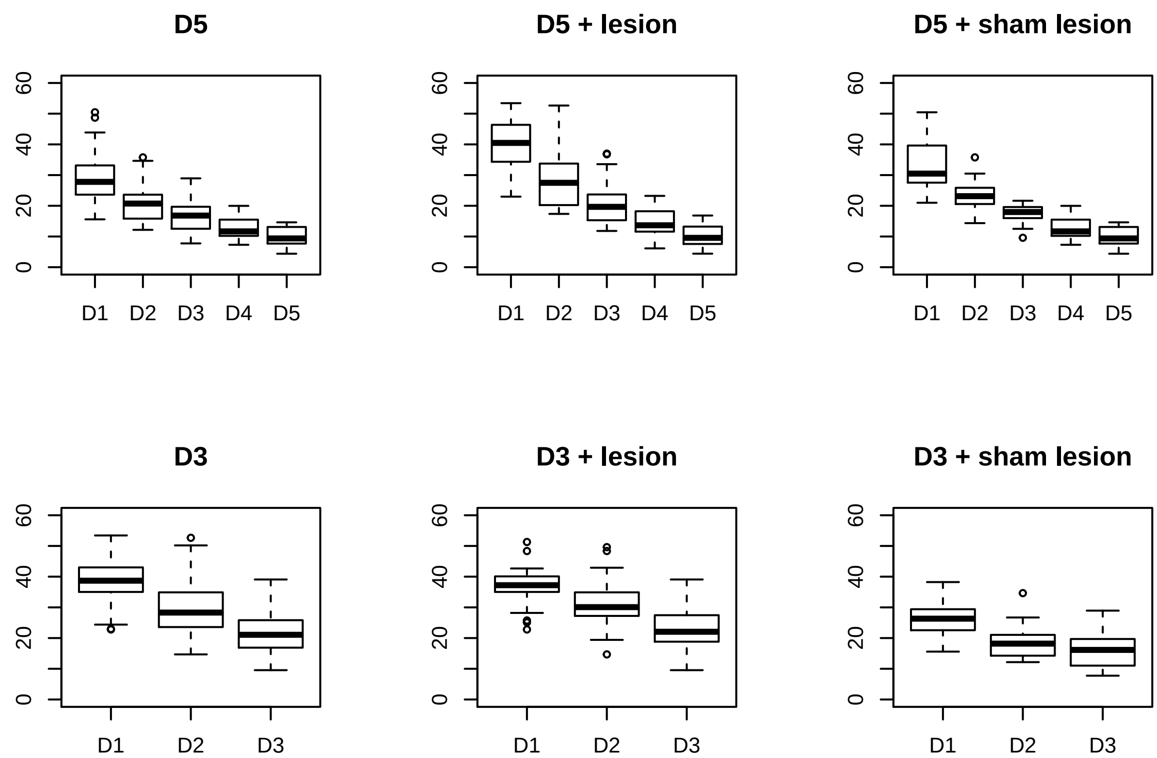
The mice were either trained in 3 sessions (D3) or in 5 sessions (D5). Within each of these two groups, the animals were either lesioned in the dorsal hippocampus (H) or they were given a sham lesion (SH). These groups were originally stored in the datafile, but will now be sorted in order to easily be able to display them seperately.

```
d3<-data[1:48,]
d5<-data[49:90,]
d3sh<-data[49:72,]
d5sh<-data[73:90,]
d3h<-data[25:48,]
d5h<-data[1:24,]
```

Learning time

The mice were given a task, and the time they spent in the dark is a measure of how well they learnt it.

```r
par(mfrow=c(2,3))
boxplot(d5[22:26],ylim=c(0, 60))
title(main="D5")
boxplot(d5h[22:26],ylim=c(0, 60))
title(main="D5 + lesion")
boxplot(d5sh[22:26],ylim=c(0, 60))
title(main="D5 + sham lesion")
boxplot(d3[22:24],ylim=c(0, 60))
title(main="D3")
boxplot(d3h[22:24],ylim=c(0, 60))
title(main="D3 + lesion")
boxplot(d3sh[22:24],ylim=c(0, 60))
title(main="D3 + sham lesion")
```




```r
mean(d5h[,27])
```

```
## [1] 29.5587
```

```r
mean(d5sh[,27])
```

```
## [1] 22.88382
```

```r
mean(d3h[,27])
```

```
## [1] 13.70844
```

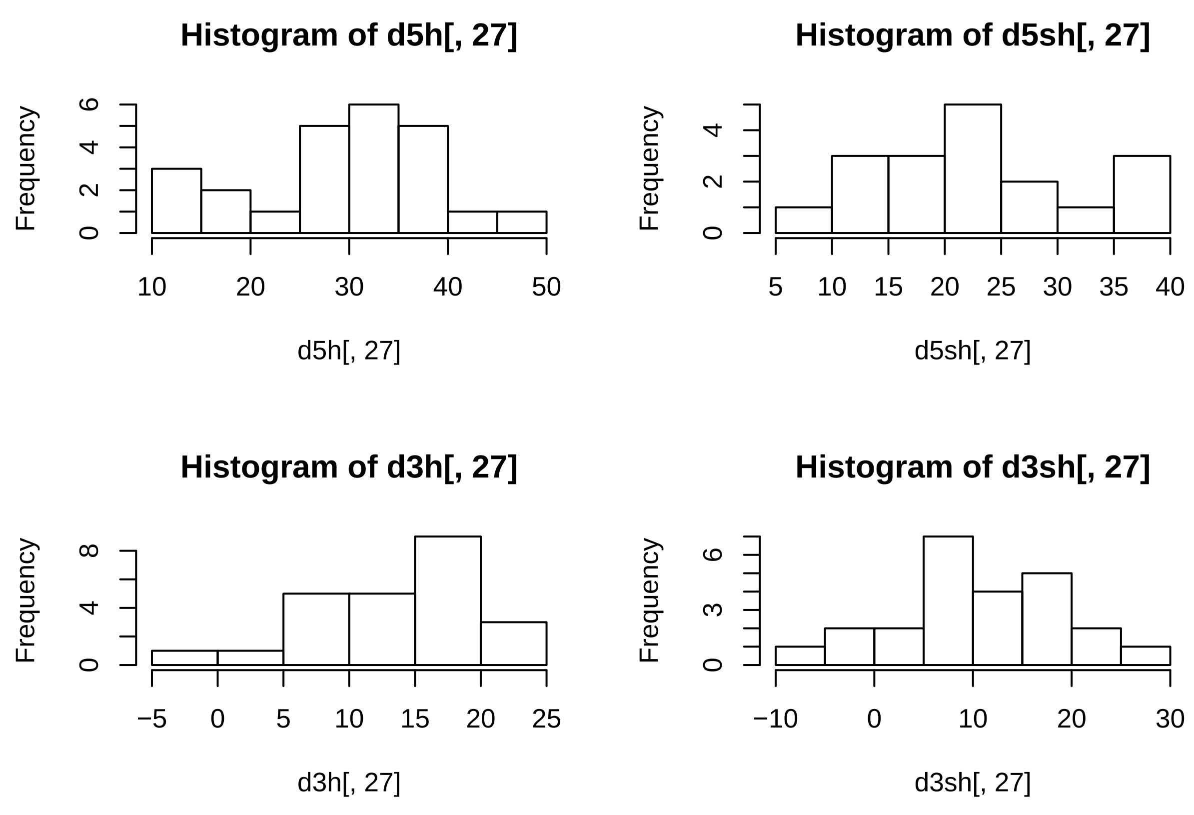
```r
mean(d3sh[,27])
```

```
## [1] 10.73177
```

Is the data normally distributed?

First, Let's look at histograms of each group.

```r
par(mfrow=c(2,2))
hist(d5h[,27])
hist(d5sh[,27])
hist(d3h[,27])
hist(d3sh[,27])
```



NORMAL DISTIBUTION?

We will carry out the shapiro-Wilk test. If p>a (bigger than 0.05 generally), the data is normal.

```
shapiro.test(d5h[,27])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  d5h[, 27]
## W = 0.96506, p-value = 0.5482
```

```
shapiro.test(d5sh[,27])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  d5sh[, 27]
## W = 0.94456, p-value = 0.3461
```

```
shapiro.test(d3h[,27])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  d3h[, 27]
## W = 0.95398, p-value = 0.3297
```

```
shapiro.test(d3sh[,27])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  d3sh[, 27]
## W = 0.98565, p-value = 0.9735
```

All the values are higher than p=0.05, so the data is normally distributed.

Making a new dataframe for ANOVA

We will make a list of the factors (d5h, d5sh, d3h, d3sh), and a list with the "learned" decrease in time needed to explore the matrix.

```
factorlist<-c((rep("d5h",24)),(rep("d5sh",18)),(rep("d3h",24)),(rep("d3sh",24)))
variablelist<-c(d5h[,27],d5sh[,27],d3h[,27],d3sh[,27])
d1<-data.frame(factorlist,variablelist)
colnames(d1)<-c("exp","values")
f1<-d1$values~d1$exp
```

Homogeneity of Variance

```r
bartlett.test(f1)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  d1$values by d1$exp
## Bartlett's K-squared = 3.6572, df = 3, p-value = 0.3009
```

ANOVA

Maybe we should instead to a repeated measures anova where we follow the animal over the different learning trials.

```r
aov1<-aov(f1)
summary(aov1)
```

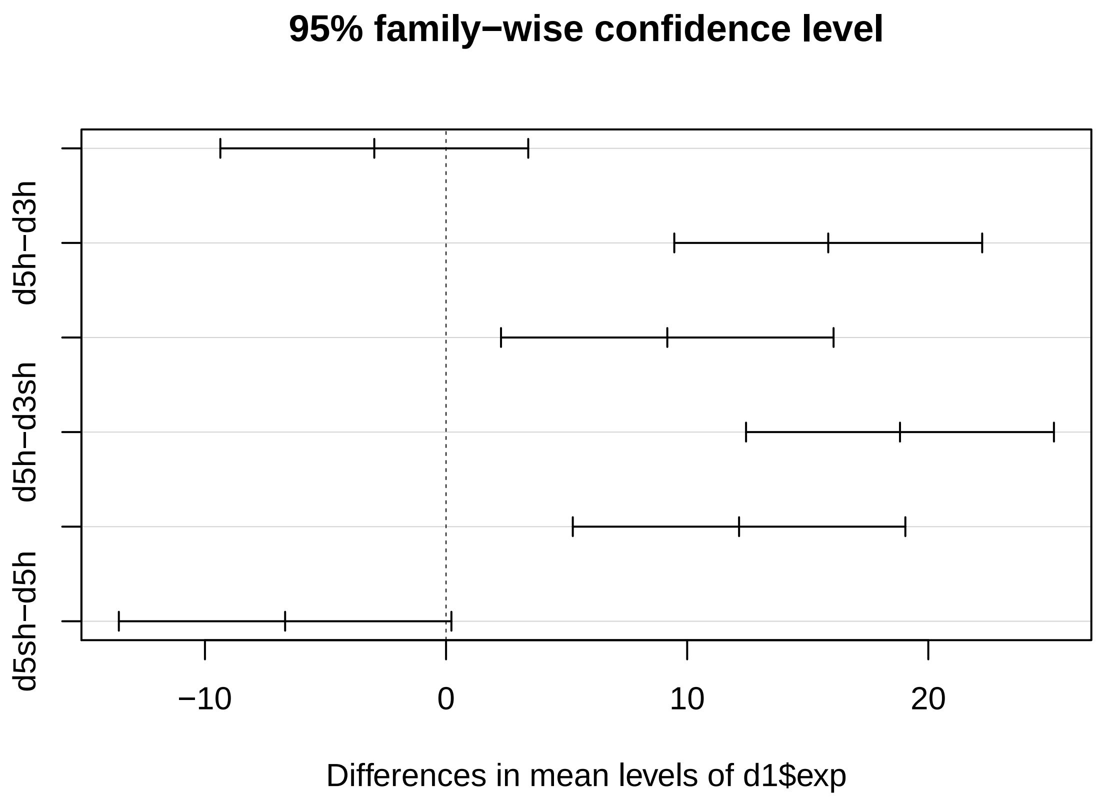
```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## d1$exp       3   5260  1753.3    24.6 1.38e-11 ***
## Residuals   86   6129    71.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The anova is highly significant at p<0.05. Let's do a post-hoc Tukey test to find where the differences are #Tukey posthoc

```r
t1<-TukeyHSD(aov1)
t1
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = f1)
##
## $`d1$exp`
##                diff        lwr        upr     p adj
## d3sh-d3h  -2.976667  -9.361469  3.4081355 0.6150204
## d5h-d3h   15.850260   9.465458 22.2350626 0.0000000
## d5sh-d3h   9.175382   2.279007 16.0717565 0.0042380
## d5h-d3sh  18.826927  12.442125 25.2117293 0.0000000
## d5sh-d3sh 12.152049   5.255674 19.0484232 0.0000790
## d5sh-d5h  -6.674878 -13.571253  0.2214961 0.0614873
```

```r
plot(t1)
```



#Repeated measures ANOVA

```
id<-vector()
for(i in 1:42){
  id<-c(id,(rep(i,5)))
  }
for(i in 43:90){
  id<-c(id,(rep(i,3)))
  }


group<-c((rep("d5h",(24*5))),(rep("d5sh",(18*5))),(rep("d3h",(24*3))),(rep("d3sh",(24*3))))



tasktime<-vector()
for(i in 1:24){
     tasktime<-c(tasktime,data1[i,22])
     tasktime<-c(tasktime,data1[i,23])
     tasktime<-c(tasktime,data1[i,24])
     tasktime<-c(tasktime,data1[i,25])
     tasktime<-c(tasktime,data1[i,26])
  }
for(i in 73:90){
    tasktime<-c(tasktime,data1[i,22])
    tasktime<-c(tasktime,data1[i,23])
    tasktime<-c(tasktime,data1[i,24])
```

```r
    tasktime<-c(tasktime,data1[i,25])
    tasktime<-c(tasktime,data1[i,26])
  }
for(i in 25:72){
      tasktime<-c(tasktime,data1[i,22])
      tasktime<-c(tasktime,data1[i,23])
      tasktime<-c(tasktime,data1[i,24])
  }

time<-c(rep(1:5,(42)),rep(1:3,(48)))

d2<-data.frame(id,group,tasktime,time)

print(d2)
```

```
##     id group tasktime time
## 1    1   d5h 48.38000    1
## 2    1   d5h 28.80625    2
## 3    1   d5h 27.01500    3
## 4    1   d5h 14.48250    4
## 5    1   d5h 10.91750    5
## 6    2   d5h 47.01750    1
## 7    2   d5h 40.18375    2
## 8    2   d5h 30.90500    3
## 9    2   d5h 16.19875    4
## 10   2   d5h  7.69250    5
## 11   3   d5h 52.95875    1
## 12   3   d5h 37.85750    2
## 13   3   d5h 18.29125    3
## 14   3   d5h 10.86500    4
## 15   3   d5h  9.94000    5
## 16   4   d5h 45.73250    1
## 17   4   d5h 52.65875    2
## 18   4   d5h 36.78000    3
## 19   4   d5h 18.55000    4
## 20   4   d5h 16.48500    5
## 21   5   d5h 40.93750    1
## 22   5   d5h 50.17625    2
## 23   5   d5h 13.62750    3
## 24   5   d5h 23.08750    4
## 25   5   d5h 14.89875    5
## 26   6   d5h 53.42175    1
## 27   6   d5h 25.65375    2
## 28   6   d5h 17.83750    3
## 29   6   d5h 11.33000    4
## 30   6   d5h  6.35250    5
## 31   7   d5h 43.23125    1
## 32   7   d5h 30.82875    2
## 33   7   d5h 14.63375    3
## 34   7   d5h 11.68125    4
## 35   7   d5h  8.93250    5
## 36   8   d5h 27.60625    1
## 37   8   d5h 28.32875    2
```

```
## 38   8   d5h 23.49625    3
## 39   8   d5h  6.13375    4
## 40   8   d5h  5.65750    5
## 41   9   d5h 24.38000    1
## 42   9   d5h 19.54625    2
## 43   9   d5h 11.80625    3
## 44   9   d5h 15.19750    4
## 45   9   d5h  6.62625    5
## 46  10   d5h 38.91875    1
## 47  10   d5h 22.88750    2
## 48  10   d5h 20.95250    3
## 49  10   d5h 12.71500    4
## 50  10   d5h  5.99875    5
## 51  11   d5h 42.79625    1
## 52  11   d5h 18.06125    2
## 53  11   d5h 15.94625    3
## 54  11   d5h 12.50000    4
## 55  11   d5h  7.45000    5
## 56  12   d5h 26.86625    1
## 57  12   d5h 26.55000    2
## 58  12   d5h 23.72750    3
## 59  12   d5h 13.73500    4
## 60  12   d5h 16.60625    5
## 61  13   d5h 22.97750    1
## 62  13   d5h 18.35375    2
## 63  13   d5h 21.80375    3
## 64  13   d5h 11.50875    4
## 65  13   d5h 10.11625    5
## 66  14   d5h 27.68375    1
## 67  14   d5h 30.80375    2
## 68  14   d5h 36.98625    3
## 69  14   d5h 16.22875    4
## 70  14   d5h  9.17250    5
## 71  15   d5h 47.80750    1
## 72  15   d5h 30.19625    2
## 73  15   d5h 16.71375    3
## 74  15   d5h 18.32500    4
## 75  15   d5h  9.06125    5
## 76  16   d5h 45.26500    1
## 77  16   d5h 19.26750    2
## 78  16   d5h 12.88500    3
## 79  16   d5h 13.42625    4
## 80  16   d5h  7.70250    5
## 81  17   d5h 39.95125    1
## 82  17   d5h 27.55250    2
## 83  17   d5h 21.18625    3
## 84  17   d5h 20.02500    4
## 85  17   d5h  9.99375    5
## 86  18   d5h 37.22125    1
## 87  18   d5h 17.36500    2
## 88  18   d5h 12.33125    3
## 89  18   d5h  7.63875    4
## 90  18   d5h  4.41625    5
## 91  19   d5h 39.75000    1
```

```
## 92  19   d5h 21.41125    2
## 93  19   d5h 20.02125    3
## 94  19   d5h 13.55875    4
## 95  19   d5h 14.46375    5
## 96  20   d5h 48.02875    1
## 97  20   d5h 20.52000    2
## 98  20   d5h 13.28375    3
## 99  20   d5h 18.12750    4
## 100 20   d5h 14.99625    5
## 101 21   d5h 31.44750    1
## 102 21   d5h 36.67625    2
## 103 21   d5h 33.55250    3
## 104 21   d5h 23.22625    4
## 105 21   d5h 16.83125    5
## 106 22   d5h 37.53750    1
## 107 22   d5h 39.69625    2
## 108 22   d5h 23.65500    3
## 109 22   d5h 13.48000    4
## 110 22   d5h 11.98875    5
## 111 23   d5h 40.02500    1
## 112 23   d5h 19.93875    2
## 113 23   d5h 18.62375    3
## 114 23   d5h  9.56375    4
## 115 23   d5h  8.35875    5
## 116 24   d5h 44.15875    1
## 117 24   d5h 27.42750    2
## 118 24   d5h 19.29750    3
## 119 24   d5h 18.90000    4
## 120 24   d5h 10.03300    5
## 121 25  d5sh 27.63250    1
## 122 25  d5sh 16.18500    2
## 123 25  d5sh 21.63625    3
## 124 25  d5sh 10.21375    4
## 125 25  d5sh 14.46875    5
## 126 26  d5sh 50.45625    1
## 127 26  d5sh 23.19625    2
## 128 26  d5sh 17.84000    3
## 129 26  d5sh 11.85500    4
## 130 26  d5sh 14.61750    5
## 131 27  d5sh 26.81750    1
## 132 27  d5sh 15.08125    2
## 133 27  d5sh 16.41000    3
## 134 27  d5sh 11.01875    4
## 135 27  d5sh  5.73250    5
## 136 28  d5sh 27.94125    1
## 137 28  d5sh 25.86500    2
## 138 28  d5sh 15.97625    3
## 139 28  d5sh 17.85375    4
## 140 28  d5sh 13.22250    5
## 141 29  d5sh 48.70000    1
## 142 29  d5sh 23.75500    2
## 143 29  d5sh 18.56875    3
## 144 29  d5sh 19.97875    4
## 145 29  d5sh 10.05625    5
```

```
## 146 30  d5sh 43.87500    1
## 147 30  d5sh 22.80625    2
## 148 30  d5sh  9.58875    3
## 149 30  d5sh 12.04125    4
## 150 30  d5sh  5.74875    5
## 151 31  d5sh 20.98500    1
## 152 31  d5sh 22.71250    2
## 153 31  d5sh 20.65000    3
## 154 31  d5sh 16.73375    4
## 155 31  d5sh 11.56625    5
## 156 32  d5sh 40.89625    1
## 157 32  d5sh 23.61625    2
## 158 32  d5sh 15.57250    3
## 159 32  d5sh 15.49125    4
## 160 32  d5sh 10.15375    5
## 161 33  d5sh 35.58500    1
## 162 33  d5sh 35.76875    2
## 163 33  d5sh 18.10125    3
## 164 33  d5sh 17.94375    4
## 165 33  d5sh 13.10750    5
## 166 34  d5sh 28.51125    1
## 167 34  d5sh 23.98875    2
## 168 34  d5sh 18.11750    3
## 169 34  d5sh  7.57750    4
## 170 34  d5sh  8.36000    5
## 171 35  d5sh 39.63125    1
## 172 35  d5sh 30.20000    2
## 173 35  d5sh 12.49250    3
## 174 35  d5sh 11.16375    4
## 175 35  d5sh 13.60625    5
## 176 36  d5sh 30.71250    1
## 177 36  d5sh 23.07375    2
## 178 36  d5sh 16.59250    3
## 179 36  d5sh 13.01500    4
## 180 36  d5sh 12.86750    5
## 181 37  d5sh 30.51250    1
## 182 37  d5sh 29.30250    2
## 183 37  d5sh 18.37500    3
## 184 37  d5sh 11.15250    4
## 185 37  d5sh  7.70000    5
## 186 38  d5sh 30.45875    1
## 187 38  d5sh 14.34500    2
## 188 38  d5sh 12.51750    3
## 189 38  d5sh 11.47625    4
## 190 38  d5sh  7.77000    5
## 191 39  d5sh 21.76750    1
## 192 39  d5sh 20.94000    2
## 193 39  d5sh 19.59250    3
## 194 39  d5sh  9.48750    4
## 195 39  d5sh  7.91250    5
## 196 40  d5sh 23.55500    1
## 197 40  d5sh 20.57625    2
## 198 40  d5sh 16.18500    3
## 199 40  d5sh  8.29625    4
```

```
## 200 40  d5sh  6.33625    5
## 201 41  d5sh 32.63625    1
## 202 41  d5sh 30.48750    2
## 203 41  d5sh 21.31125    3
## 204 41  d5sh  7.29500    4
## 205 41  d5sh  4.39375    5
## 206 42  d5sh 27.54250    1
## 207 42  d5sh 20.16250    2
## 208 42  d5sh 21.02125    3
## 209 42  d5sh 13.19500    4
## 210 42  d5sh  8.68750    5
## 211 43   d3h 38.45500    1
## 212 43   d3h 31.34125    2
## 213 43   d3h 20.06875    3
## 214 44   d3h 36.06000    1
## 215 44   d3h 31.03125    2
## 216 44   d3h 17.04250    3
## 217 45   d3h 25.73250    1
## 218 45   d3h 27.72125    2
## 219 45   d3h 19.04375    3
## 220 46   d3h 51.29000    1
## 221 46   d3h 27.79125    2
## 222 46   d3h 39.07875    3
## 223 47   d3h 37.21375    1
## 224 47   d3h 19.41250    2
## 225 47   d3h 15.61875    3
## 226 48   d3h 42.67000    1
## 227 48   d3h 23.43375    2
## 228 48   d3h 21.39375    3
## 229 49   d3h 35.25375    1
## 230 49   d3h 28.28375    2
## 231 49   d3h 15.45750    3
## 232 50   d3h 42.59000    1
## 233 50   d3h 23.73875    2
## 234 50   d3h 19.08375    3
## 235 51   d3h 35.59500    1
## 236 51   d3h 28.12250    2
## 237 51   d3h 24.63625    3
## 238 52   d3h 38.18875    1
## 239 52   d3h 23.89000    2
## 240 52   d3h 18.60250    3
## 241 53   d3h 34.79750    1
## 242 53   d3h 40.18125    2
## 243 53   d3h 27.06750    3
## 244 54   d3h 39.42875    1
## 245 54   d3h 42.90875    2
## 246 54   d3h 23.73500    3
## 247 55   d3h 35.24375    1
## 248 55   d3h 49.61500    2
## 249 55   d3h 20.04125    3
## 250 56   d3h 40.69250    1
## 251 56   d3h 30.02375    2
## 252 56   d3h 22.00375    3
## 253 57   d3h 40.02000    1
```

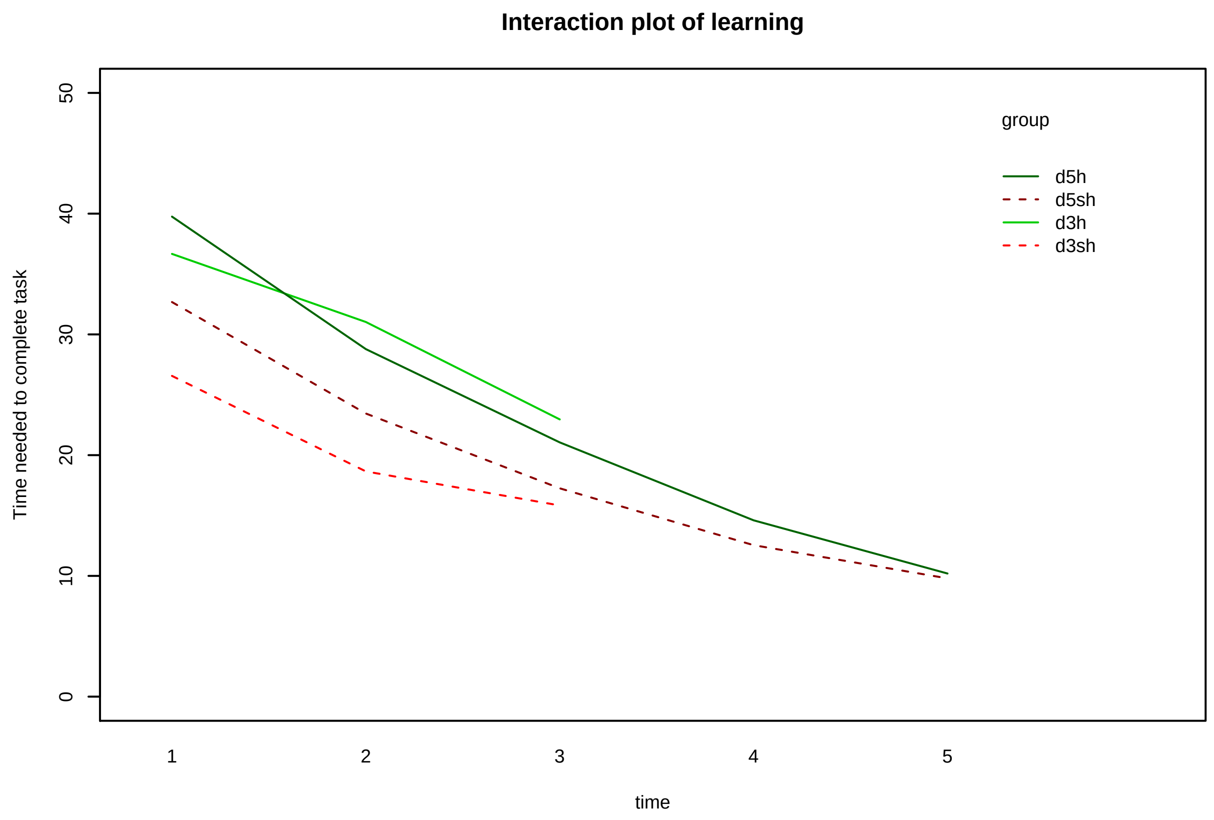
```
## 254 57   d3h 26.71750    2
## 255 57   d3h 30.05125    3
## 256 58   d3h 36.64000    1
## 257 58   d3h 30.53500    2
## 258 58   d3h 30.19375    3
## 259 59   d3h 48.34750    1
## 260 59   d3h 36.45875    2
## 261 59   d3h 28.64125    3
## 262 60   d3h 22.79625    1
## 263 60   d3h 34.76250    2
## 264 60   d3h 23.13875    3
## 265 61   d3h 37.84875    1
## 266 61   d3h 48.36250    2
## 267 61   d3h 35.83500    3
## 268 62   d3h 40.15375    1
## 269 62   d3h 32.54000    2
## 270 62   d3h 27.83250    3
## 271 63   d3h 25.10500    1
## 272 63   d3h 14.68375    2
## 273 63   d3h  9.54750    3
## 274 64   d3h 28.19000    1
## 275 64   d3h 35.02500    2
## 276 64   d3h 22.10375    3
## 277 65   d3h 37.22125    1
## 278 65   d3h 27.92000    2
## 279 65   d3h 24.46750    3
## 280 66   d3h 30.50125    1
## 281 66   d3h 30.12375    2
## 282 66   d3h 16.34750    3
## 283 67  d3sh 24.10125    1
## 284 67  d3sh 15.22250    2
## 285 67  d3sh 10.98625    3
## 286 68  d3sh 33.15000    1
## 287 68  d3sh 22.35375    2
## 288 68  d3sh 17.04250    3
## 289 69  d3sh 21.65875    1
## 290 69  d3sh 12.32500    2
## 291 69  d3sh 15.87375    3
## 292 70  d3sh 34.51875    1
## 293 70  d3sh 17.49750    2
## 294 70  d3sh 16.10750    3
## 295 71  d3sh 26.60625    1
## 296 71  d3sh 18.59125    2
## 297 71  d3sh 19.70750    3
## 298 72  d3sh 38.23375    1
## 299 72  d3sh 25.22375    2
## 300 72  d3sh 16.17875    3
## 301 73  d3sh 33.56625    1
## 302 73  d3sh 21.19000    2
## 303 73  d3sh  9.54375    3
## 304 74  d3sh 37.42125    1
## 305 74  d3sh 18.94875    2
## 306 74  d3sh  9.00750    3
## 307 75  d3sh 26.10375    1
```

```
## 308 75  d3sh 20.93250    2
## 309 75  d3sh 18.65375    3
## 310 76  d3sh 28.99375    1
## 311 76  d3sh 20.74125    2
## 312 76  d3sh 19.89500    3
## 313 77  d3sh 29.47000    1
## 314 77  d3sh 21.15500    2
## 315 77  d3sh 22.76375    3
## 316 78  d3sh 27.30250    1
## 317 78  d3sh 13.93875    2
## 318 78  d3sh  7.75000    3
## 319 79  d3sh 25.36000    1
## 320 79  d3sh 15.82125    2
## 321 79  d3sh 11.95500    3
## 322 80  d3sh 20.47375    1
## 323 80  d3sh 12.16375    2
## 324 80  d3sh 19.68500    3
## 325 81  d3sh 24.52625    1
## 326 81  d3sh 13.64125    2
## 327 81  d3sh 13.21125    3
## 328 82  d3sh 19.71875    1
## 329 82  d3sh 26.70875    2
## 330 82  d3sh 18.10625    3
## 331 83  d3sh 29.32000    1
## 332 83  d3sh 20.49625    2
## 333 83  d3sh 16.07875    3
## 334 84  d3sh 29.16125    1
## 335 84  d3sh 14.39000    2
## 336 84  d3sh 10.30125    3
## 337 85  d3sh 18.60500    1
## 338 85  d3sh 15.97625    2
## 339 85  d3sh 20.55750    3
## 340 86  d3sh 15.58750    1
## 341 86  d3sh 17.78625    2
## 342 86  d3sh 17.89500    3
## 343 87  d3sh 26.54500    1
## 344 87  d3sh 14.14500    2
## 345 87  d3sh 20.66125    3
## 346 88  d3sh 23.45875    1
## 347 88  d3sh 20.70875    2
## 348 88  d3sh  7.99625    3
## 349 89  d3sh 20.00875    1
## 350 89  d3sh 13.06375    2
## 351 89  d3sh 11.06125    3
## 352 90  d3sh 23.63125    1
## 353 90  d3sh 34.64250    2
## 354 90  d3sh 28.94125    3
```

```
##convert variables to factor
d2<-within(d2, {
  group<-factor(group)
  time<-factor(time)
  id<-factor(id)
})
```

```r
par(cex = .6)

with(d2,interaction.plot(time, group, tasktime,
                         ylim= c(0,50), lty=c(1,20,1,20),col = c(3,2,"darkgreen","darkred"),
                         ylab= "Time needed to complete task", xlab= "time",trace.label="group",main="In
```




```r
d2.aov<-aov(tasktime ~ group * time + Error(id), data=d2)
summary(d2.aov)
```

```
##
## Error: id
##           Df Sum Sq Mean Sq F value   Pr(>F)
## group      3   5594  1864.7   27.11 1.95e-12 ***
## Residuals 86   5914    68.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Within
##             Df Sum Sq Mean Sq F value  Pr(>F)
## time         4  22320    5580 169.998 < 2e-16 ***
## group:time   8    751      94   2.861 0.00462 **
## Residuals  252   8272      33
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

model.cs <- gls(tasktime ~ group * time , data = d2, corr = corCompSymm(, form = ~ 1 | id))

summary(model.cs)

The between group tests indicates that the variable group is significant. consequently, int he graph we see that the liens for the two groups are rather far apart. The within subject test indicates that there is a significant time effect, inother words, the groups do change other time, both groups are taking less time to complete the task over time. Morover, the interaction of time and group is significant which means that the groups are changing over time but are changing in different ways, which means that in the graph, the liens will not be parallel.

Are there structures that are differentially activated depending on the duration of the training?

I think the best option here is MANOVA. I used this video first https://www.youtube.com/watch?v=48cZ2cMBpio

```
manova1<-manova( cbind(STLD,STMD,AMBASLAT,AMLAT,ENTORH,PERIRH,CA1,CA3,DG,CINGULAR,PRELIMB,SOMSENS,SUBICU
summary(manova1)
```

```
##                   Df  Pillai approx F num Df den Df    Pr(>F)
## as.factor(TRAIN)  1 0.87467   8.0807     19     22 4.842e-06 ***
## Residuals        40
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I don't think these results are helpful at all. So there are areas that differ.. ok..

Lets do t-tests…

```
factorlist<-c((rep("d5h",24)),(rep("d5sh",18)),(rep("d3h",24)),(rep("d3sh",24)))
variablelist<-c(d5h[,3],d5sh[,3],d3h[,3],d3sh[,3])
d1<-data.frame(factorlist,variablelist)
colnames(d1)<-c("exp","values")
f1<-d1$values~d1$exp
```

What are the structure activities that are correlated with performance (in the last training session) ?

Ok. So we need a correlation. then we need a matrix.

```
library(Hmisc)
```

```
## Loading required package: grid
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
## Loading required package: ggplot2
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor  =(cormat)[ut],
    p = pmat[ut]
    )
}

cordata<-cbind(data1[,3:21],data1[,24],data1[,26])
final<-c(data1[c(1:24,73:90),26],data1[c(25:72),24])
pooledcordata<-cbind(data1[,3:21],final)
cordataD5H<-cordata[1:24,c(1:6,10:19,21)]
cordataD3H<-cordata[25:48,1:20]
cordataD5SH<-cordata[73:90,c(1:19,21)]
cordataD3SH<-cordata[49:72,1:20]

resp<-rcorr(as.matrix(pooledcordata))
matrixp<-flattenCorrMatrix(resp$r, resp$P)
print(matrixp[172:190,])
```

```
##            row column         cor          p
## 172       STLD  final -0.25688878 0.01451892
## 173       STMD  final -0.11459476 0.28215434
## 174   AMBASLAT  final  0.12201443 0.25194559
## 175      AMLAT  final  0.03474115 0.74512462
## 176     ENTORH  final -0.08409247 0.43068690
## 177     PERIRH  final -0.06621543 0.53520851
## 178        CA1  final -0.29772902 0.05549844
## 179        CA3  final -0.30402470 0.05029389
## 180         DG  final -0.15854518 0.31593655
## 181   CINGULAR  final -0.23694505 0.02454052
## 182    PRELIMB  final  0.06786582 0.52505980
## 183    SOMSENS  final  0.09701581 0.36300415
## 184  SUBICULUM  final  0.24090412 0.02218022
## 185    ACCCORE  final -0.08076198 0.44922795
## 186   ACCSHELL  final -0.06811436 0.52353991
## 187     VISUAL  final  0.05980199 0.57554592
## 188   PIRIFORM  final -0.05737391 0.59117707
## 189   PARIETAL  final  0.06761815 0.52657656
## 190 RETROSPLEN  final  0.23501243 0.02576832
```

```
res1<-rcorr(as.matrix(cordataD5H))
matrix1<-flattenCorrMatrix(res1$r, res1$P)
print(matrix1[121:136,])
```

```
##            row      column         cor          p
## 121       STLD data1[, 26]  0.420943916 0.04052046
## 122       STMD data1[, 26]  0.508609712 0.01115149
## 123   AMBASLAT data1[, 26]  0.314945042 0.13387549
## 124      AMLAT data1[, 26]  0.153313786 0.47446618
```

```
## 125     ENTORH data1[, 26]  0.022545485 0.91671974
## 126     PERIRH data1[, 26] -0.029052418 0.89280380
## 127   CINGULAR data1[, 26]  0.275005847 0.19340416
## 128    PRELIMB data1[, 26] -0.273405820 0.19611358
## 129    SOMSENS data1[, 26] -0.005872661 0.97827284
## 130  SUBICULUM data1[, 26] -0.185807616 0.38469403
## 131    ACCCORE data1[, 26]  0.351101816 0.09251817
## 132   ACCSHELL data1[, 26]  0.088317879 0.68153189
## 133     VISUAL data1[, 26]  0.062166829 0.77290674
## 134   PIRIFORM data1[, 26]  0.175138682 0.41304156
## 135   PARIETAL data1[, 26]  0.292301059 0.16573976
## 136 RETROSPLEN data1[, 26]  0.003533508 0.98692608
```

```
res2<-rcorr(as.matrix(cordataD3H))
```

```
## Warning in sqrt(npair - 2): NaNs produced
```

```
matrix2<-flattenCorrMatrix(res2$r, res2$P)
print(matrix2[172:190,])
```

```
##            row      column          cor           p
## 172       STLD data1[, 24] -0.076227695 0.723325284
## 173       STMD data1[, 24] -0.002770538 0.989748891
## 174   AMBASLAT data1[, 24]  0.225778431 0.288775563
## 175      AMLAT data1[, 24]  0.246296376 0.245979988
## 176     ENTORH data1[, 24] -0.179720566 0.400729262
## 177     PERIRH data1[, 24] -0.532149076 0.007435526
## 178        CA1 data1[, 24]           NA          NA
## 179        CA3 data1[, 24]           NA          NA
## 180         DG data1[, 24]           NA          NA
## 181   CINGULAR data1[, 24] -0.004120424 0.984754725
## 182    PRELIMB data1[, 24] -0.236160472 0.266572650
## 183    SOMSENS data1[, 24]  0.205784410 0.334698112
## 184  SUBICULUM data1[, 24] -0.109996065 0.608889819
## 185    ACCCORE data1[, 24]  0.075546660 0.725703792
## 186   ACCSHELL data1[, 24]  0.177521572 0.406612533
## 187     VISUAL data1[, 24] -0.021707390 0.919805660
## 188   PIRIFORM data1[, 24]  0.150478333 0.482773527
## 189   PARIETAL data1[, 24]  0.236988276 0.264850721
## 190 RETROSPLEN data1[, 24] -0.021632330 0.920082088
```

```
res3<-rcorr(as.matrix(cordataD3SH))
matrix3<-flattenCorrMatrix(res3$r, res3$P)
print(matrix3[172:190,])
```

```
##            row      column         cor          p
## 172       STLD data1[, 24] -0.32145479 0.12559258
## 173       STMD data1[, 24] -0.28133541 0.18293688
## 174   AMBASLAT data1[, 24] -0.25245550 0.23398786
## 175      AMLAT data1[, 24] -0.21934773 0.30309215
## 176     ENTORH data1[, 24] -0.44434154 0.02960353
## 177     PERIRH data1[, 24] -0.30310613 0.14993311
```

```
## 178        CA1 data1[, 24] -0.14480291 0.49961947
## 179        CA3 data1[, 24] -0.10614695 0.62155328
## 180         DG data1[, 24]  0.06378335 0.76715843
## 181   CINGULAR data1[, 24] -0.45524520 0.02539334
## 182    PRELIMB data1[, 24]  0.06031009 0.77952358
## 183    SOMSENS data1[, 24] -0.22309427 0.29469885
## 184  SUBICULUM data1[, 24] -0.36111617 0.08296873
## 185    ACCCORE data1[, 24] -0.35409132 0.08958555
## 186   ACCSHELL data1[, 24] -0.06335748 0.76867166
## 187     VISUAL data1[, 24]  0.41607288 0.04314844
## 188   PIRIFORM data1[, 24] -0.36375323 0.08058232
## 189   PARIETAL data1[, 24] -0.29271892 0.16510766
## 190 RETROSPLEN data1[, 24] -0.17931114 0.40182103
```

```
res4<-rcorr(as.matrix(cordataD5SH))
matrix4<-flattenCorrMatrix(res4$r, res4$P)
print(matrix4[172:190,])
```

```
##            row      column         cor          p
## 172       STLD data1[, 26]  0.35961094 0.14272193
## 173       STMD data1[, 26]  0.11980815 0.63584001
## 174   AMBASLAT data1[, 26]  0.08361112 0.74151822
## 175      AMLAT data1[, 26]  0.20436074 0.41597587
## 176     ENTORH data1[, 26] -0.01583276 0.95028112
## 177     PERIRH data1[, 26]  0.12335004 0.62581283
## 178        CA1 data1[, 26] -0.06897060 0.78567310
## 179        CA3 data1[, 26]  0.20157377 0.42249904
## 180         DG data1[, 26]  0.12214736 0.62921076
## 181   CINGULAR data1[, 26]  0.26162213 0.29432116
## 182    PRELIMB data1[, 26]  0.52393258 0.02562968
## 183    SOMSENS data1[, 26]  0.39525074 0.10449866
## 184  SUBICULUM data1[, 26]  0.35399941 0.14951909
## 185    ACCCORE data1[, 26]  0.18270271 0.46806182
## 186   ACCSHELL data1[, 26]  0.13056333 0.60558531
## 187     VISUAL data1[, 26]  0.27053288 0.27757543
## 188   PIRIFORM data1[, 26]  0.25667286 0.30388027
## 189   PARIETAL data1[, 26]  0.38416871 0.11549134
## 190 RETROSPLEN data1[, 26]  0.35882717 0.14365820
```

First took all the data to see whether they are normally distributed or not: per brain area per group 4 groups, lesion vs. non lesion depending on normality: t test or wilcox

NORMAL DISTIBUTION?

We will carry out the shapiro-Wilk test. If p>a (bigger than 0.05 generally), the data is normal.

d5h[,c(3,17,20,21)] are not normally distributed d5sh[,c(3)] is not normally distributed d3h[,c(15,21)] are not normally distributed d3sh[,c(3,8,2,14,16,18)] is not normally distributed. You have a small sample size but the population is actually normally distributed, so we will use a parametric test anyway

ok fuck this shit let's do a t test for all these fuckers

t.test(collumn 1, collumn 2) skip collumns 9:11 for d3h vs d5h

```
for(i in c(3:8,12:21)){
  print(t.test(d3h[,i],d5h[,i]))
}
for(i in c(3:21)){
  print(t.test(d3sh[,i],d5sh[,i]))
}
```

FIGURETIME.