DREADDs PST2020 NORAD

EP

03/02/2021

Table of Contents

```
last pdf update generated ->  03 juillet, 2024 - 09:13:46
```

Introduction

A behavioral study of foraging and Noradrenaline input to MCC

The data were acquired by JS and CG at Inserm Lyon. The animals were tested with the modern apparatus,

On this board the 25 locations are numbered from *1* to *25* from the *upper left* corner to the *lower right* corner, going from left to right.

Data loading and formating

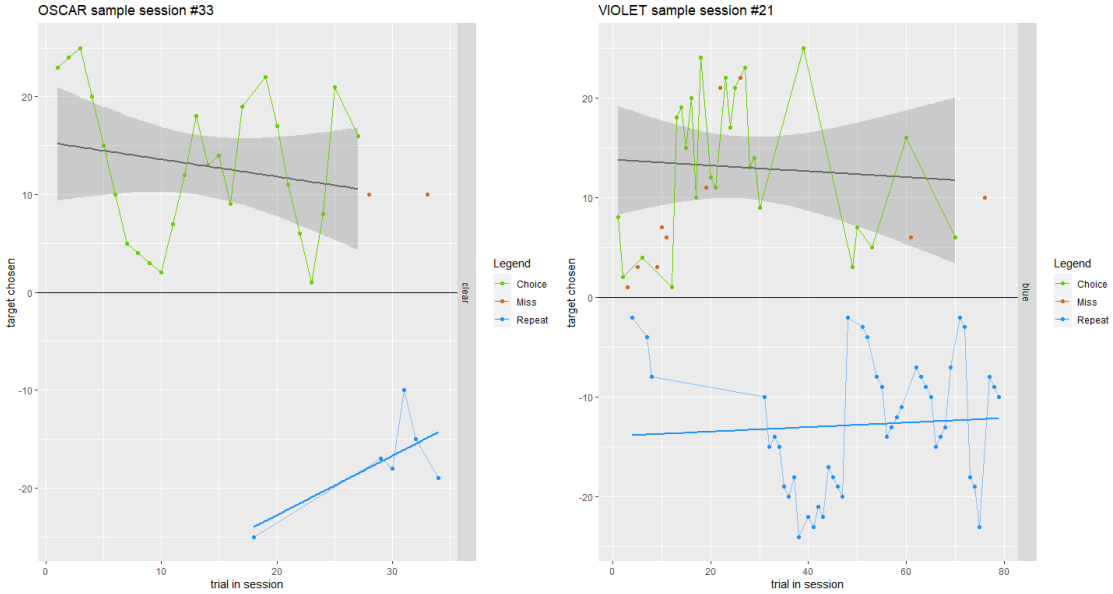
Data files include all data tested in monkeys *OSCAR* and *VIOLETTE* for the NORAD project.

session types are initially “0” for control (transparent doors) and “1” TEST with opaque blue doors, but we use the label Clear vs BLue in figures and analyses. Codes used for target chosen and repeats or miss reflect the position of the choice (location of the hole selected from 1 TO 25 top to bottom) , if negative value then it’s a repeat, if 999 then it’s a pause in the task, if it’s a value between 100 and 2500 then it is a mistake (the animal tried but missed the reward) the number /100 giving the location.

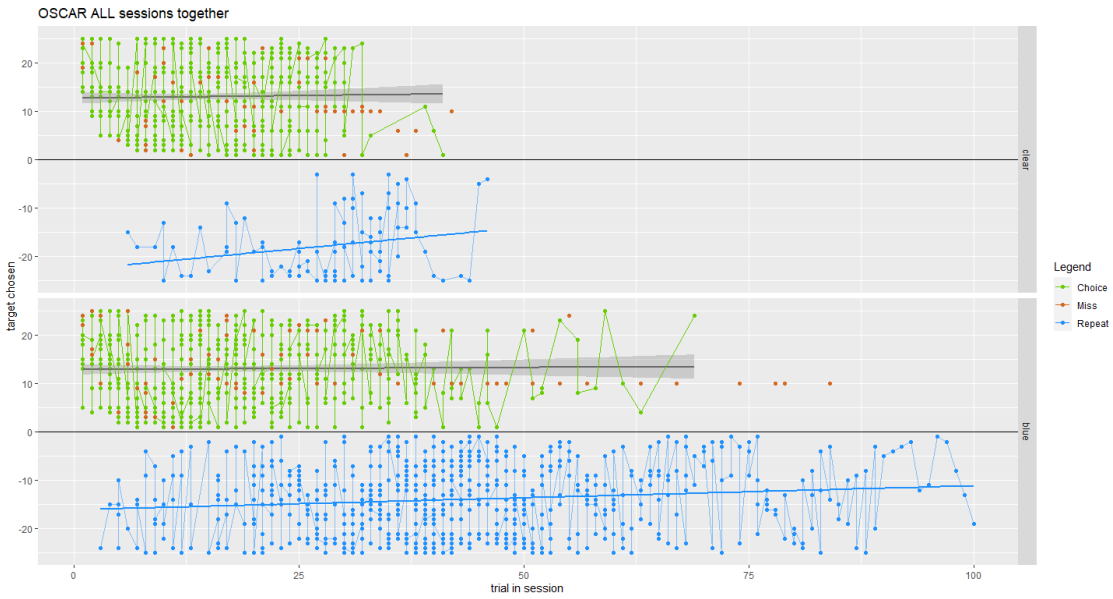
Main general Plots

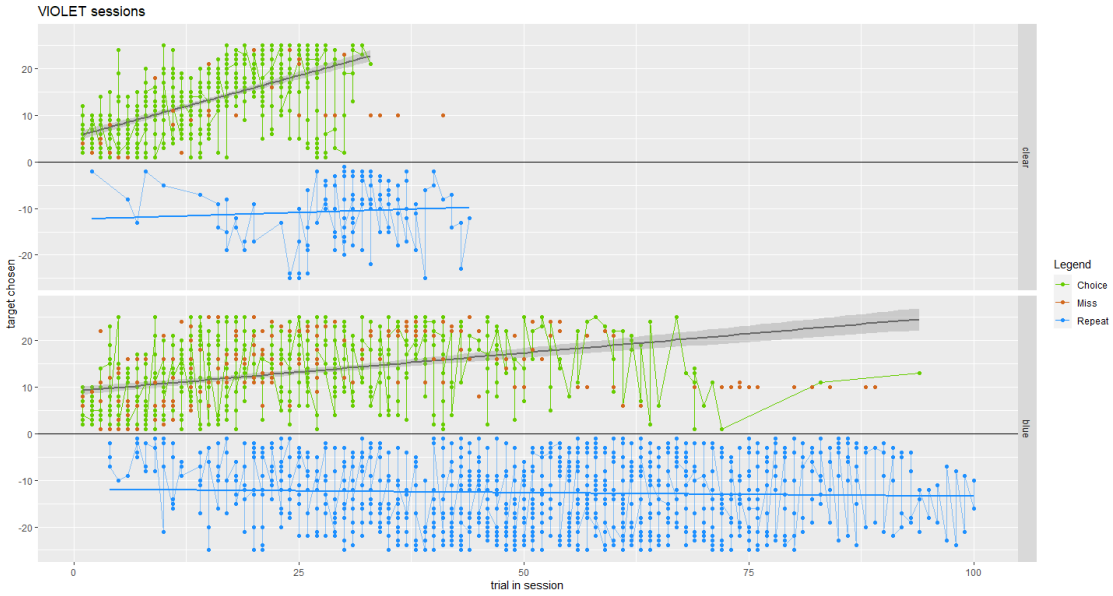
Let first see descriptive graphs for all sessions per monkey. The figures show for each monkey the choices (location of choice on board) selected by the animal trial by trial (chronological order from left to right). Green dots represent correct choices i.e. location chosen for the 1st time in session and with correct pick up of the reward. Blue dots represent returns to previously chosen location. These are presented as negative values to show their time course independently for correct choices. Orange is when reward is missed. Clear (transparent doors) and Blue (blue doors) sessions are presented separately First

sample sessions:



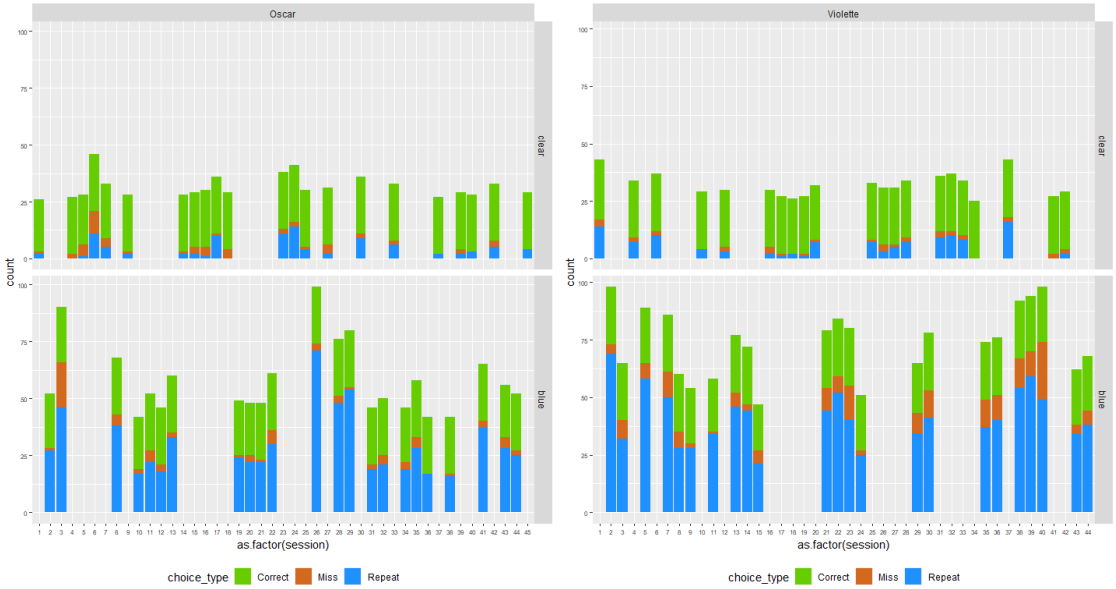
Then all data overlap across sessions to see the tendencies. In particular one can see the positive and negative trends that reflect monkeys choosing holes from top to bottom or bottom to top.OSCAR and VIOLET have different prefered directions but this is due to the different position of the setup in the housing.





Summary

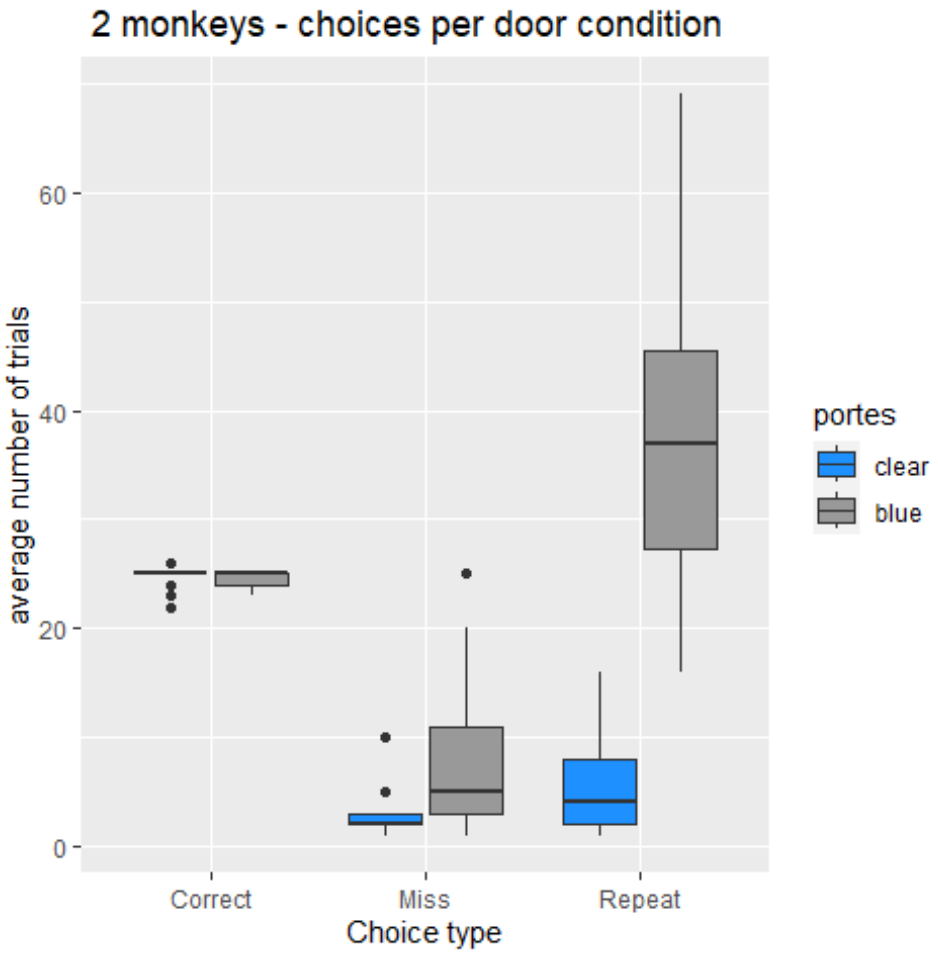
The data are then summarized in terms of frequency of each trial (choice) type: Correct, Miss, Repeat.

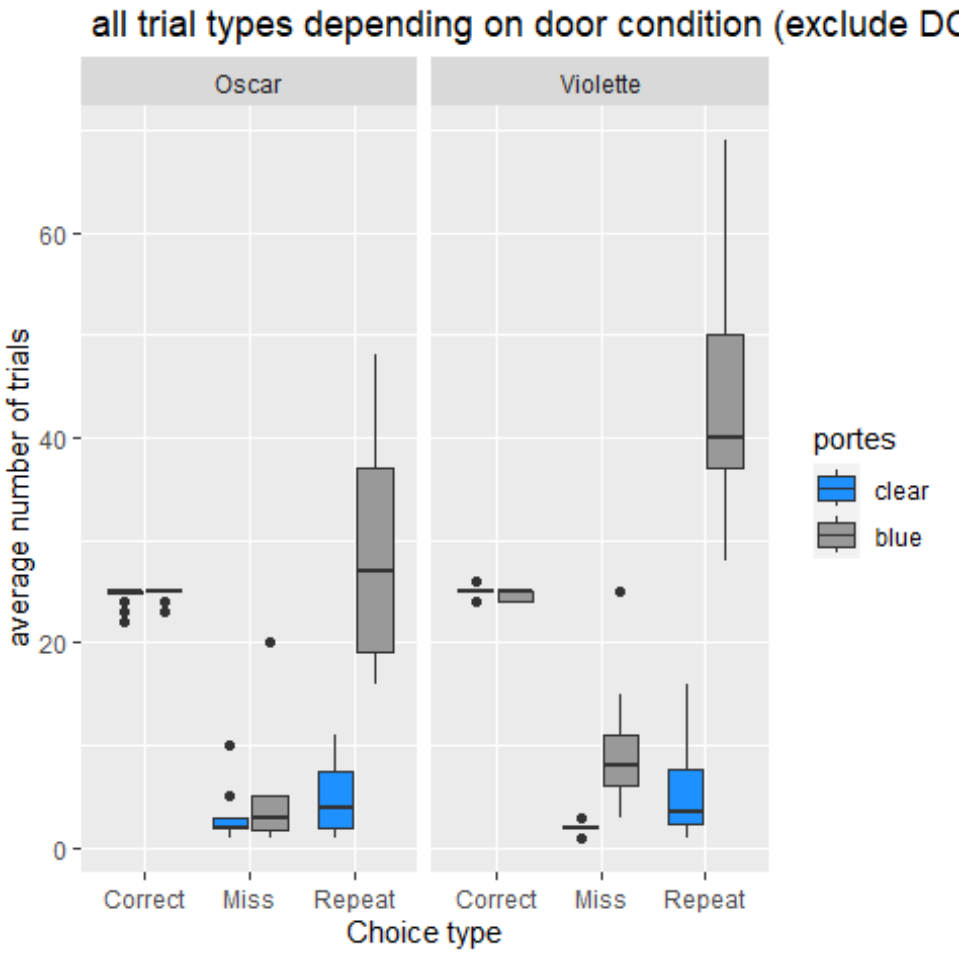


Summary all trials

Below the average number of each trial type for the different ‘portes’ (door) conditions.

We do not look at injections yet, this will be done later in the statistical anayses (DCZ vs Sham). See for all 8 monkeys or for VIOLET and OSCAR that there is a main effect of doors in particular on the number of repeats. Which makes sense because monkeys in blue condition (compared to clear) monkeys need to rely on memory to avoid repeating ; which obviously they don’t really succeed. We see later the #repeat is a very relevant parameter.


2 monkeys - choices per door condition
average number of trials
60
40
20
0
Correct
Miss
Repeat
Choice type
portes
clear
blue



##Stats We perform a logistic regression on the door effect for each monkey separately and test whether it influences the number of trial types - Still excluding DCZ sessions

```
## Warning: Dans subset.data.frame(agg.data4BnoDCZ, singe = "OSCAR") :
## l'argument supplémentaire 'singe' sera ignoré

## Warning: Dans subset.data.frame(agg.data4BnoDCZ, singe = "OSCAR") :
## l'argument supplémentaire 'singe' sera ignoré

##
## Call:
## glm(formula = trial ~ choice_type * portes, family = "poisson",
##     data = subset(agg.data4BnoDCZ, singe = "OSCAR"))
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   3.208386   0.041922  76.532  < 2e-16 ***
## choice_typeMiss              -2.272293   0.146169 -15.546  < 2e-16 ***
## choice_typeRepeat            -1.516710   0.102613 -14.781  < 2e-16 ***
## portesblue                   -0.003453   0.057598  -0.060    0.952
## choice_typeMiss:portesblue    1.013270   0.169233   5.987 2.13e-09 ***
```

```
## choice_typeRepeat:portesblue  1.906980   0.114654  16.633  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1708.34  on 140  degrees of freedom
## Residual deviance:  330.09  on 135  degrees of freedom
## AIC: 938.17
##
## Number of Fisher Scoring iterations: 5
```

```
## Warning: Dans subset.data.frame(agg.data4BnoDCZ, singe = "VIOLET") :
## l'argument supplémentaire 'singe' sera ignoré
```

```
## Warning: Dans subset.data.frame(agg.data4BnoDCZ, singe = "VIOLET") :
## l'argument supplémentaire 'singe' sera ignoré
```

```
##
## Call:
## glm(formula = trial ~ choice_type * portes, family = "poisson",
##     data = subset(agg.data4BnoDCZ, singe = "VIOLET"))
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   3.208386   0.041922  76.532  < 2e-16 ***
## choice_typeMiss              -2.272293   0.146169 -15.546  < 2e-16 ***
## choice_typeRepeat            -1.516710   0.102613 -14.781  < 2e-16 ***
## portesblue                   -0.003453   0.057598  -0.060    0.952
## choice_typeMiss:portesblue    1.013270   0.169233   5.987 2.13e-09 ***
## choice_typeRepeat:portesblue  1.906980   0.114654  16.633  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1708.34  on 140  degrees of freedom
## Residual deviance:  330.09  on 135  degrees of freedom
## AIC: 938.17
##
## Number of Fisher Scoring iterations: 5
```

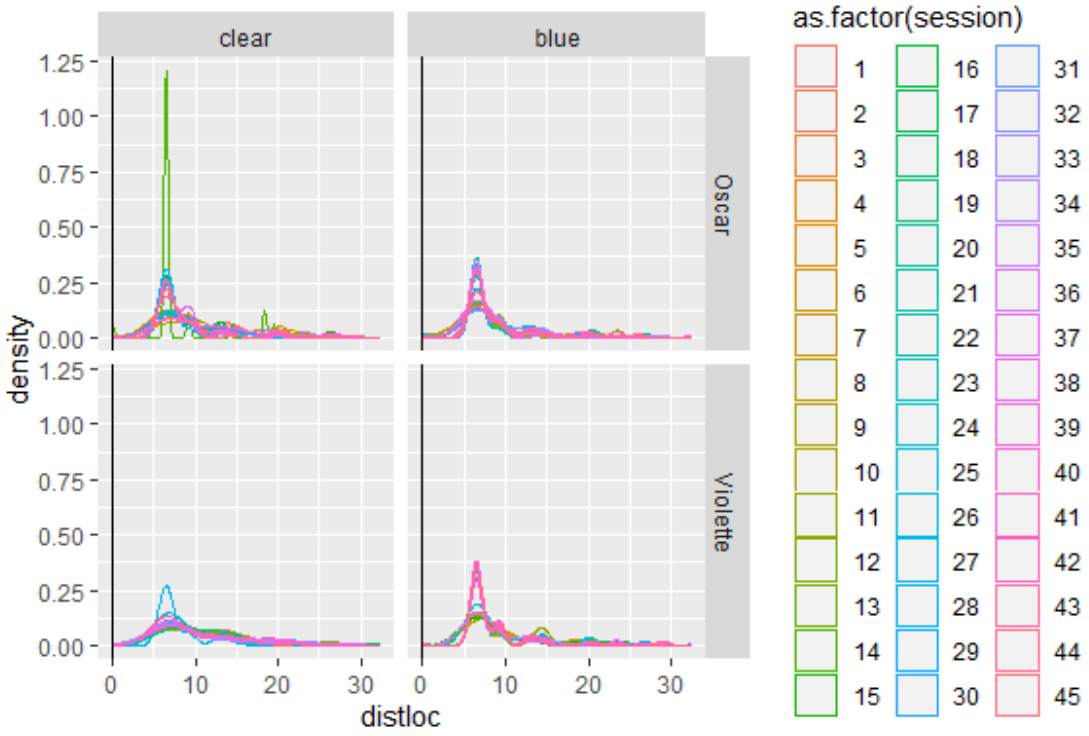
##Exploration strategies

One future interesting set of analyses we can do regards the strategies of exploration: how animals scan through the setup , and then of course how they forget and repeat choices. Needs to be quantified to be used when comparing ON/OFF DCZ sessions.

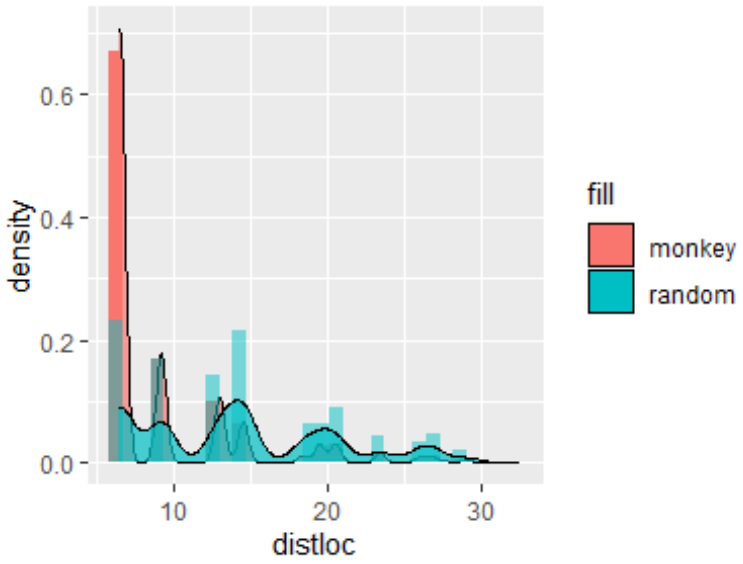
One thing we can look at is the spatial variance between successive choices (here I do not differentiate Miss, Correct or repeat).

The figures below show the distributions of euclidean distances between 2 successive choices. The absolute distance between 2 holes (vertically and horizontally is 6.5cm). So we can see harmonics at 6.5 cm approximately in the distributions. Something quite obvious is that the harmonic is stronger in TEST than in CONTROL.

Note that here every choice is counted even repeats.

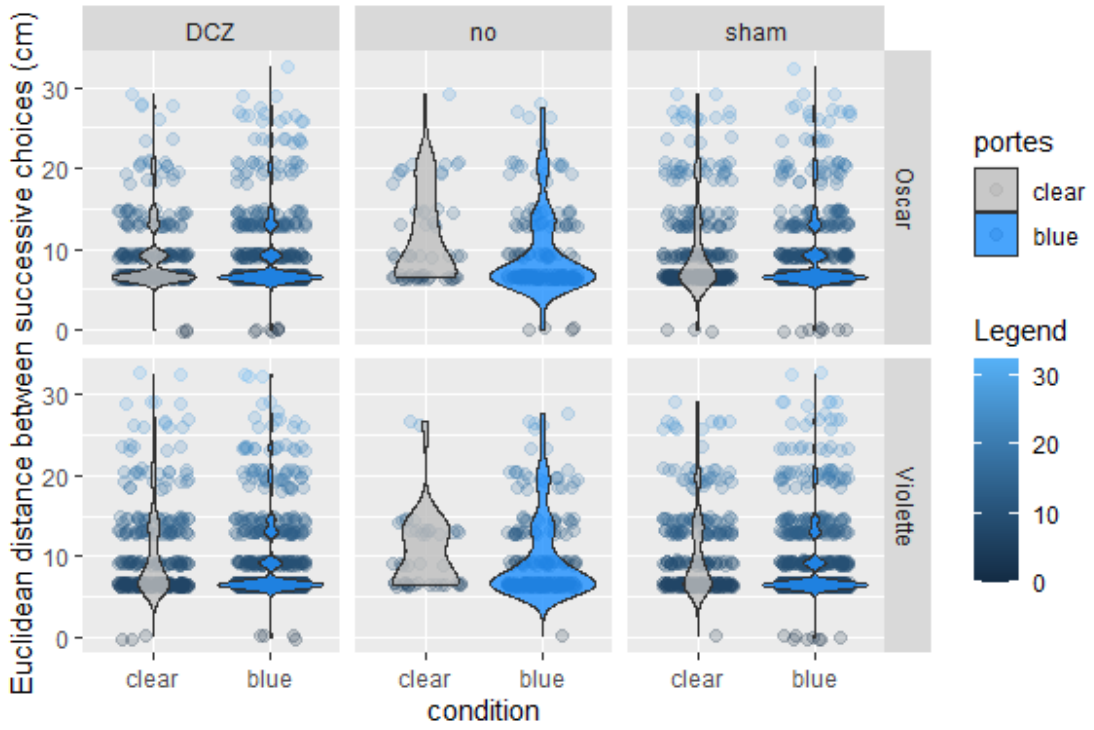


The distribution of distances between choices is of a particular form, somewhat LogNormal. We can look at this distributions depending on conditions and also compare with a random sampling of distances. Let's first look at this accross the 2 monkeys.



The red shows distributions for monkeys, and blue shows a random sampling of 10000 distances.

Seperated for the 2 animals for Clear and Blue conditions we can see (below) that the distributions and the oscillation effects are stronger on Blue compared to Clear.



Spatial strategy. Distributions of euclidean distances

We test the difference in distributions between clear and blue for the 2 animals separately - we exclude DCZ sessions:

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  data4B$distloc[data4B$singe == "Oscar" & data4B$portes == "clear" &
data4B$Injection != "DCZ"] and data4B$distloc[data4B$singe == "Oscar" &
data4B$portes == "blue" & data4B$Injection != "DCZ"]
## D = 0.085628, p-value = 0.05027
## alternative hypothesis: two-sided

##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  data4B$distloc[data4B$singe == "Violette" & data4B$portes ==
"clear" & data4B$Injection != "DCZ"] and data4B$distloc[data4B$singe ==
"Violette" & data4B$portes == "blue" & data4B$Injection != "DCZ"]
## D = 0.17064, p-value = 4.494e-07
## alternative hypothesis: two-sided
```

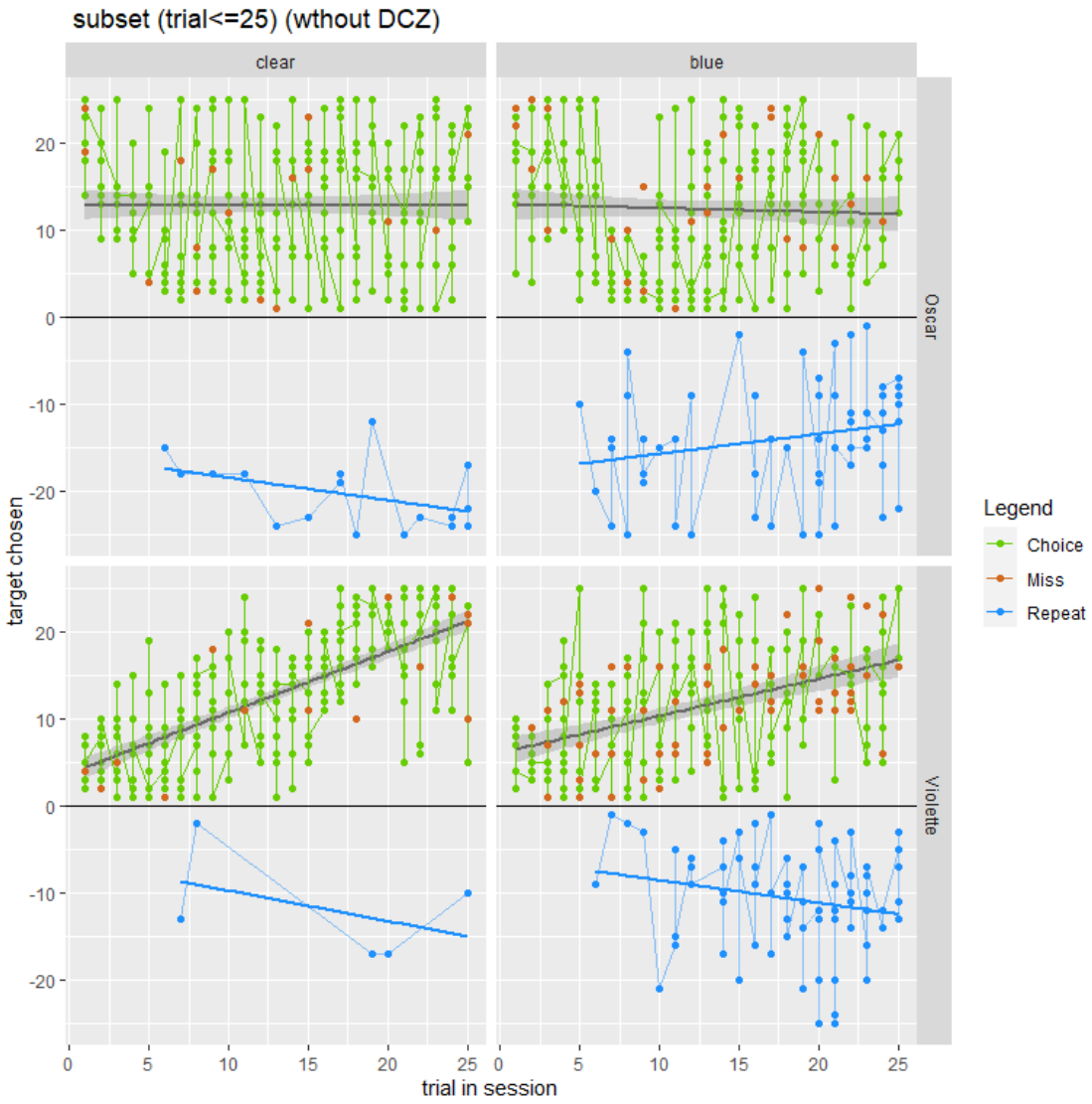
The KS tests indicate a difference between the 2 distributions (Clear and Blue) for both monkeys.

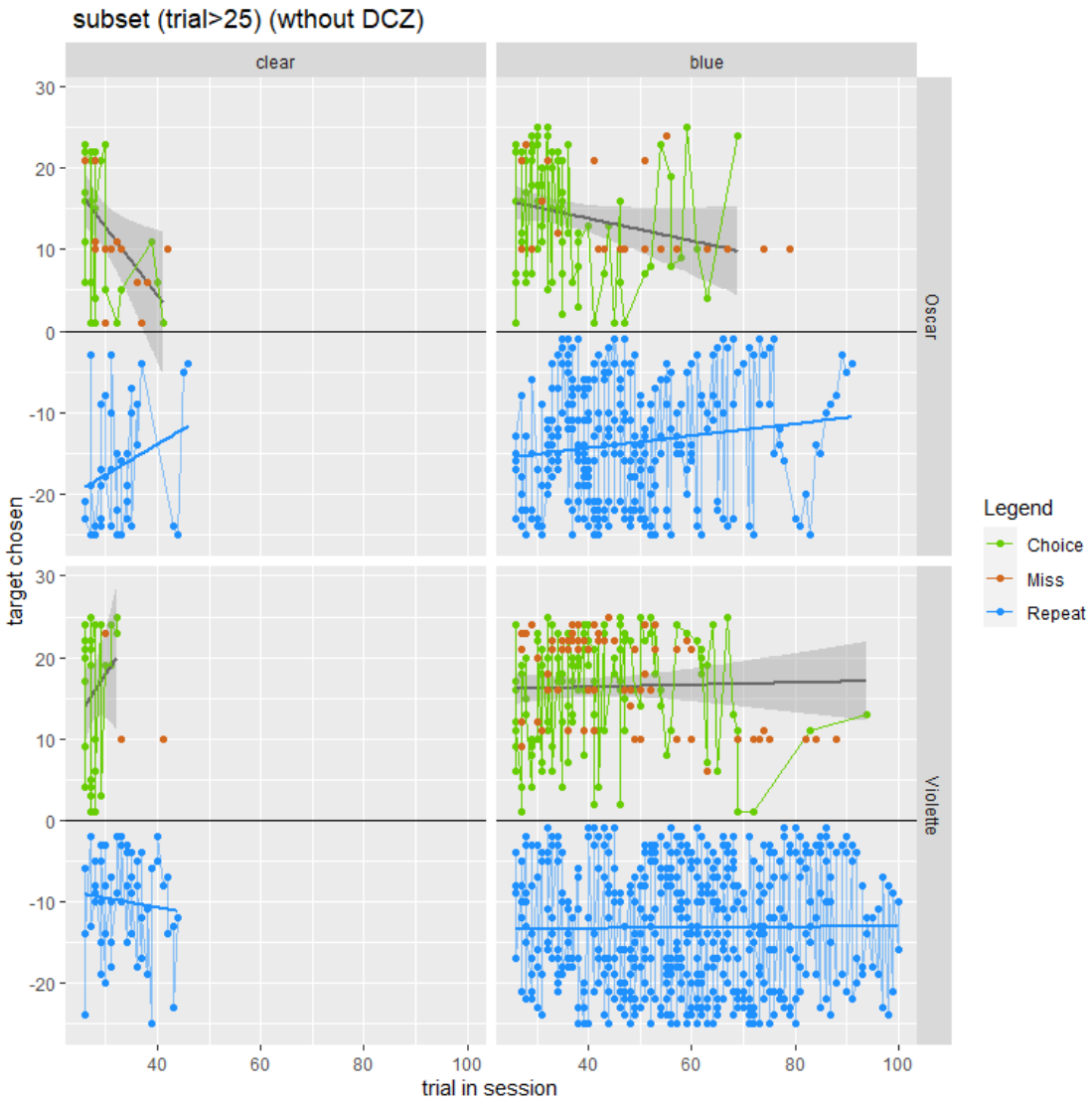
The strength of the 'harmonic' could be a marker of the two strategies used in the different sessions. The heavier harmonics in TEST could reflect an increased number of jumps between distant targets, whereas in CONTROL the animal would be more attracted to the visible reward which is just closeby to the current choice, hence proportionally more cases in which the animal choose the target just next the current one (6.5 cm distance). But we would need more control sessions to be sure.

Subset of trials

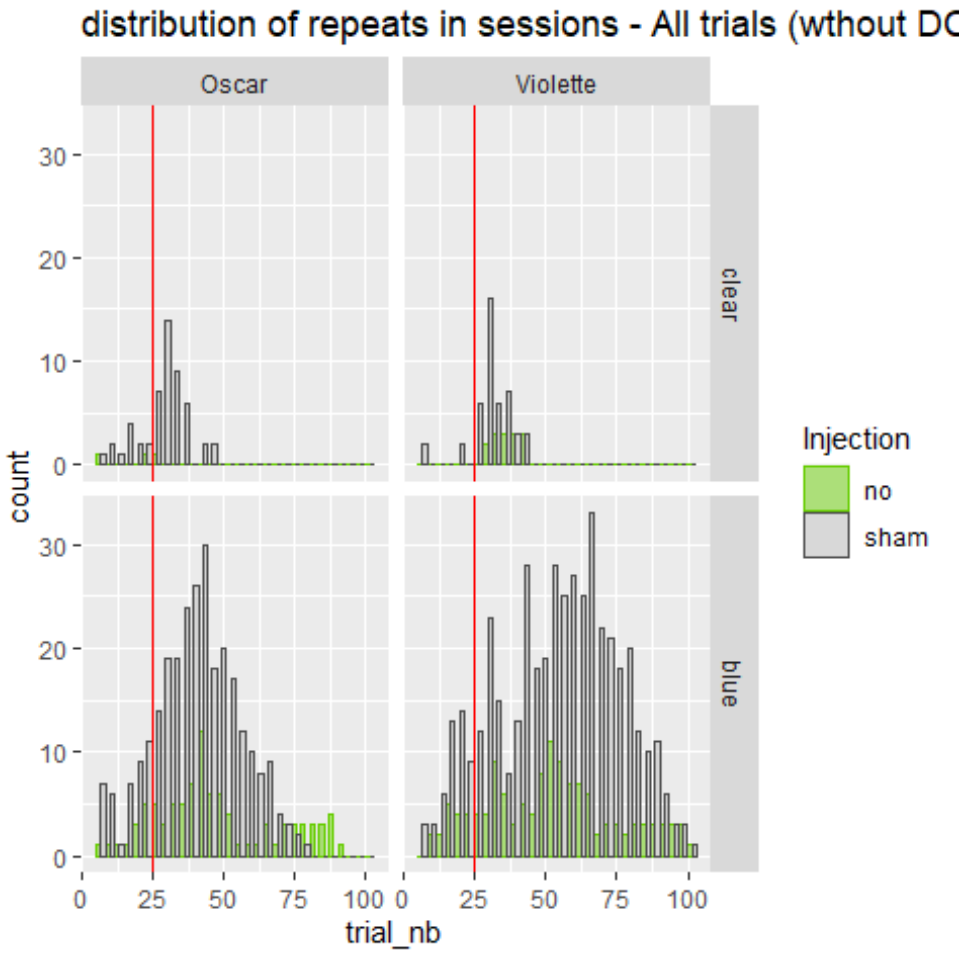
One observation is that monkeys often behave in a somewhat more controlled, organized, manner at the beginning of a session and then choices become more dispersed. This could correlate approximately with the completion of the task (i.e. having gone through all locations). So here we seperate the first 25 from the oher trials in 2 subset (25 corresponding to the number of locations on the setup).

Again here we remove DCZ (DCZ is taken into account in statistical analyses below)


subset (trial<=25) (wthout DCZ)
clear
blue
Oscar
Violette
target chosen
trial in session
Legend
Choice
Miss
Repeat

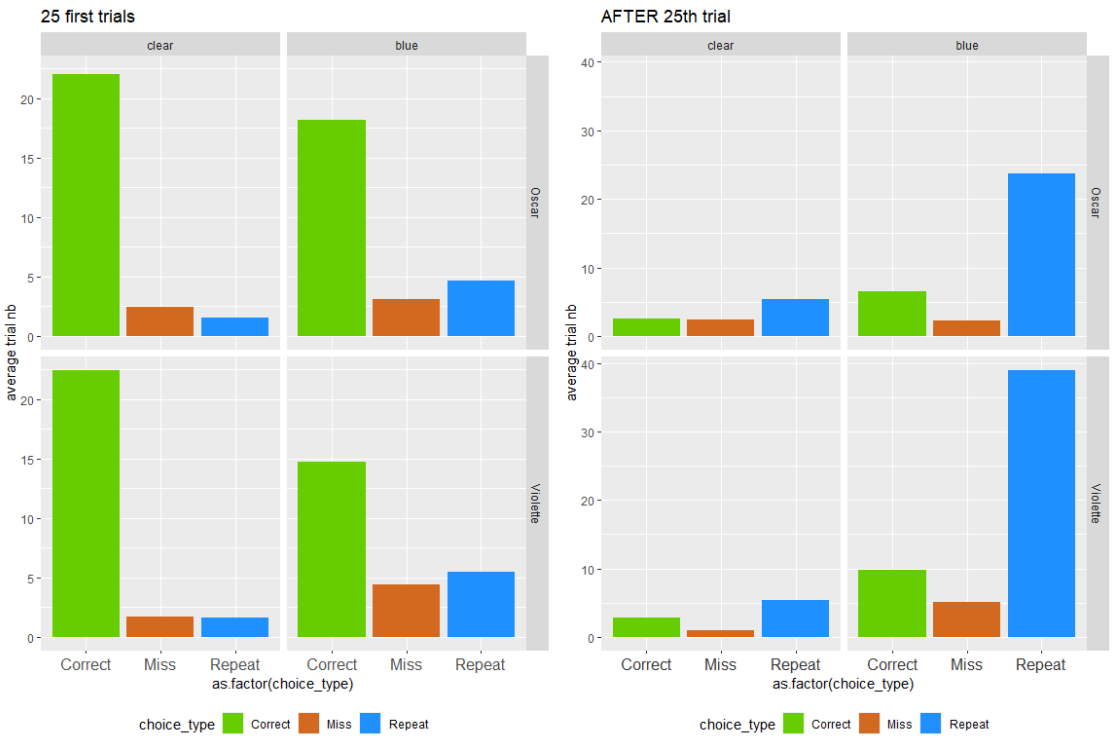


There is obviously a lot of repeats (checks?) after 25 when the animals continue trying to get rewards, and of course especially in the blue sessions. And VIOLET is a particularly good checker…



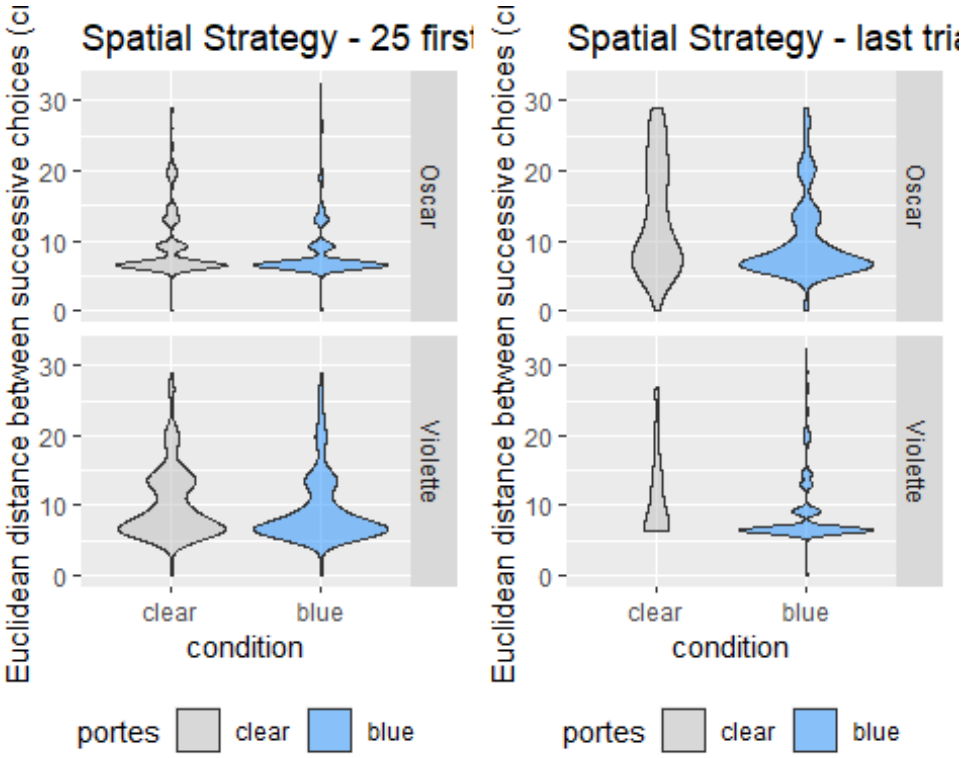
##The summary:

Below the average number of each trial type for the different conditions of injection (here again for the first 25 trials) :



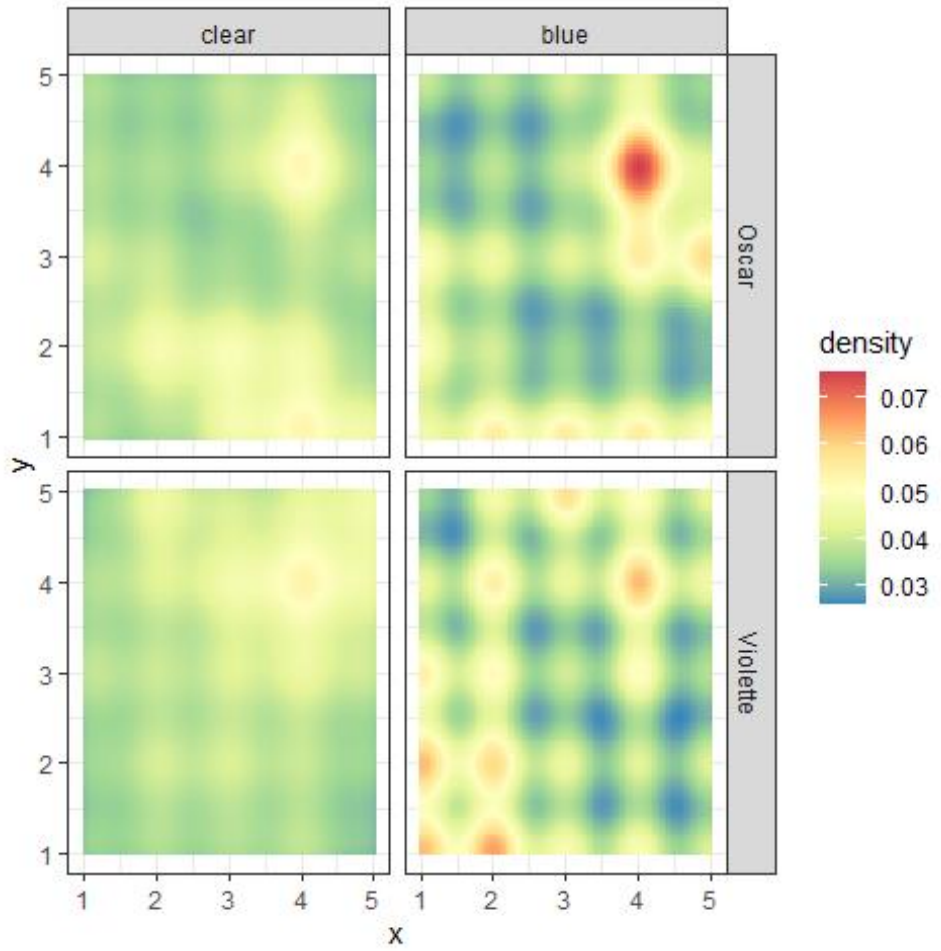
Injection type 25 trials

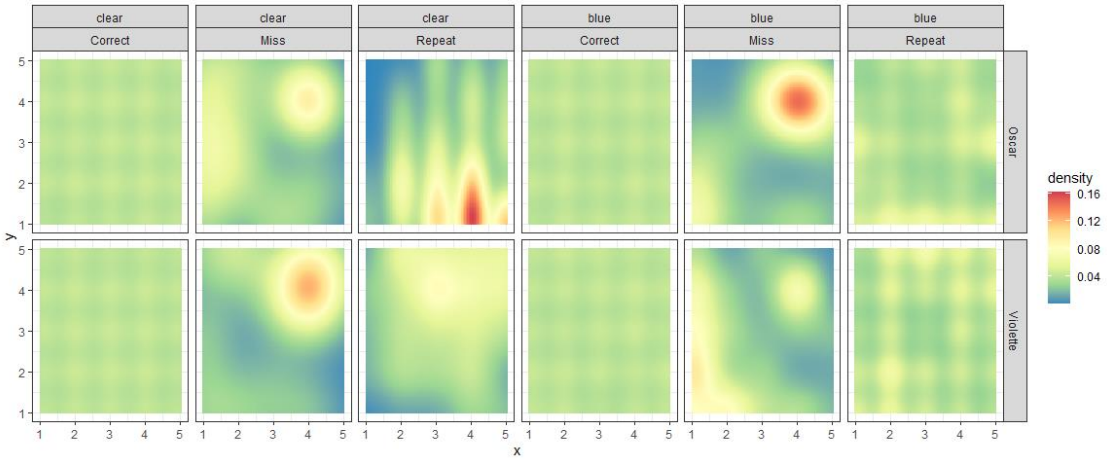
##Spatial strategy:



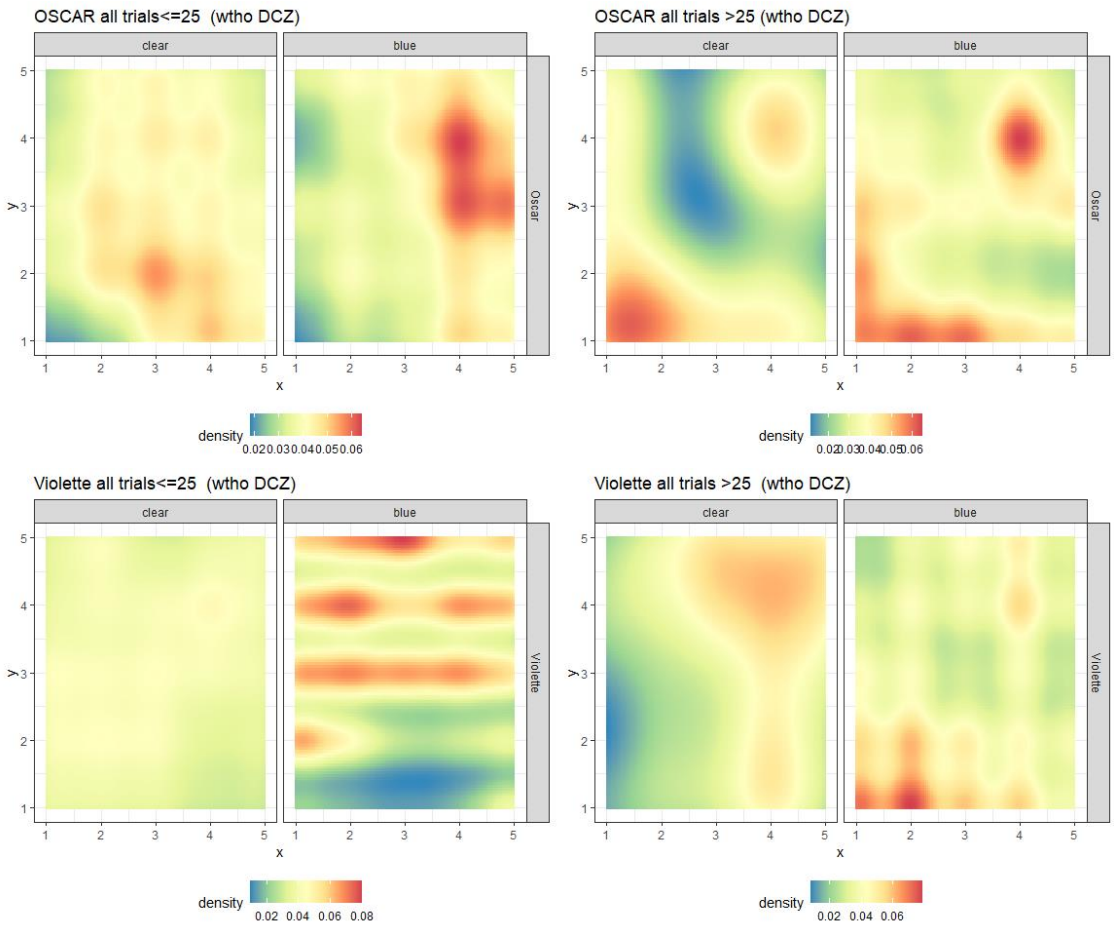
Spatial organization in 2D space

The tendencies for choosing some locations rather than others can reveal spatial biases. SO we use here a 2D density mapping of choices to look at that.





Let's look also at the patterns of choices separating between before and after 25:



STATISTICAL ANALYSES on DCZ vs Sham sessions

(Note we have no injection pre-surgery)

Here are the analyses and description of data for the 2 main types of sessions used on DCZ conditions. We subset the data for just the 2 monkeys and the 2 session types with an injection (sham and DCZ). We will also go through some more measures:

- test the numbers of repeats, and length distribution of distance to repeat
- test the number of pauses (code 99)

ATTENTION we remove the CONTROL sessions!!!!

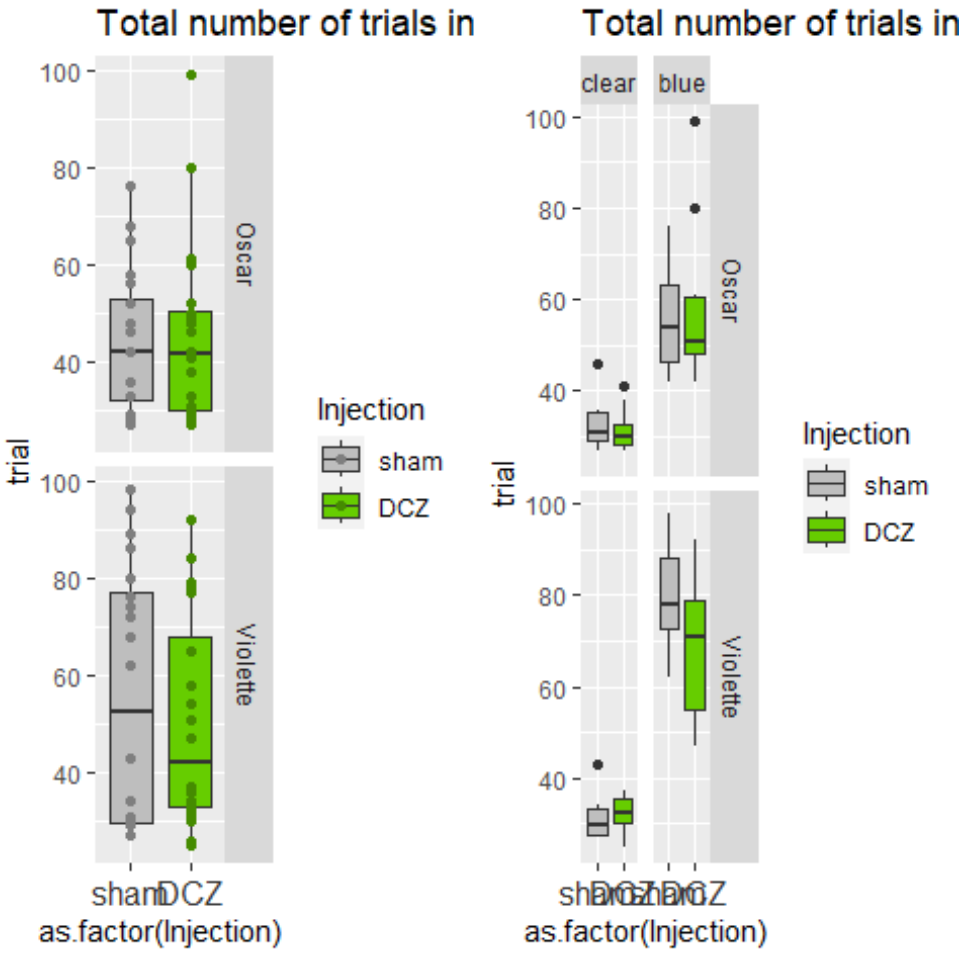
```
## , ,  = Oscar
##
##
##         sham DCZ
##   clear   10  10
##   blue    10  10
##
## , ,  = Violette
```

```
##
##
##         sham DCZ
##   clear   10  10
##   blue    10  10
```

##Descriptions of sessions

Here we answer a few general questions on the sessions, choices, repeats etc..

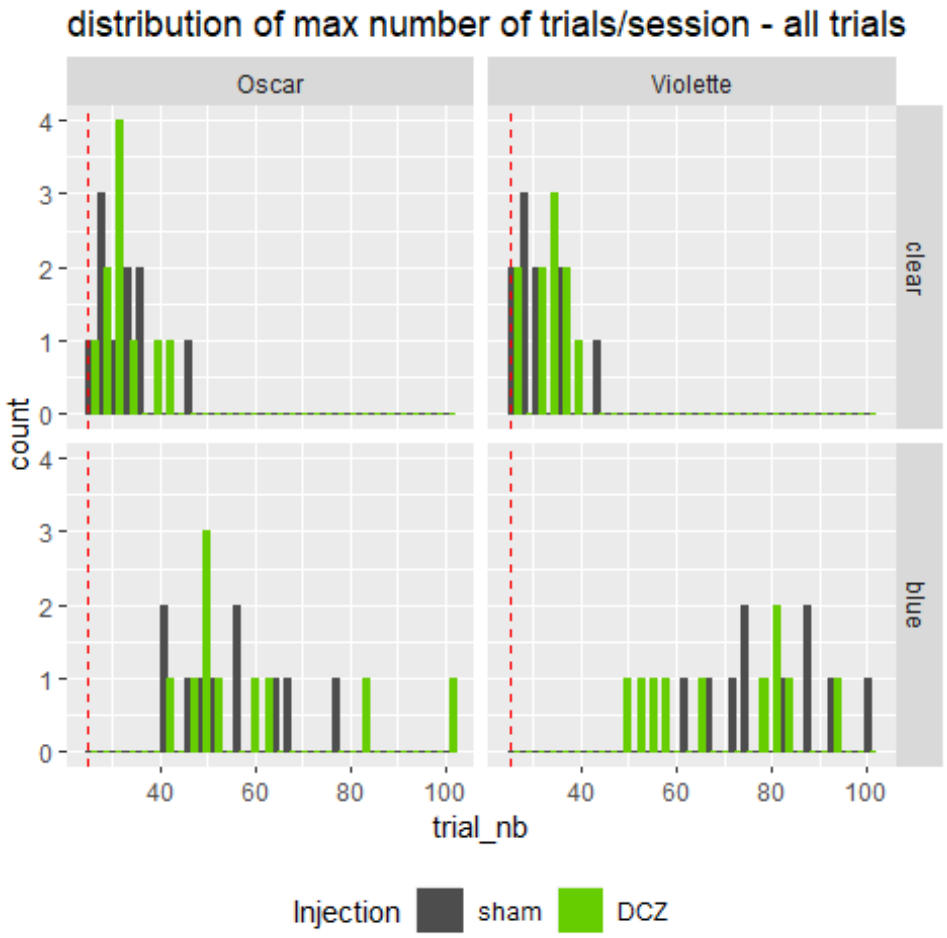
First, were there more trials (choices+misses+repeats) in DCZ compared to sham sessions?

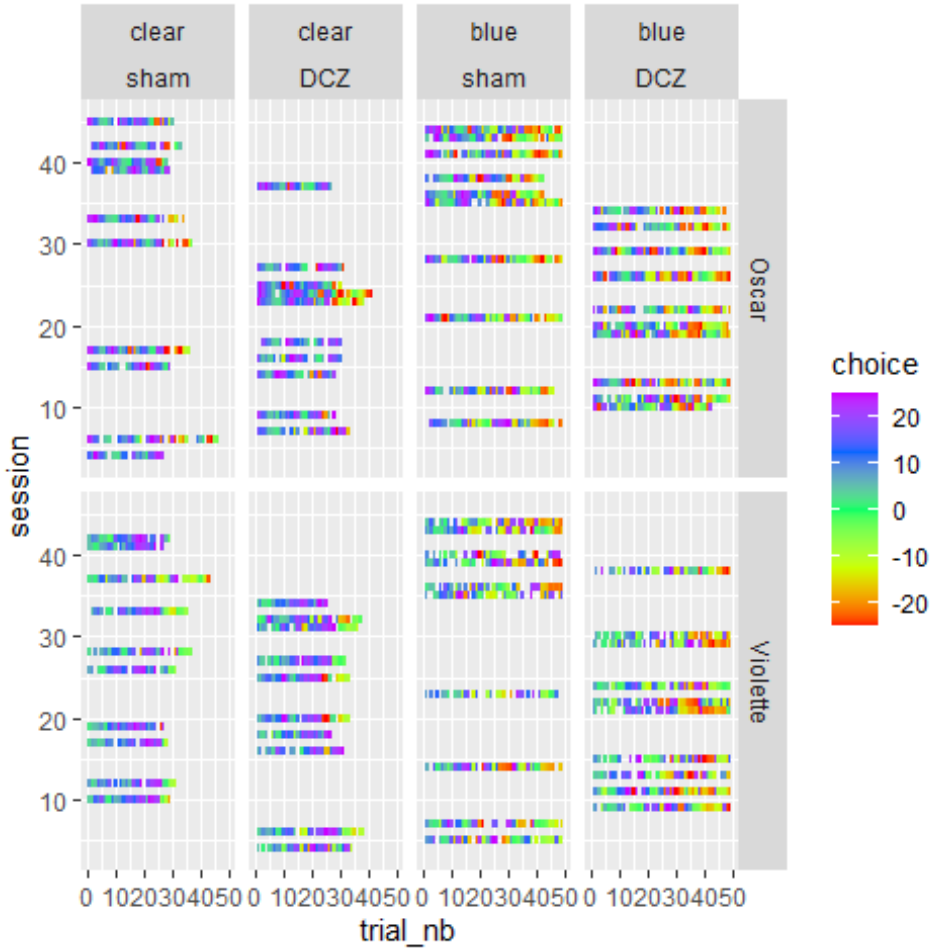


Summary 25 trials

```
##
## Call:
## glm(formula = trial ~ Injection * portes, family = "poisson",
##     data = subset(agg.data4B, singe == "Oscar"))
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                3.48431    0.05538  62.911  < 2e-16 ***
```

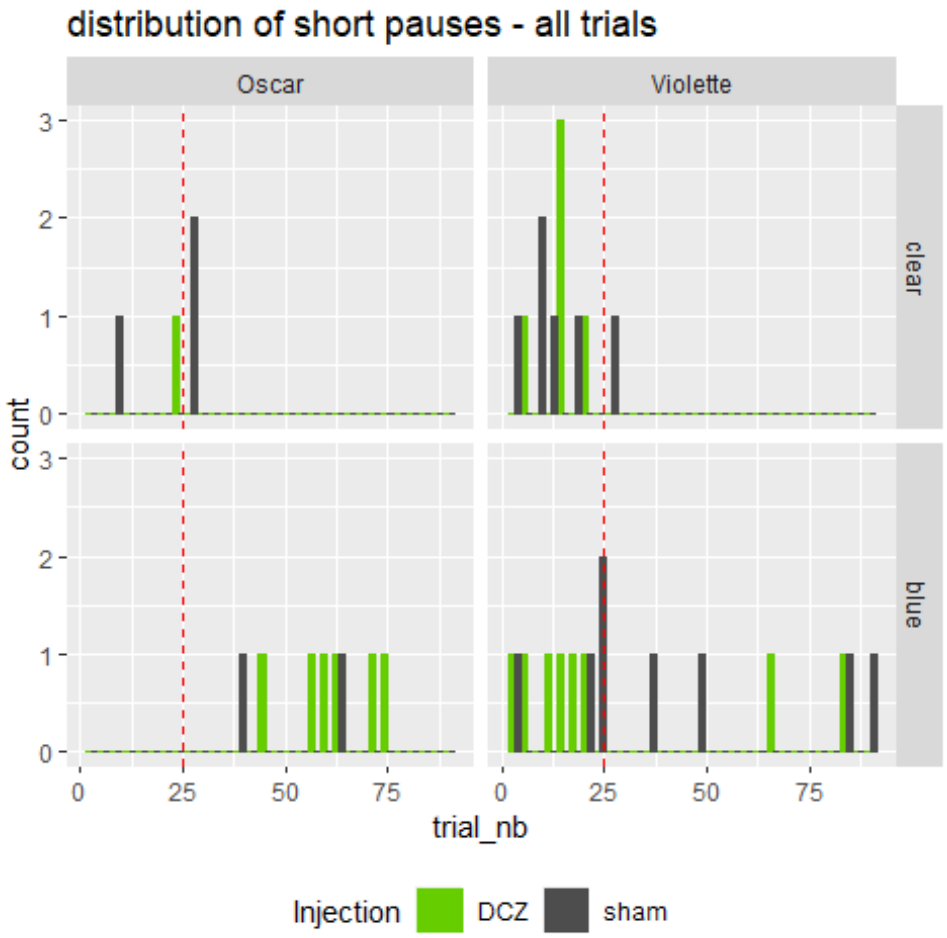
```
## InjectionDCZ            -0.03432    0.07901  -0.434    0.664
## portesblue               0.52846    0.06983   7.568 3.79e-14 ***
## InjectionDCZ:portesblue  0.09399    0.09876   0.952    0.341
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 222.575  on 39  degrees of freedom
## Residual deviance:  79.673  on 36  degrees of freedom
## AIC: 310.8
##
## Number of Fisher Scoring iterations: 4

##
## Call:
## glm(formula = trial ~ Injection * portes, family = "poisson",
##     data = subset(agg.data4B, singe == "Violette"))
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              3.43721    0.05670  60.616   <2e-16 ***
## InjectionDCZ             0.03165    0.07957   0.398    0.691
## portesblue               0.94357    0.06684  14.118   <2e-16 ***
## InjectionDCZ:portesblue -0.18559    0.09509  -1.952    0.051 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 420.995  on 39  degrees of freedom
## Residual deviance:  59.087  on 36  degrees of freedom
## AIC: 295.37
##
## Number of Fisher Scoring iterations: 4
```

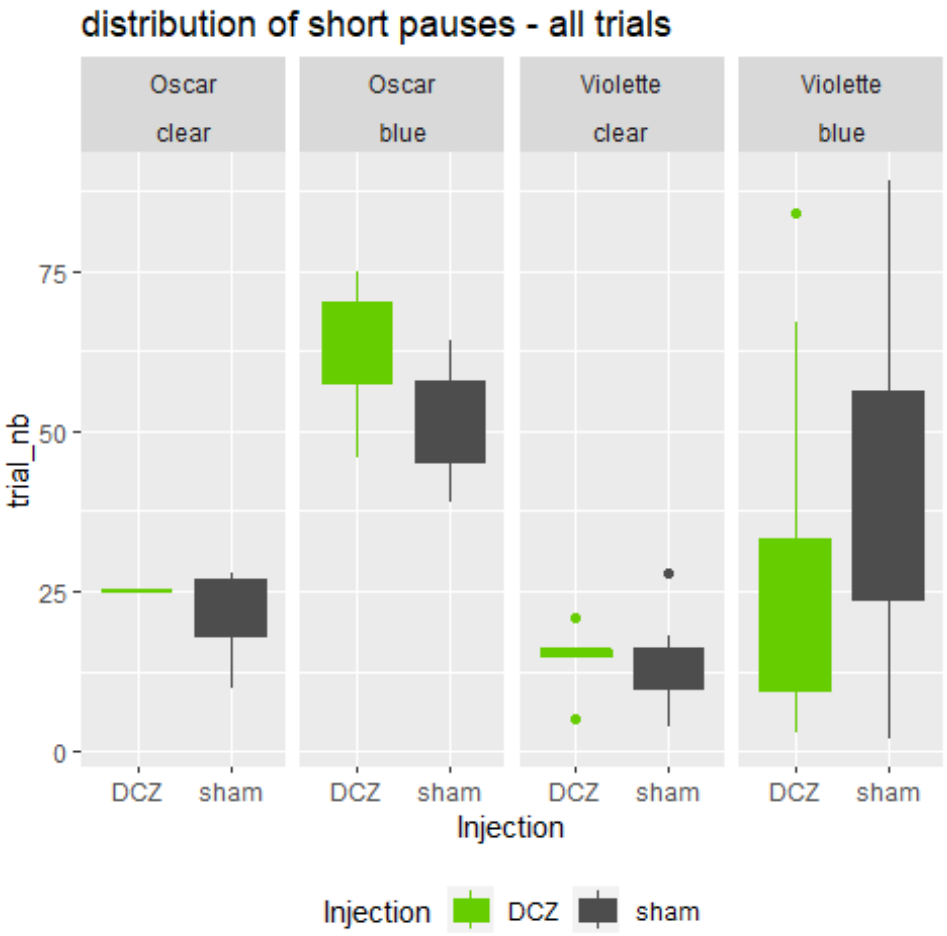

distribution of max number of trials/session - all trials
Oscar
Violette
clear
blue
count
trial_nb
0
1
2
3
4
40
60
80
100
Injection
sham
DCZ

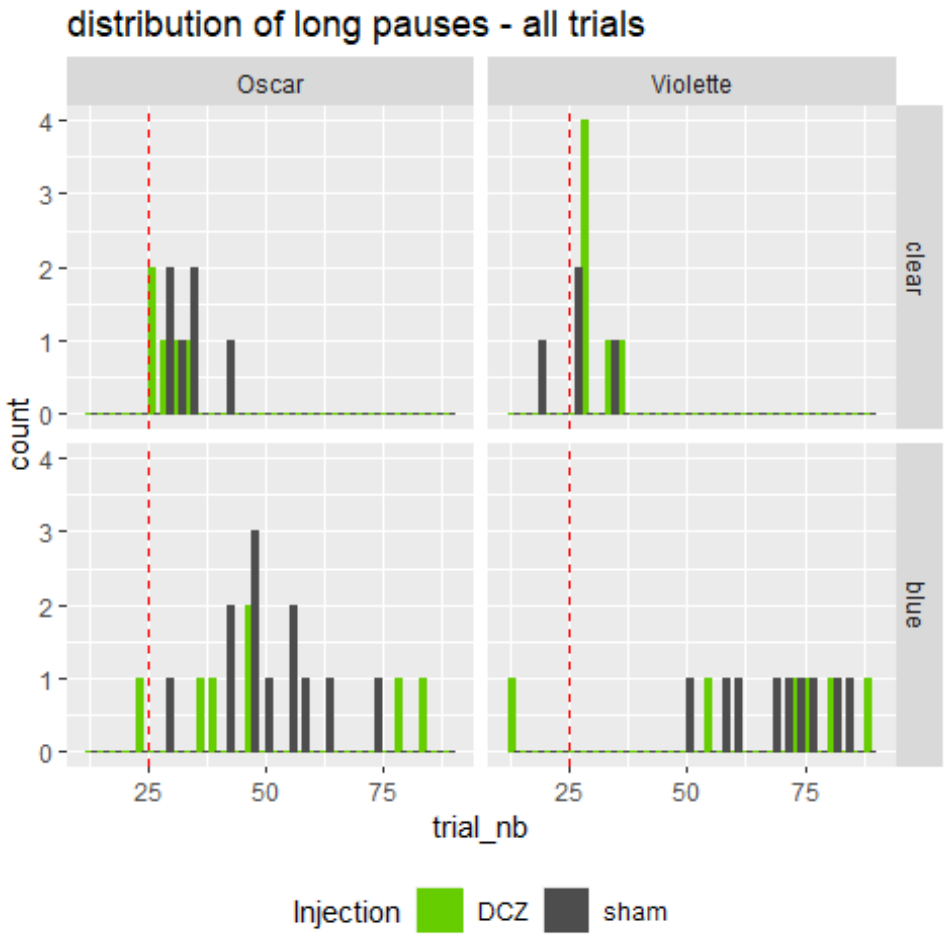


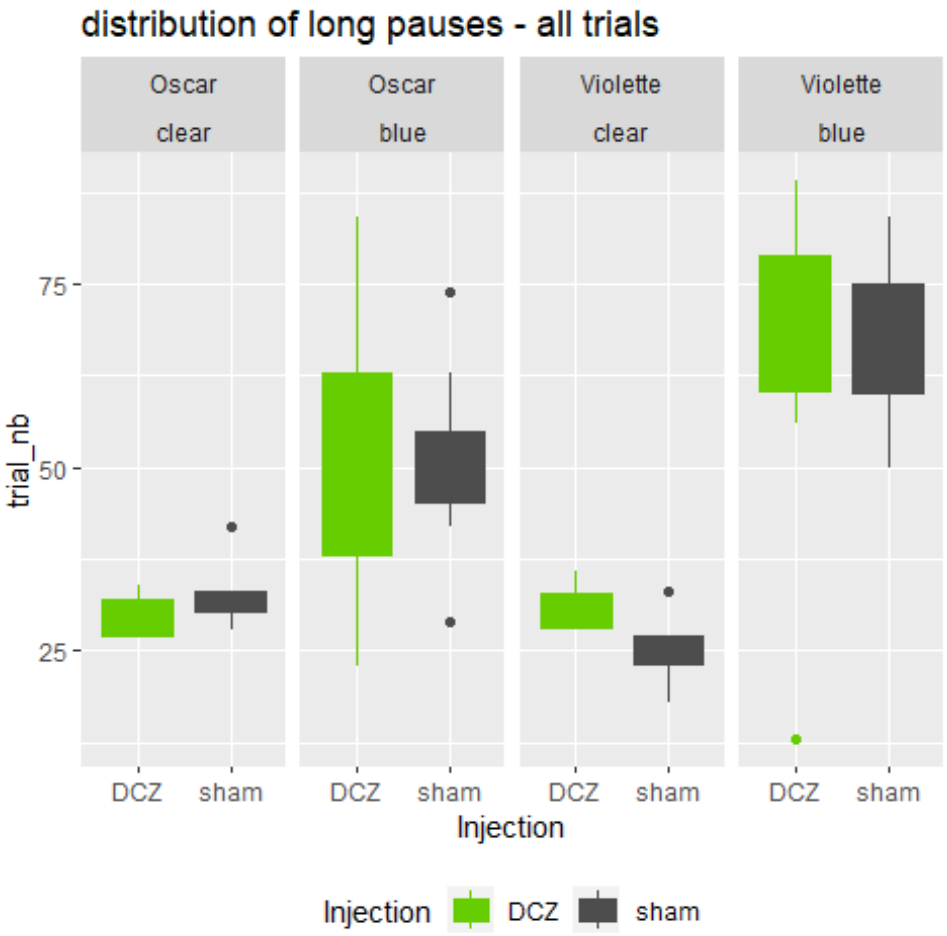
No, apparently no main effect for Injection on the length of sessions. We have a significant interaction for OSCAR, suggesting a lower number of trials in DCZ (shorter sessions) in the blue door condition.

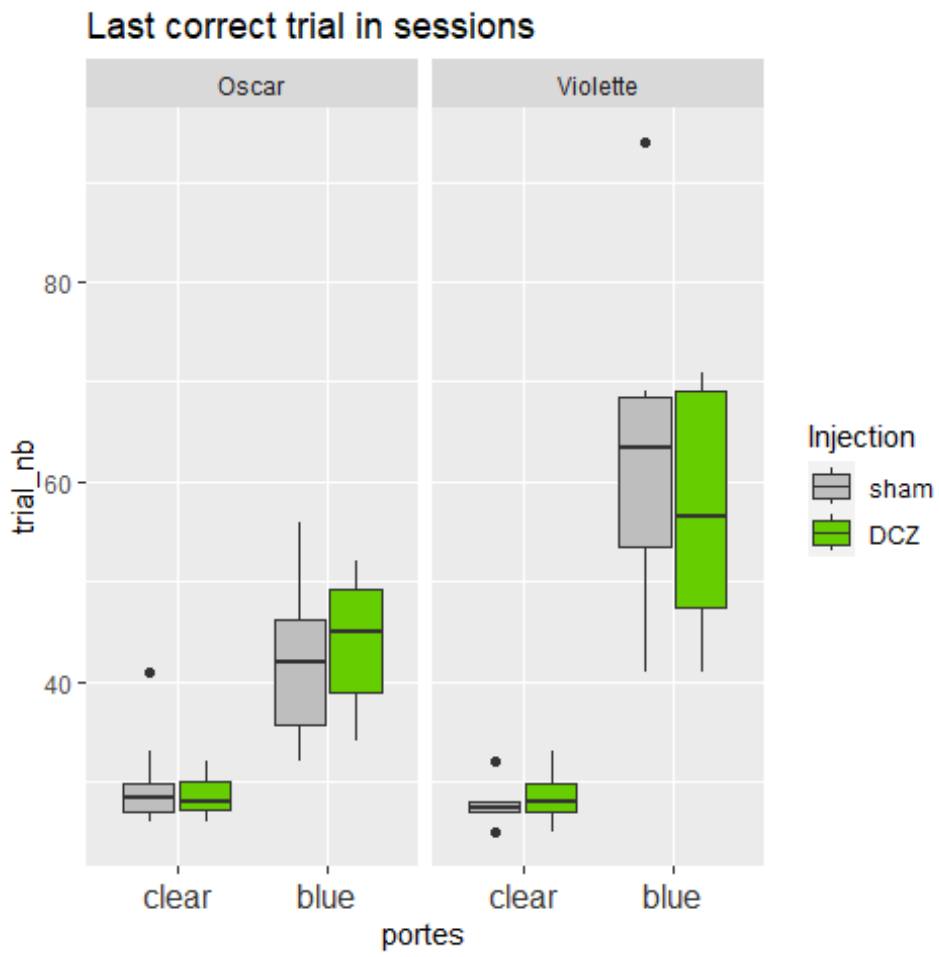
Then we ask whether the last correct trial performed is later in DCZ than in sham: this would mean that monkeys have more problems, or take more time, to find all or the max of rewards.


distribution of short pauses - all trials
Oscar
Violette
clear
blue
count
trial_nb
0
25
50
75
1
2
3
Injection
DCZ
sham


distribution of short pauses - all trials
Oscar
clear
Oscar
blue
Violette
clear
Violette
blue
trial_nb
75
50
25
0
DCZ
sham
DCZ
sham
DCZ
sham
DCZ
sham
Injection
Injection
DCZ
sham


distribution of long pauses - all trials
Oscar
Violette
clear
blue
count
trial_nb
0
1
2
3
4
25
50
75
Injection
DCZ
sham


distribution of long pauses - all trials
Oscar
clear
Oscar
blue
Violette
clear
Violette
blue
trial_nb
75
50
25
DCZ
sham
DCZ
sham
DCZ
sham
DCZ
sham
Injection
Injection
DCZ
sham



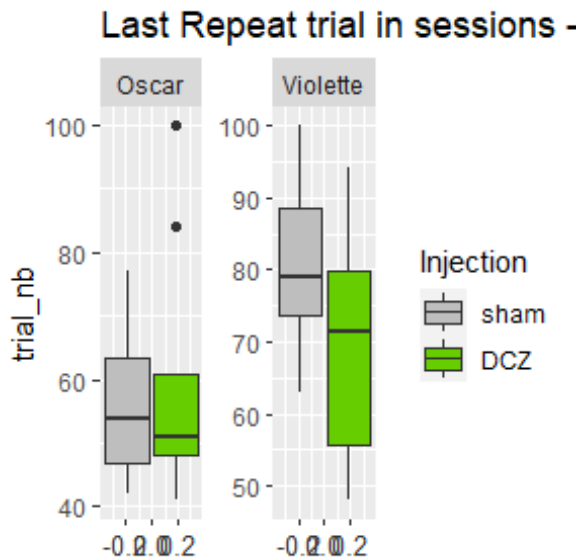

```
##
## Call:
## glm(formula = trial_nb ~ Injection * portes, family = "poisson",
##     data = subset(agg.maxcor, singe == "Oscar"))
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               3.39115    0.05803  58.442  < 2e-16 ***
## InjectionDCZ             -0.03774    0.08285  -0.456    0.649
## portesblue                0.35364    0.07570   4.671 2.99e-06 ***
## InjectionDCZ:portesblue   0.07714    0.10724   0.719    0.472
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 82.751  on 39  degrees of freedom
## Residual deviance: 27.646  on 36  degrees of freedom
## AIC: 251.74
##
## Number of Fisher Scoring iterations: 4
```

```
##
## Call:
## glm(formula = trial_nb ~ Injection * portes, family = "poisson",
##     data = subset(agg.maxcor, singe == "Violette"))
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                3.33577    0.05965  55.918   <2e-16 ***
## InjectionDCZ               0.01413    0.08407   0.168    0.866
## portesblue                 0.80100    0.07181  11.155   <2e-16 ***
## InjectionDCZ:portesblue -0.10260    0.10203  -1.006    0.315
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 288.673  on 39  degrees of freedom
## Residual deviance:  54.146  on 36  degrees of freedom
## AIC: 284.06
##
## Number of Fisher Scoring iterations: 4
```

Here we have an interaction for OSCAR and VIOLET (although in different directions): - OSCAR , an earlier last correct under DCZ trial than in sham session and clear session too maybe. - VIOLET , slightly later last correct in blue DCZ compared to sham.

This might mean that trials are instead repeats (since we have the same number of trials in sessions). So let see let's do the same analysis for Repeats, and then analyze the Repeats altogether

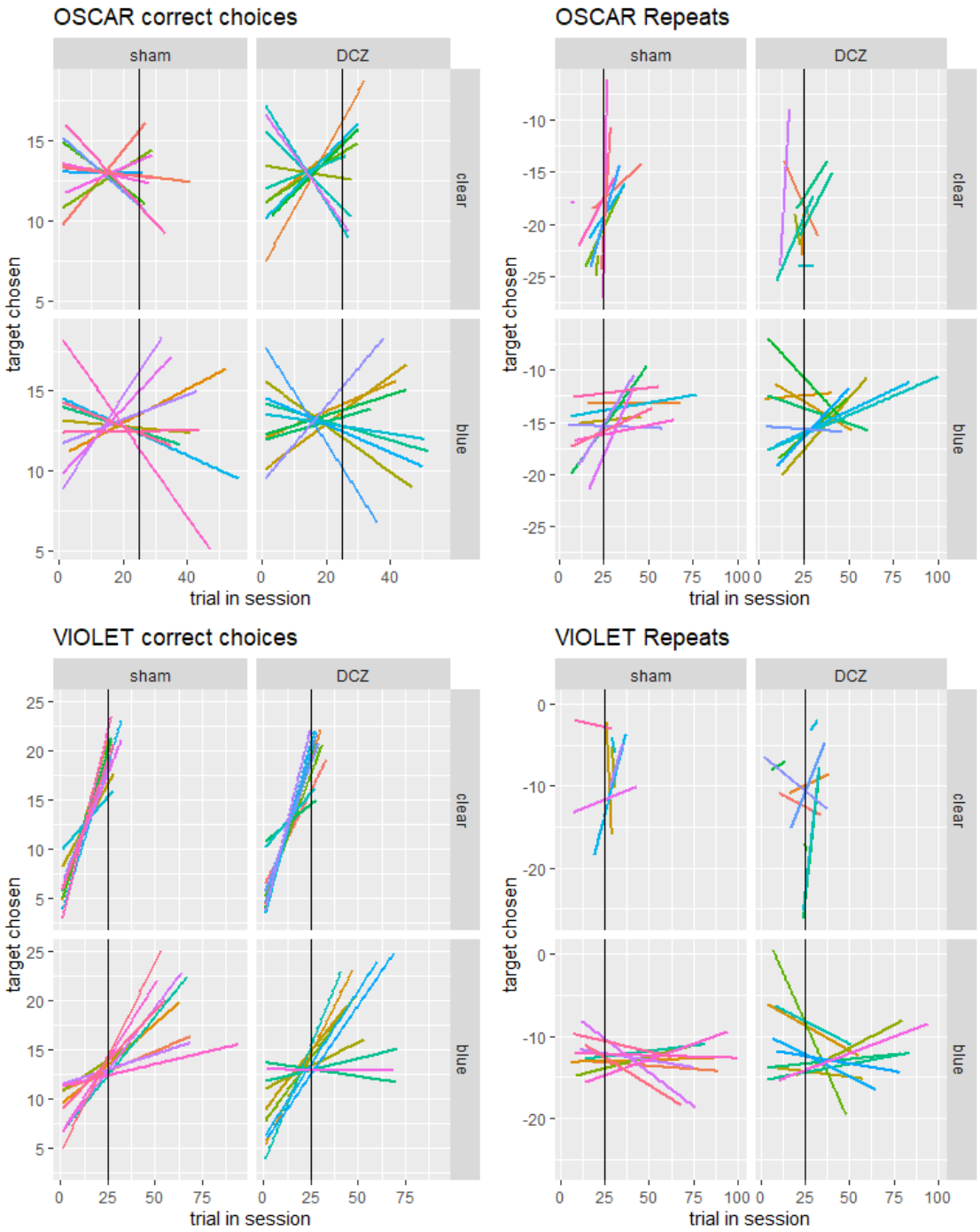
In the first analysis we take only the "blue" sessions because we have very few repeats in "clear":



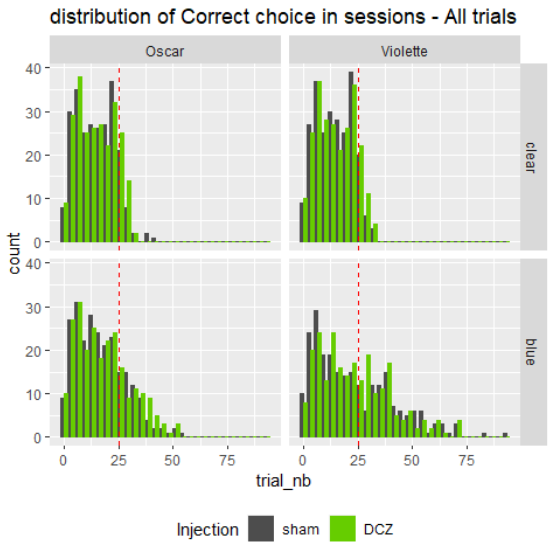
```
##
## Call:
## glm(formula = trial_nb ~ Injection, family = "poisson", data =
subset(agg.maxrpt,
##     singe == "Oscar"))
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.01638    0.04245  94.620   <2e-16 ***
## InjectionDCZ  0.06454    0.05908   1.092    0.275
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 70.860  on 19  degrees of freedom
## Residual deviance: 69.667  on 18  degrees of freedom
## AIC: 190.89
##
## Number of Fisher Scoring iterations: 4

##
## Call:
## glm(formula = trial_nb ~ Injection, family = "poisson", data =
subset(agg.maxrpt,
##     singe == "Violette"))
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.39074    0.03520 124.731   <2e-16 ***
## InjectionDCZ -0.15374    0.05181  -2.967    0.003 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 57.111  on 19  degrees of freedom
## Residual deviance: 48.279  on 18  degrees of freedom
## AIC: 175.02
##
## Number of Fisher Scoring iterations: 4
```

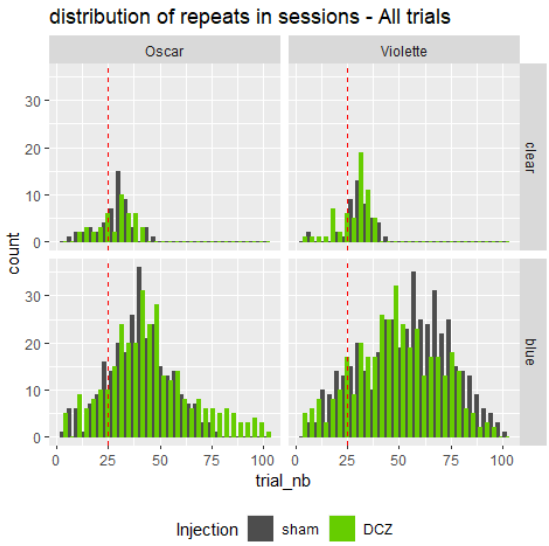
Main effect for both monkeys although, again, in opposite direction; in other words the rank of the last repeat trial differ in DCZ and sham: earlier for OSCAR, and later for VIOLET.

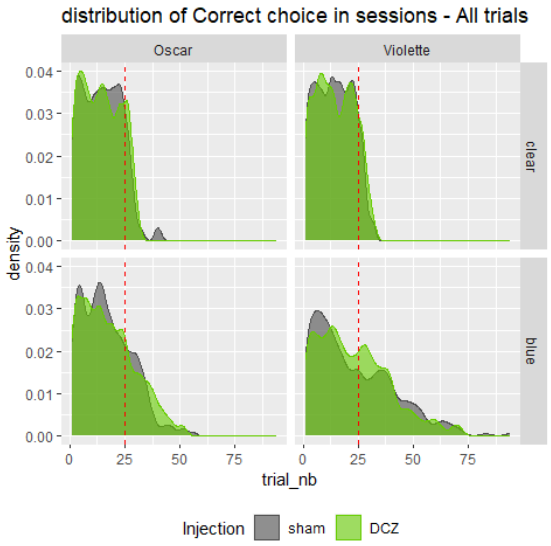
Below we can graph the overall trends of choices across sessions. The lines represent the fit to choice patterns (as shown in the very first figures). Positive slopes means searching holes from top to bottom, and negative slopes from bottom to top. The length actually covers the number of trials.

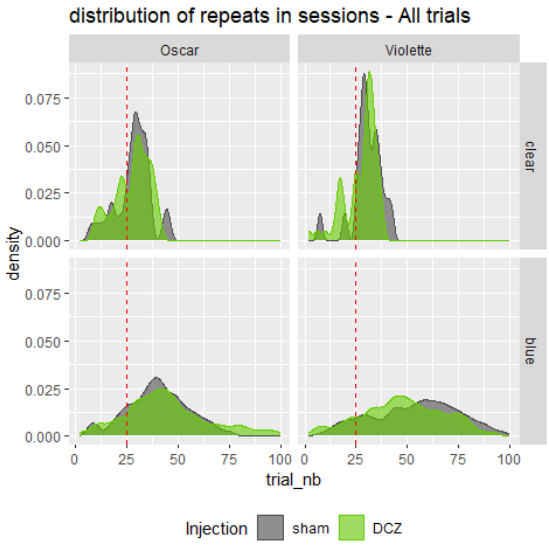


There are changes regarding repeats but these are different for the 2 monkeys. Let's test statistically the data for All trials, for <25 trials and for >25 trials:



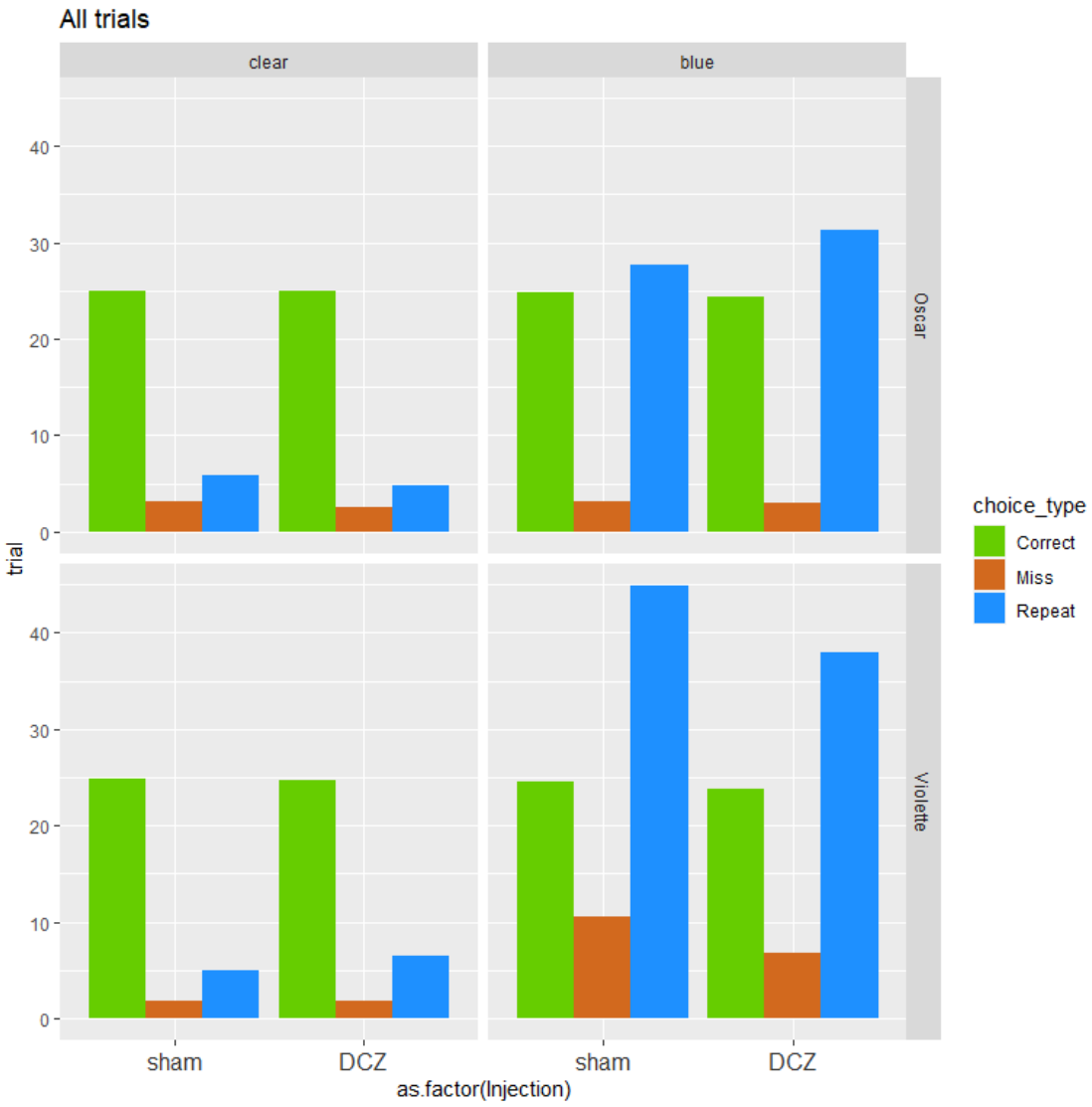


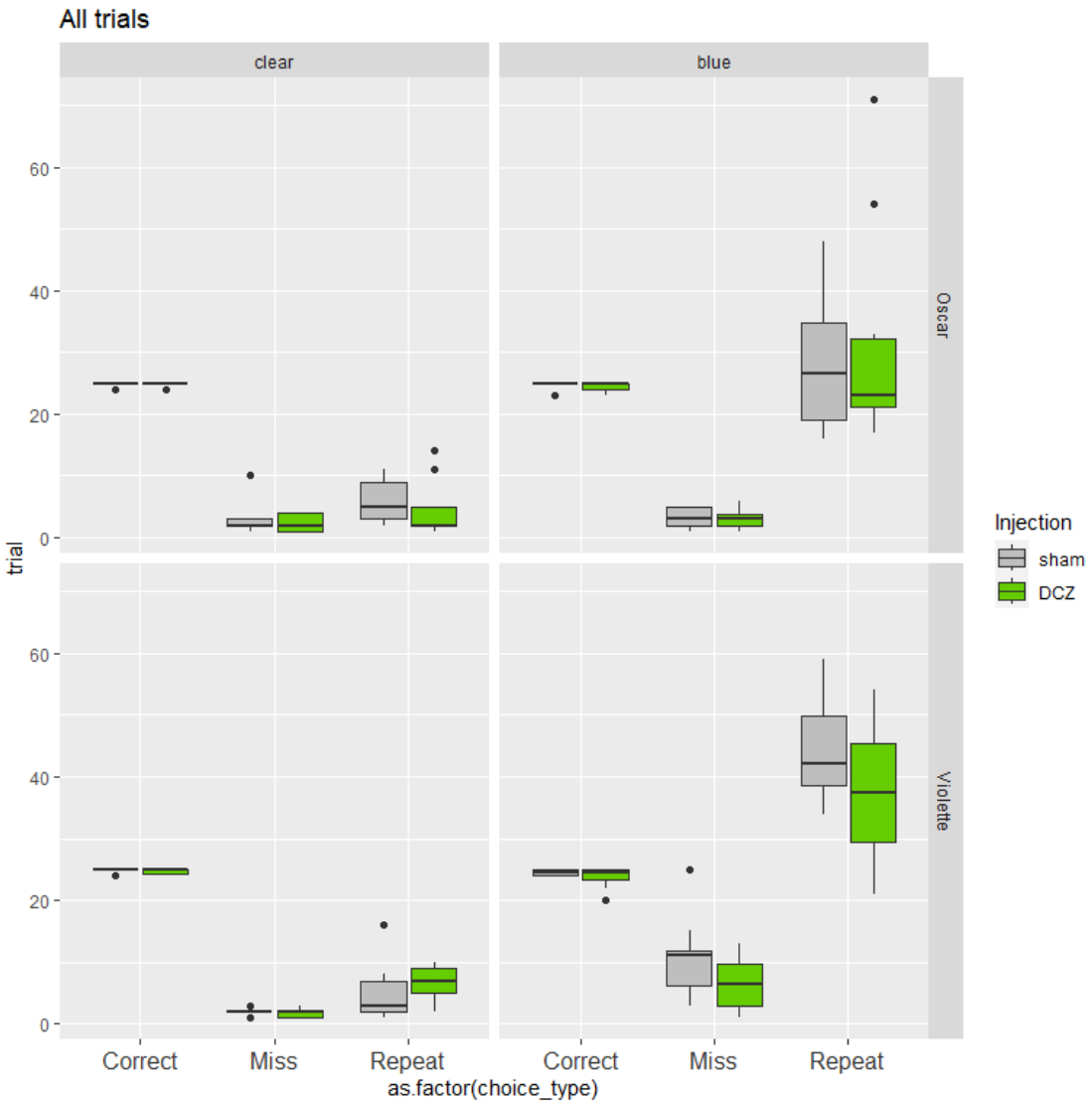


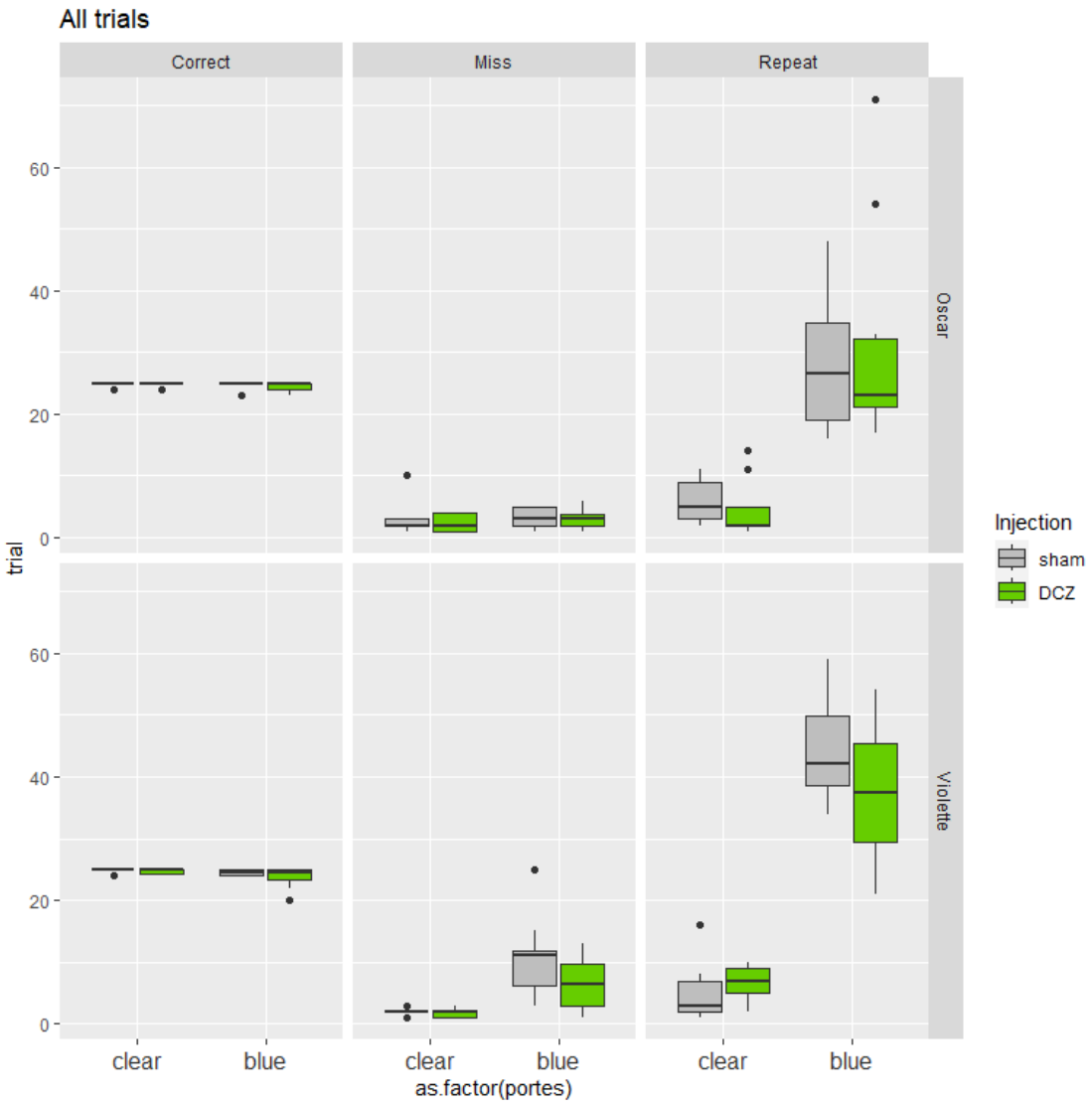


##Trial types:

First let's look at the frequency of repeats, miss, etc...


All trials
clear
blue
Oscar
Violette
trial
0
10
20
30
40
sham
DCZ
as.factor(Injection)
choice_type
Correct
Miss
Repeat

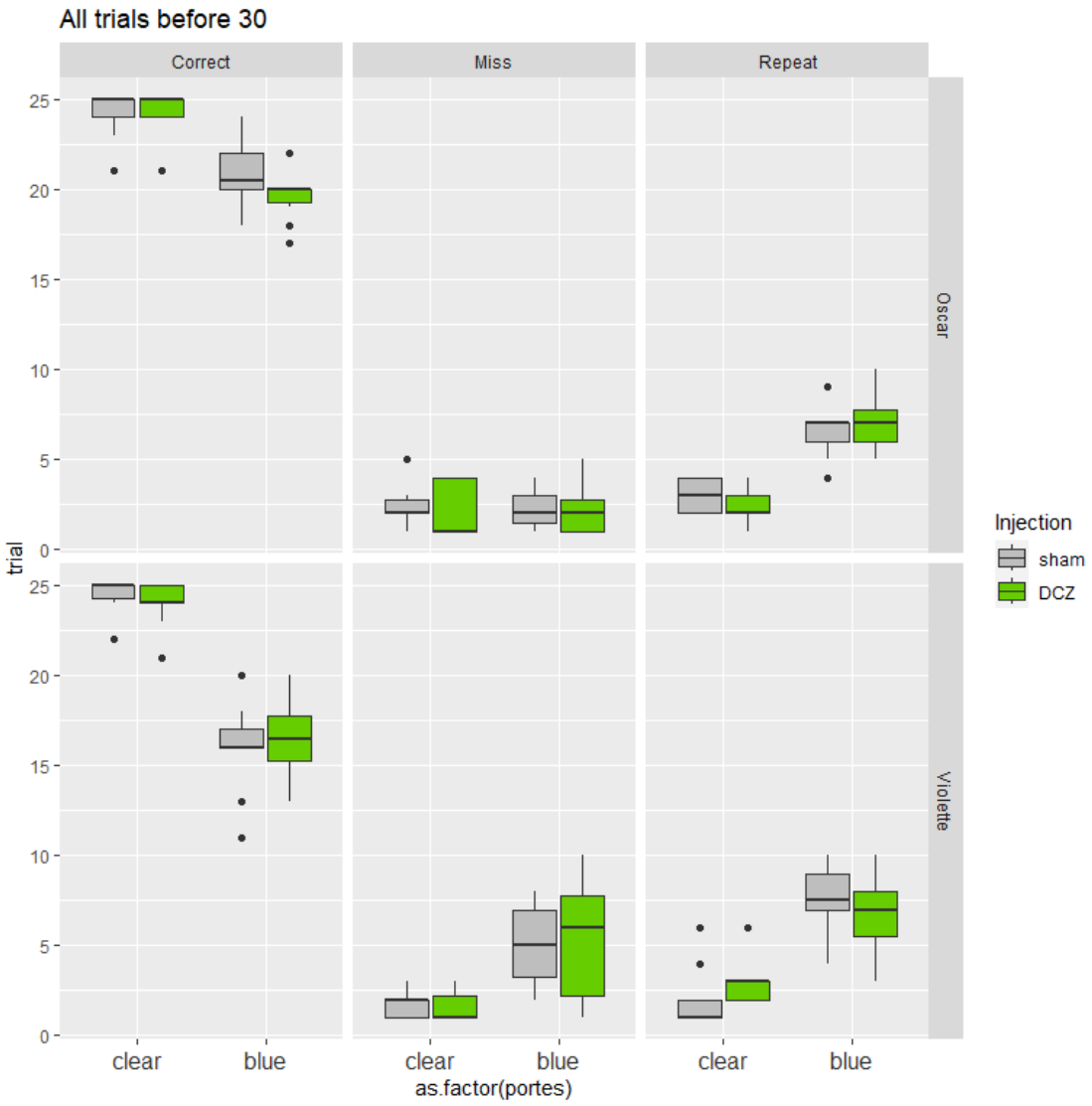

All trials
clear
blue
Oscar
Violette
trial
0
20
40
60
Correct
Miss
Repeat
as.factor(choice_type)
Injection
sham
DCZ




```
##
## Call:
## glm(formula = trial ~ choice_type/Injection, family = "poisson",
##     data = subset(agg.data4B, singe == "Oscar"))
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     3.212858   0.044856  71.626  < 2e-16 ***
## choice_typeMiss                -2.075779   0.144499 -14.365  < 2e-16 ***
## choice_typeRepeat              -0.361239   0.071075  -5.083 3.72e-07 ***
## choice_typeCorrect:InjectionDCZ -0.008081   0.063565  -0.127    0.899
## choice_typeMiss:InjectionDCZ    -0.111226   0.194257  -0.573    0.567
## choice_typeRepeat:InjectionDCZ   0.078873   0.076475   1.031    0.302
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
```

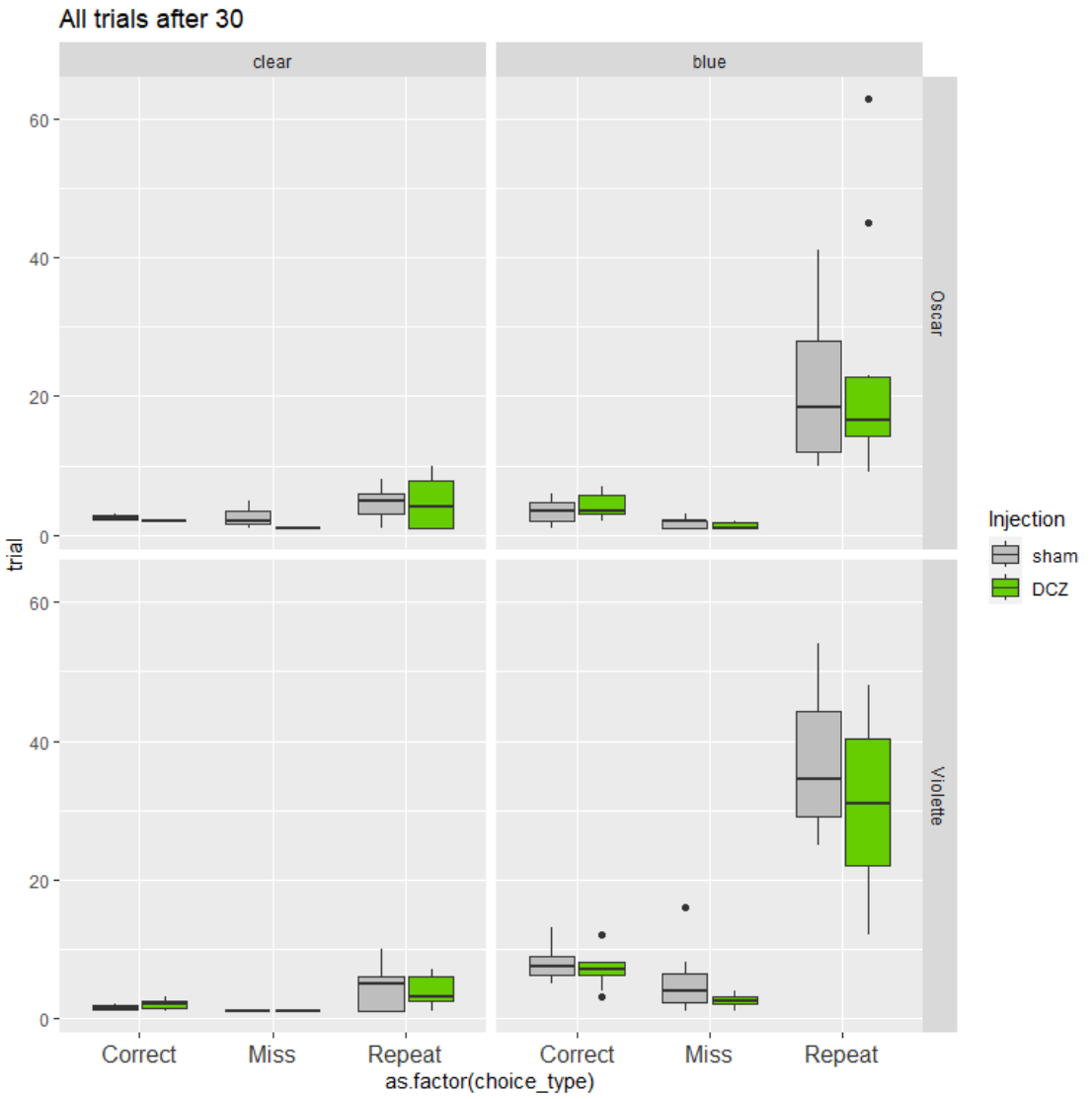
```
##
##     Null deviance: 1298.38  on 113  degrees of freedom
## Residual deviance:  543.72  on 108  degrees of freedom
## AIC: 1021.2
##
## Number of Fisher Scoring iterations: 5
```

```
##
## Call:
## glm(formula = trial ~ choice_type/Injection, family = "poisson",
##     data = subset(agg.data4B, singe == "Violette"))
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      3.20680    0.04499  71.275   <2e-16 ***
## choice_typeMiss                 -1.34722    0.10110 -13.326   <2e-16 ***
## choice_typeRepeat                0.05129    0.06363   0.806   0.4202
## choice_typeCorrect:InjectionDCZ -0.01839    0.06392  -0.288   0.7736
## choice_typeMiss:InjectionDCZ    -0.33111    0.14228  -2.327   0.0200 *
## choice_typeRepeat:InjectionDCZ  -0.12032    0.06563  -1.833   0.0668 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1425.76  on 114  degrees of freedom
## Residual deviance:  814.58  on 109  degrees of freedom
## AIC: 1317.9
##
## Number of Fisher Scoring iterations: 5
```




```
##
## Call:
## glm(formula = trial ~ Injection * portes, family = "poisson",
##     data = subset(agg.data4B.first30, singe == "Oscar" & choice_type ==
##         "Repeat"))
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)                1.0986     0.1925   5.709 1.14e-08 ***
## InjectionDCZ              -0.2513     0.2910  -0.864 0.387724
## portesblue                 0.7732     0.2290   3.377 0.000733 ***
## InjectionDCZ:portesblue    0.3254     0.3381   0.962 0.335828
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

```
##     Null deviance: 46.102  on 37  degrees of freedom
## Residual deviance: 10.744  on 34  degrees of freedom
## AIC: 144.5
##
## Number of Fisher Scoring iterations: 4

##
## Call:
## glm(formula = trial ~ Injection * portes, family = "poisson",
##     data = subset(agg.data4B.first30, singe == "Violette" & choice_type ==
##         "Repeat"))
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               0.7472     0.2294   3.257  0.00113 **
## InjectionDCZ              0.3137     0.3018   1.039  0.29870
## portesblue                1.2677     0.2568   4.936 7.98e-07 ***
## InjectionDCZ:portesblue  -0.4265     0.3455  -1.234  0.21705
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 69.373  on 37  degrees of freedom
## Residual deviance: 24.673  on 34  degrees of freedom
## AIC: 156.46
##
## Number of Fisher Scoring iterations: 5
```

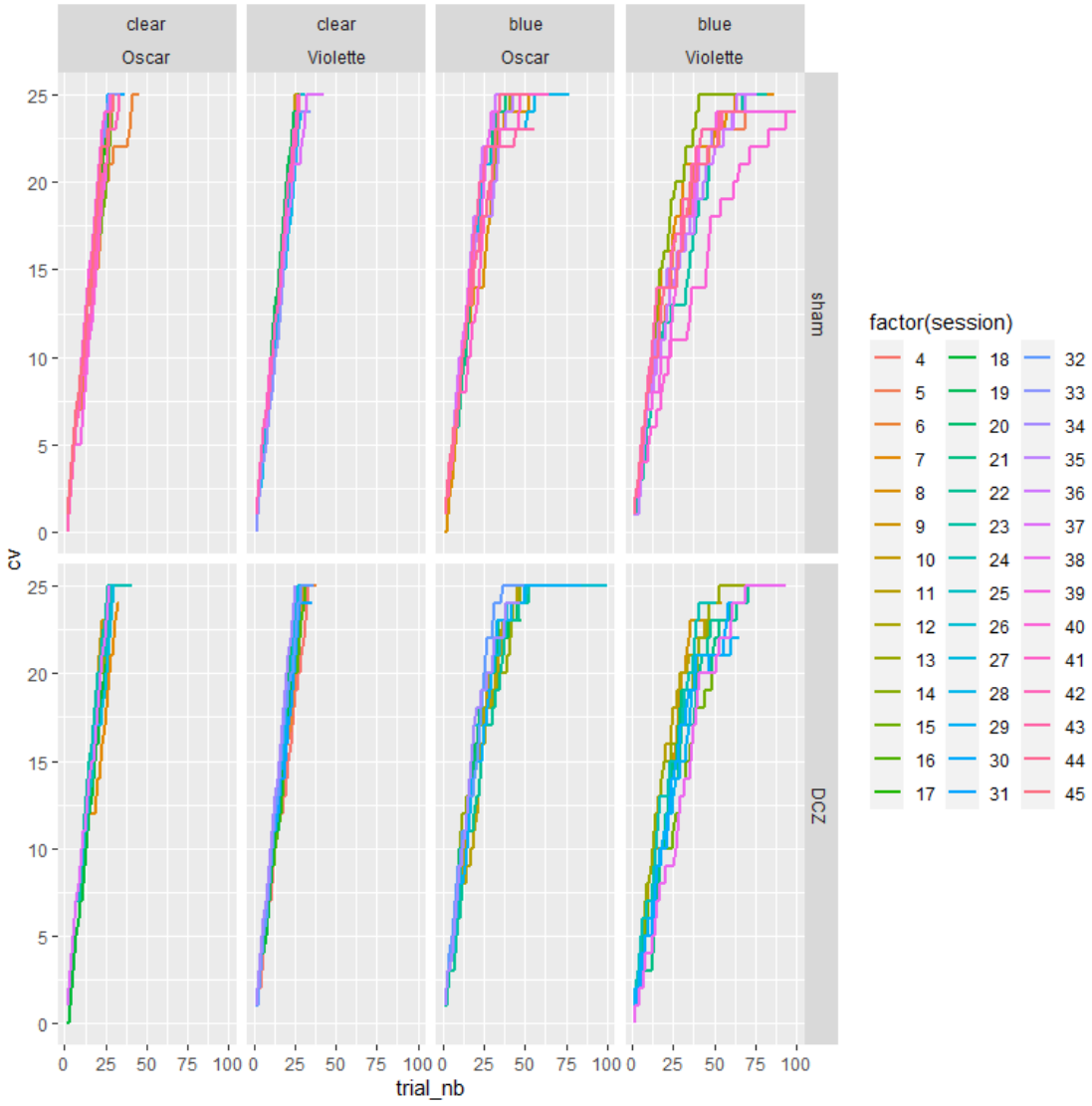


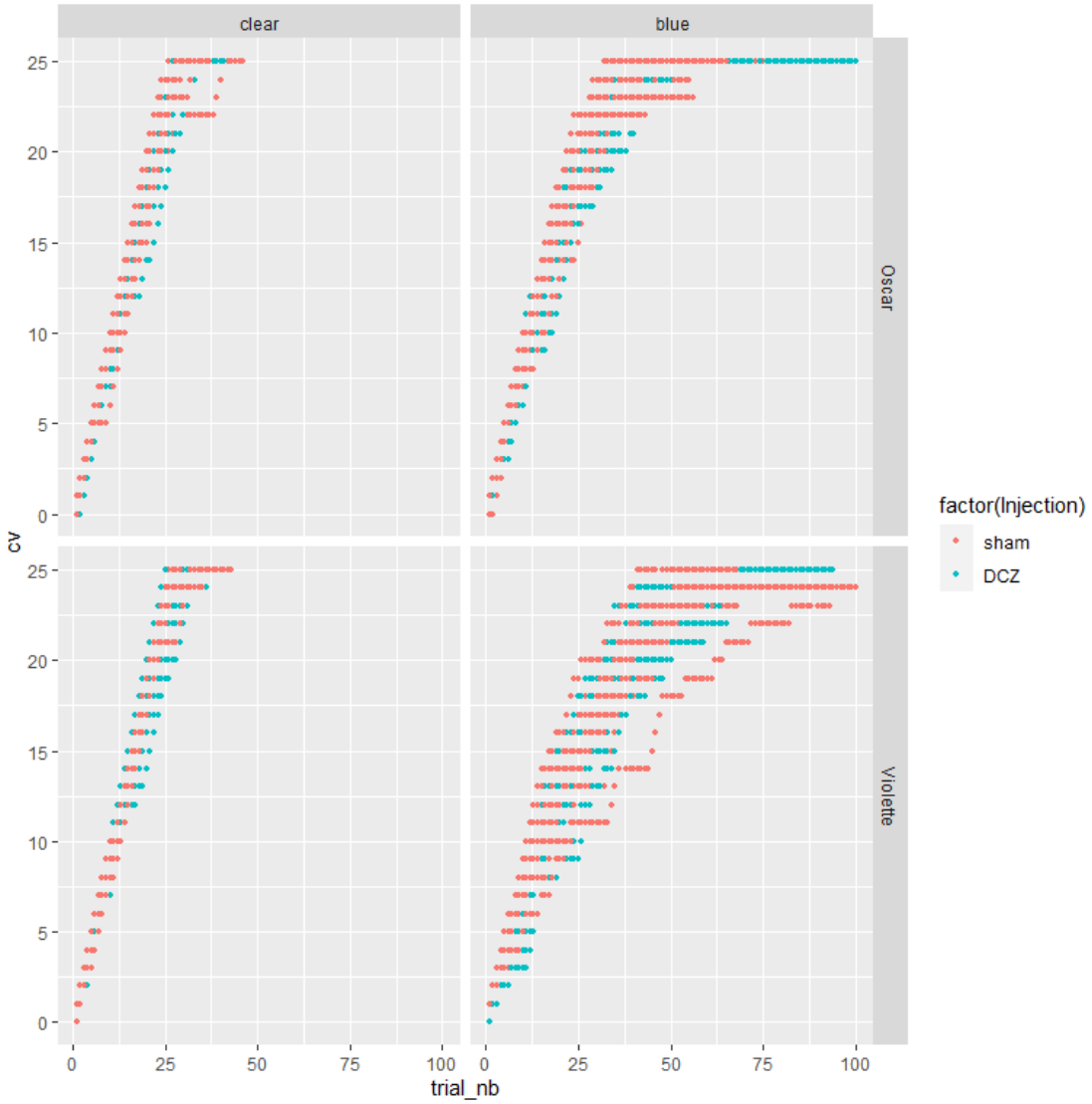

```
##
## Call:
## glm(formula = trial ~ Injection * portes, family = "poisson",
##     data = subset(agg.data4B.after30, singe == "Oscar" & choice_type ==
##         "Repeat"))
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               1.52606    0.20851   7.319 2.50e-13 ***
## InjectionDCZ              0.03209    0.31002   0.104    0.918
## portesblue                1.50890    0.21974   6.867 6.57e-12 ***
## InjectionDCZ:portesblue   0.09843    0.32425   0.304    0.761
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

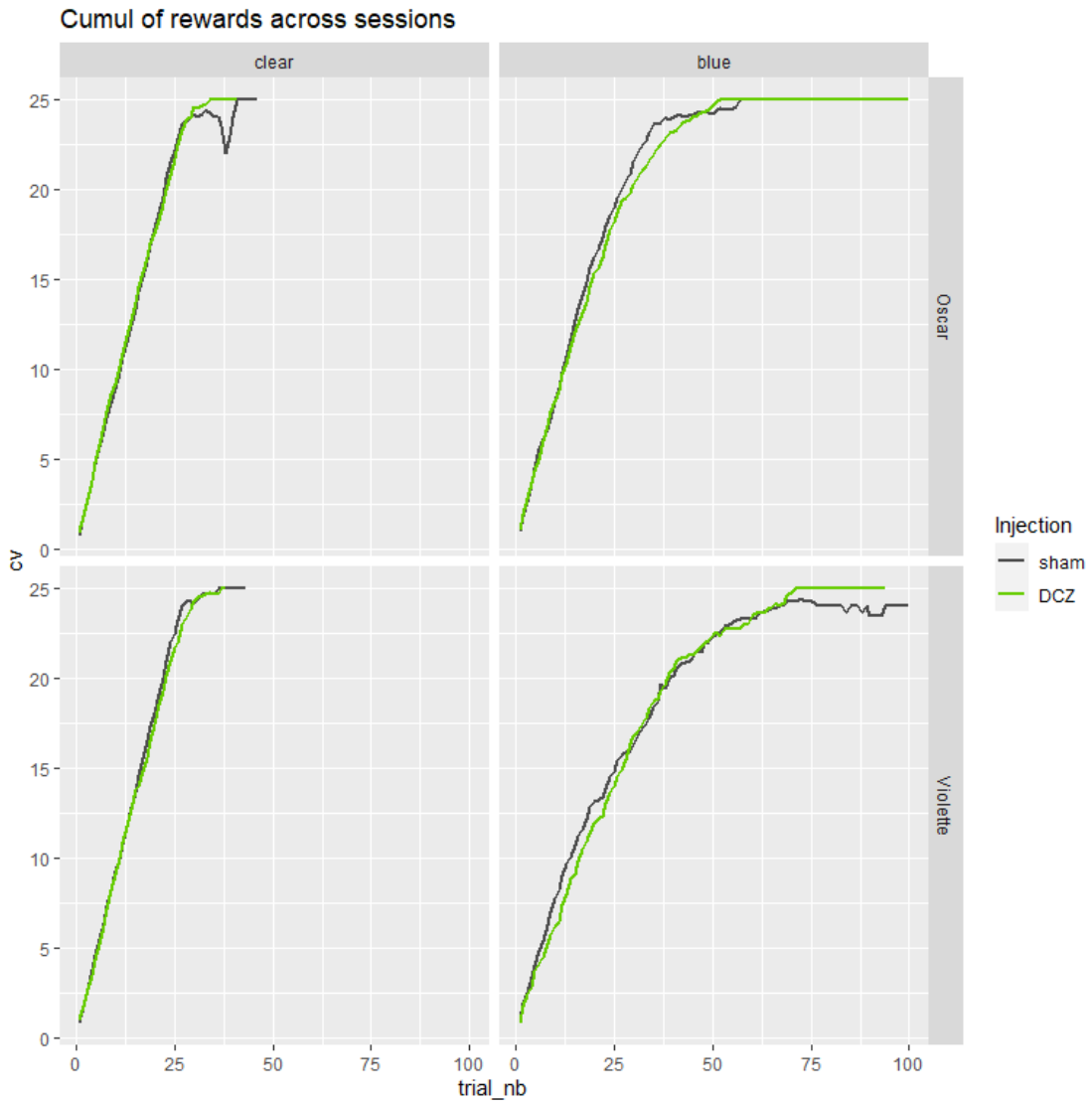
```
##     Null deviance: 301.32  on 28  degrees of freedom
## Residual deviance: 156.56  on 25  degrees of freedom
## AIC: 288.7
##
## Number of Fisher Scoring iterations: 5

##
## Call:
## glm(formula = trial ~ Injection * portes, family = "poisson",
##     data = subset(agg.data4B.after30, singe == "Violette" & choice_type ==
##         "Repeat"))
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               1.52606    0.20851   7.319  2.5e-13 ***
## InjectionDCZ             -0.13976    0.28141  -0.497    0.619
## portesblue                2.08216    0.21491   9.688  < 2e-16 ***
## InjectionDCZ:portesblue -0.04745    0.29184  -0.163    0.871
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 467.137  on 31  degrees of freedom
## Residual deviance:  95.668  on 28  degrees of freedom
## AIC: 246.59
##
## Number of Fisher Scoring iterations: 5
```

The statistics show that there is no effect of the condition (Injection) for OSCAR but there is a significant increase of repeat for VIOLET. This is when we do not take PORTES into account. If Portes is used as an interacting fixed effect the effect size for Repeats in VIOLET goes down (p=0.052).

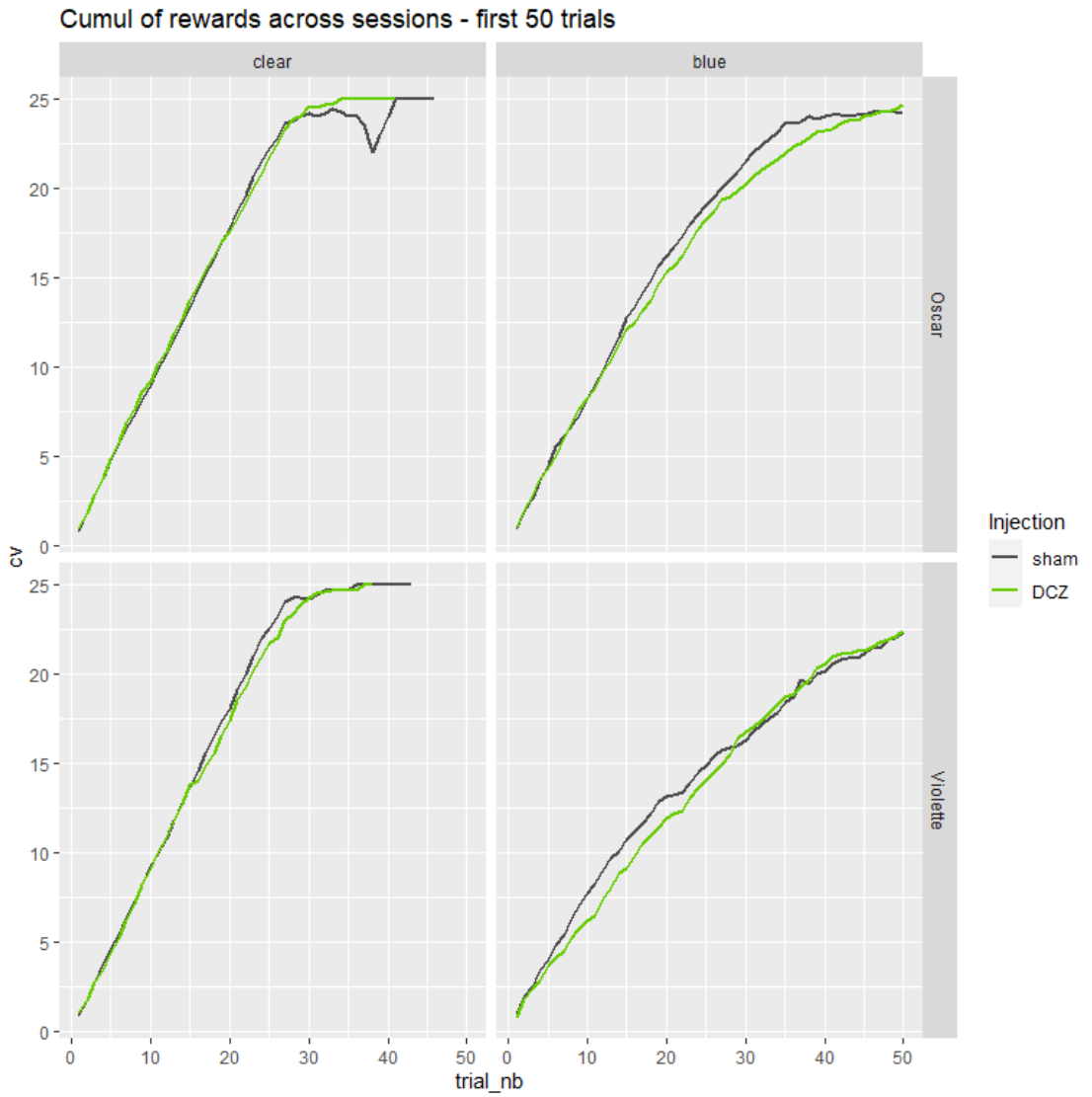
##Cumsum Reward Test whether the speed of getting rewards changes between conditions and under DCZ vs sham.


clear
Oscar
clear
Violette
blue
Oscar
blue
Violette
sham
DCZ
cv
trial_nb
0
5
10
15
20
25
0
25
50
75
100
factor(session)
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45


clear
blue
Oscar
Violette
cv
trial_nb
0
5
10
15
20
25
0
25
50
75
100
factor(Injection)
sham
DCZ




```
##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data:  cum$cv[cum$portes == "blue" & cum$singe == "Oscar" & cum$Injection
== "sham"] and cum$cv[cum$portes == "blue" & cum$singe == "Oscar" &
cum$Injection == "DCZ"]
## D = 0.23727, p-value = 0.009899
## alternative hypothesis: two-sided

##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data:  cum$cv[cum$portes == "blue" & cum$singe == "Violette" &
cum$Injection == "sham"] and cum$cv[cum$portes == "blue" & cum$singe ==
"Violette" & cum$Injection == "DCZ"]
## D = 0.2766, p-value = 0.0007964
## alternative hypothesis: two-sided
```



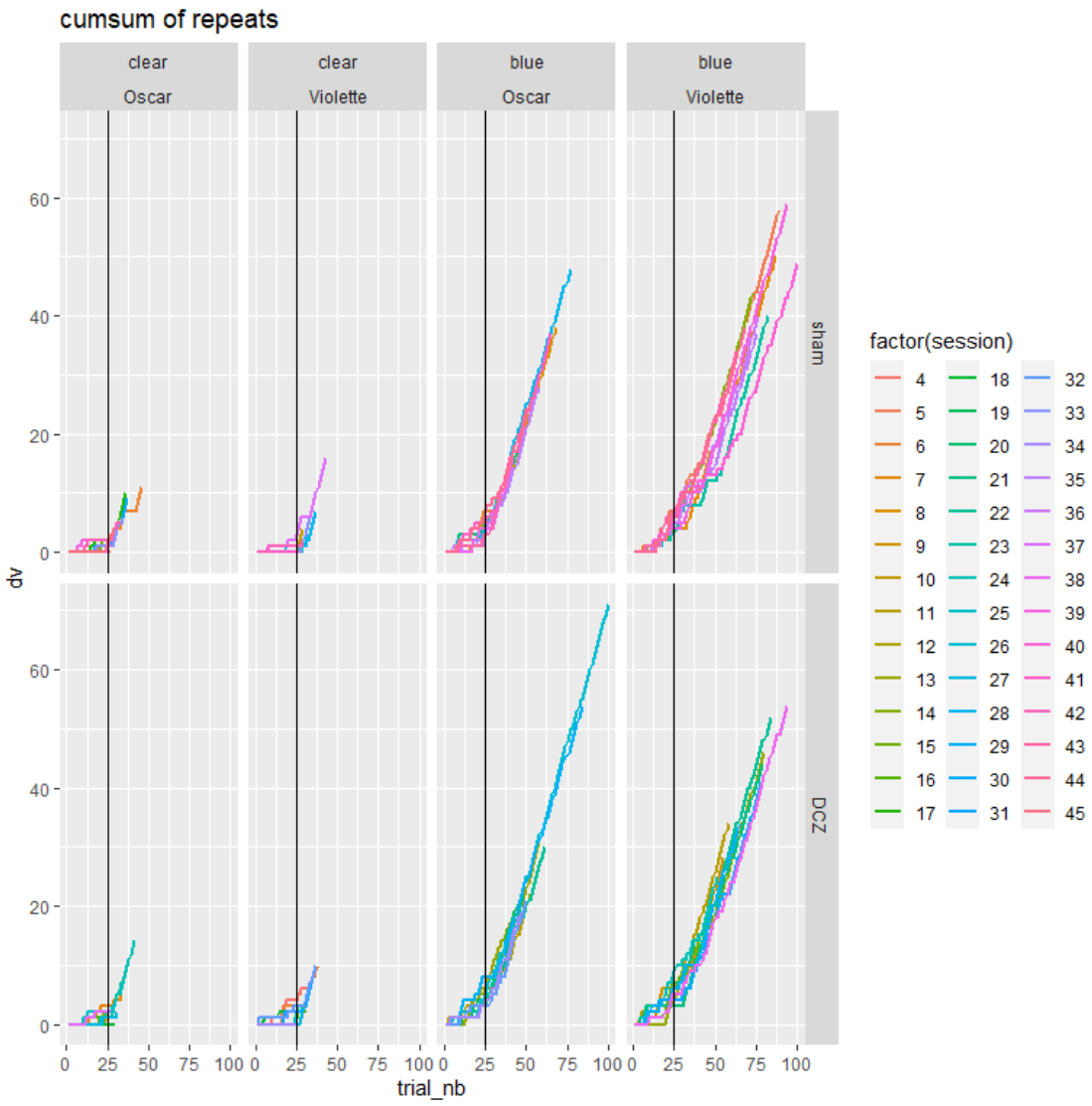

```
##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data:  cum30$cv[cum30$portes == "blue" & cum30$singe == "Oscar" &
cum30$Injection == "sham"] and cum30$cv[cum30$portes == "blue" & cum30$singe
== "Oscar" & cum30$Injection == "DCZ"]
## D = 0.14, p-value = 0.6955
## alternative hypothesis: two-sided

##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data:  cum30$cv[cum30$portes == "blue" & cum30$singe == "Violette" &
cum30$Injection == "sham"] and cum30$cv[cum30$portes == "blue" & cum30$singe
== "Violette" & cum30$Injection == "DCZ"]
## D = 0.08, p-value = 0.9969
## alternative hypothesis: two-sided
```

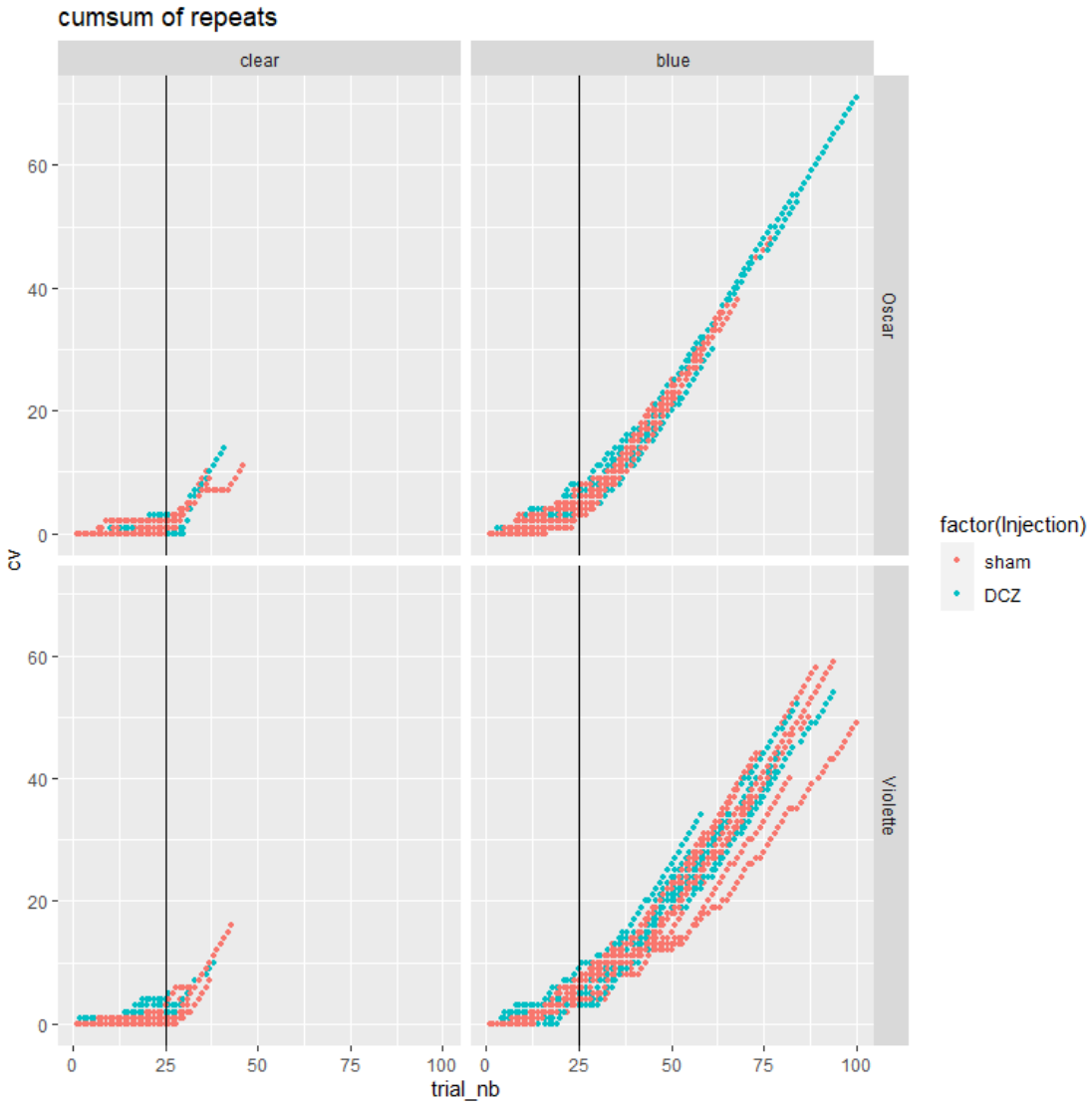
There is an impact of DCZ on the speed at which animals get rewards under DCZ compared to Sham, in the opaque condition, not in the clear. Specifically it seems that animals accumulate rewards faster under DCZ.IT is not significant if one takes only the first 50 trials for the 2 animals

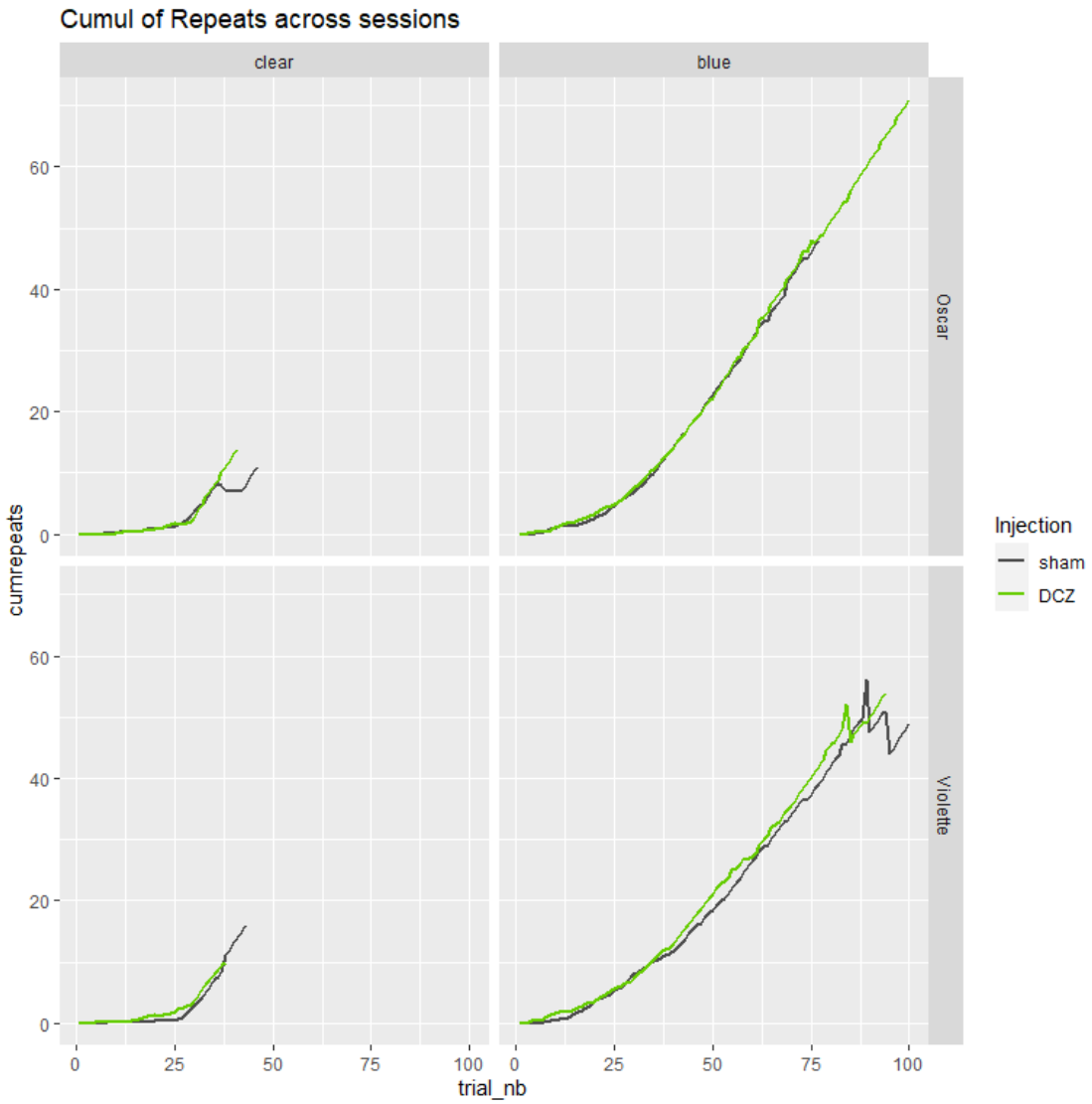
One possibility is that it's because animals make less repeats at the begining thus cumulating rewards faster.

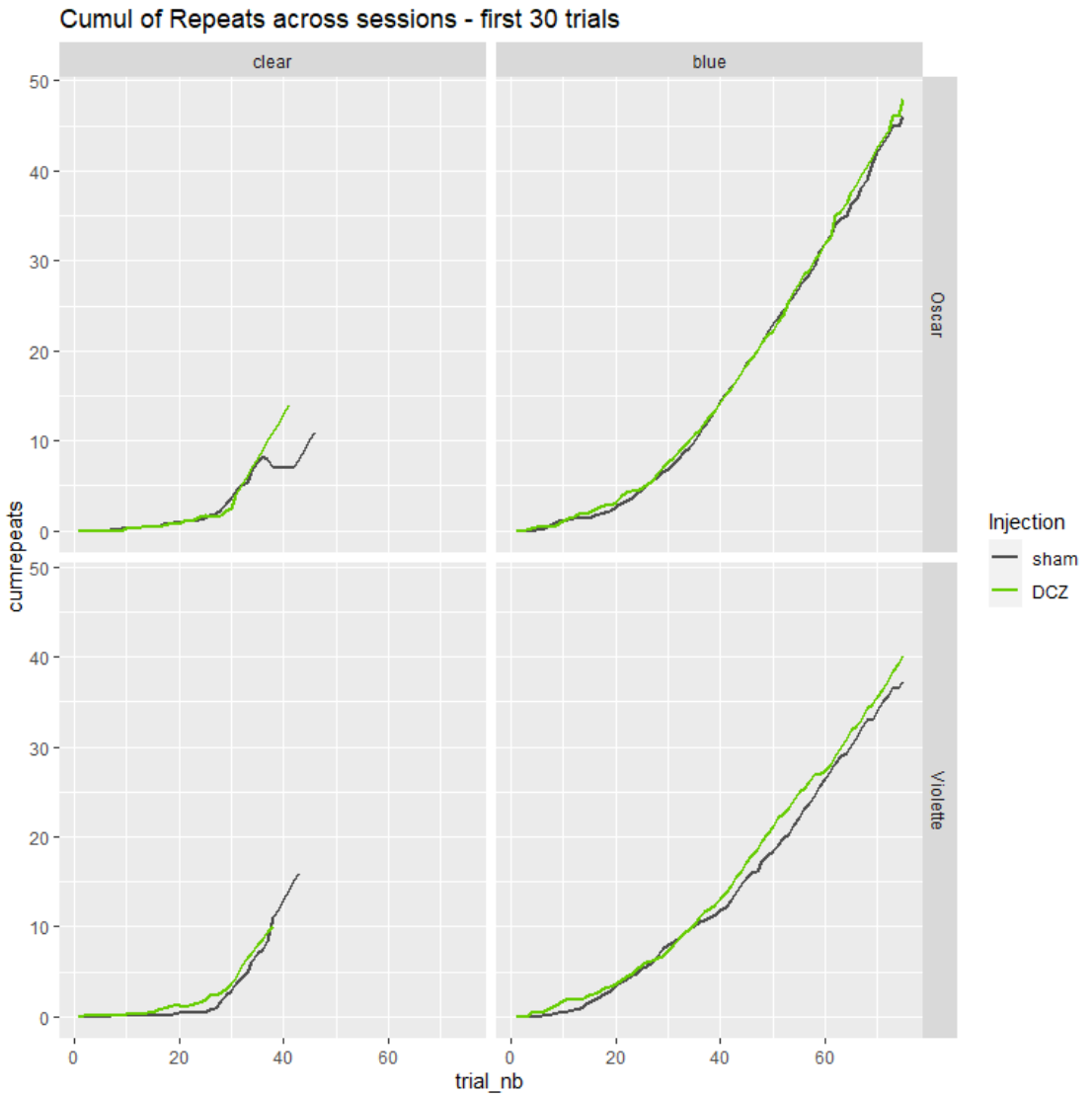
##Cumsum repeats

Test whether the repeats appear later using a cumsum curve


cumsum of repeats
clear
Oscar
clear
Violette
blue
Oscar
blue
Violette
sham
DCZ
dv
trial_nb
0
20
40
60
0
25
50
75
100
factor(session)
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45


cumsum of repeats
clear
blue
Oscar
Violette
cv
trial_nb
0
20
40
60
0
25
50
75
100
factor(Injection)
sham
DCZ


Cumul of Repeats across sessions
clear
blue
Oscar
Violette
cumrepeats
trial_nb
0
20
40
60
0
25
50
75
100
Injection
sham
DCZ

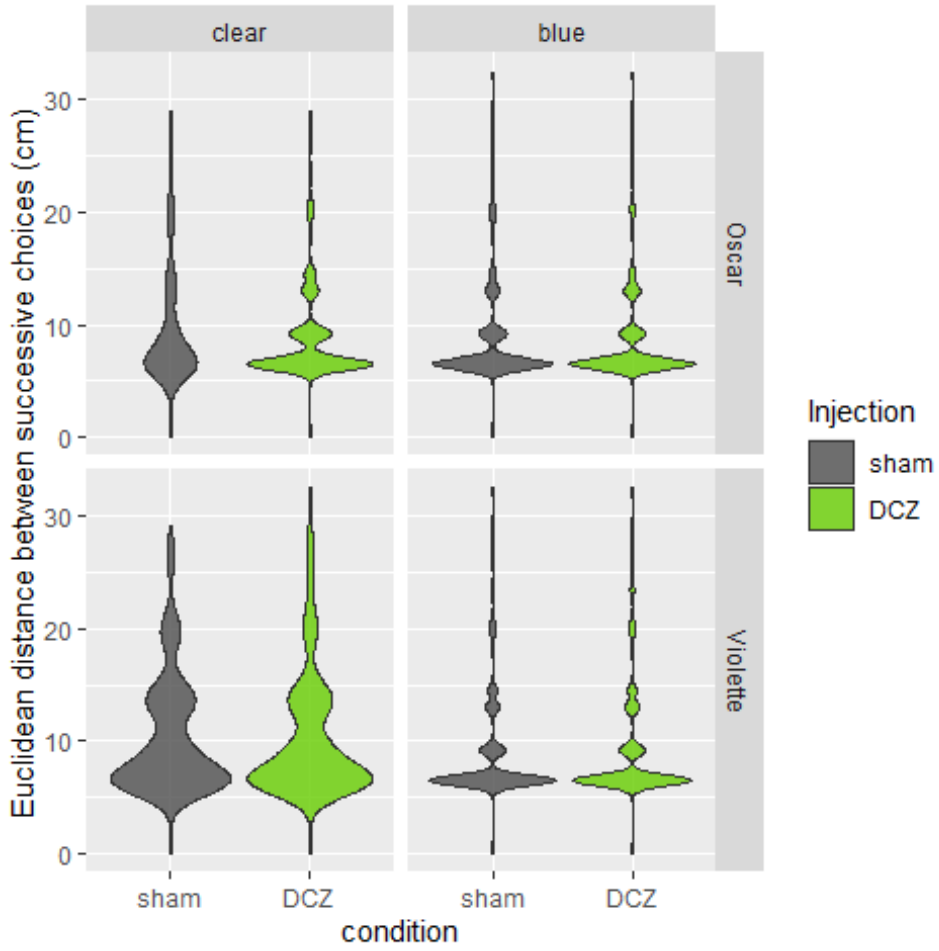



```
##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data:  cumdr30$cumrepeats[cumdr30$portes == "blue" & cumdr30$singe ==
"Oscar" & cumdr30$Injection == "sham"] and cumdr30$cumrepeats[cumdr30$portes
== "blue" & cumdr30$singe == "Oscar" & cumdr30$Injection == "DCZ"]
## D = 0.053333, p-value = 0.9999
## alternative hypothesis: two-sided

##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data:  cumdr30$cumrepeats[cumdr30$portes == "blue" & cumdr30$singe ==
"Violette" & cumdr30$Injection == "sham"] and
cumdr30$cumrepeats[cumdr30$portes == "blue" & cumdr30$singe == "Violette" &
cumdr30$Injection == "DCZ"]
```

```
## D = 0.08, p-value = 0.9683
## alternative hypothesis: two-sided
```

No different.

##stats spatial strategy



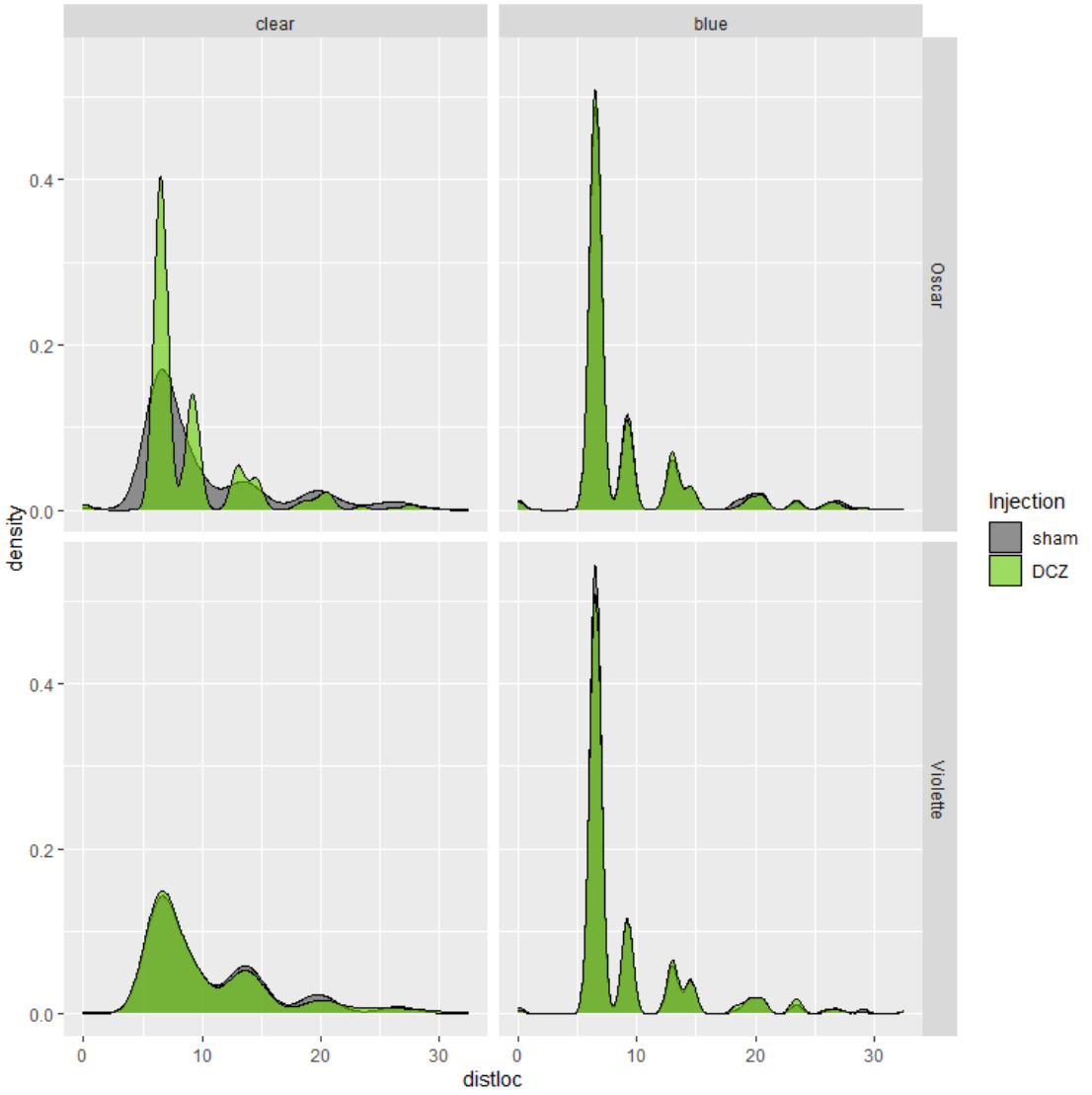

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  data4B$distloc[data4B$singe == "Oscar" & data4B$portes == "clear" &
data4B$Injection == "DCZ"] and data4B$distloc[data4B$singe == "Oscar" &
data4B$portes == "clear" & data4B$Injection == "sham"]
## D = 0.072792, p-value = 0.364
## alternative hypothesis: two-sided

##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  data4B$distloc[data4B$singe == "Oscar" & data4B$portes == "blue" &
data4B$Injection == "DCZ"] and data4B$distloc[data4B$singe == "Oscar" &
data4B$portes == "blue" & data4B$Injection == "sham"]
```

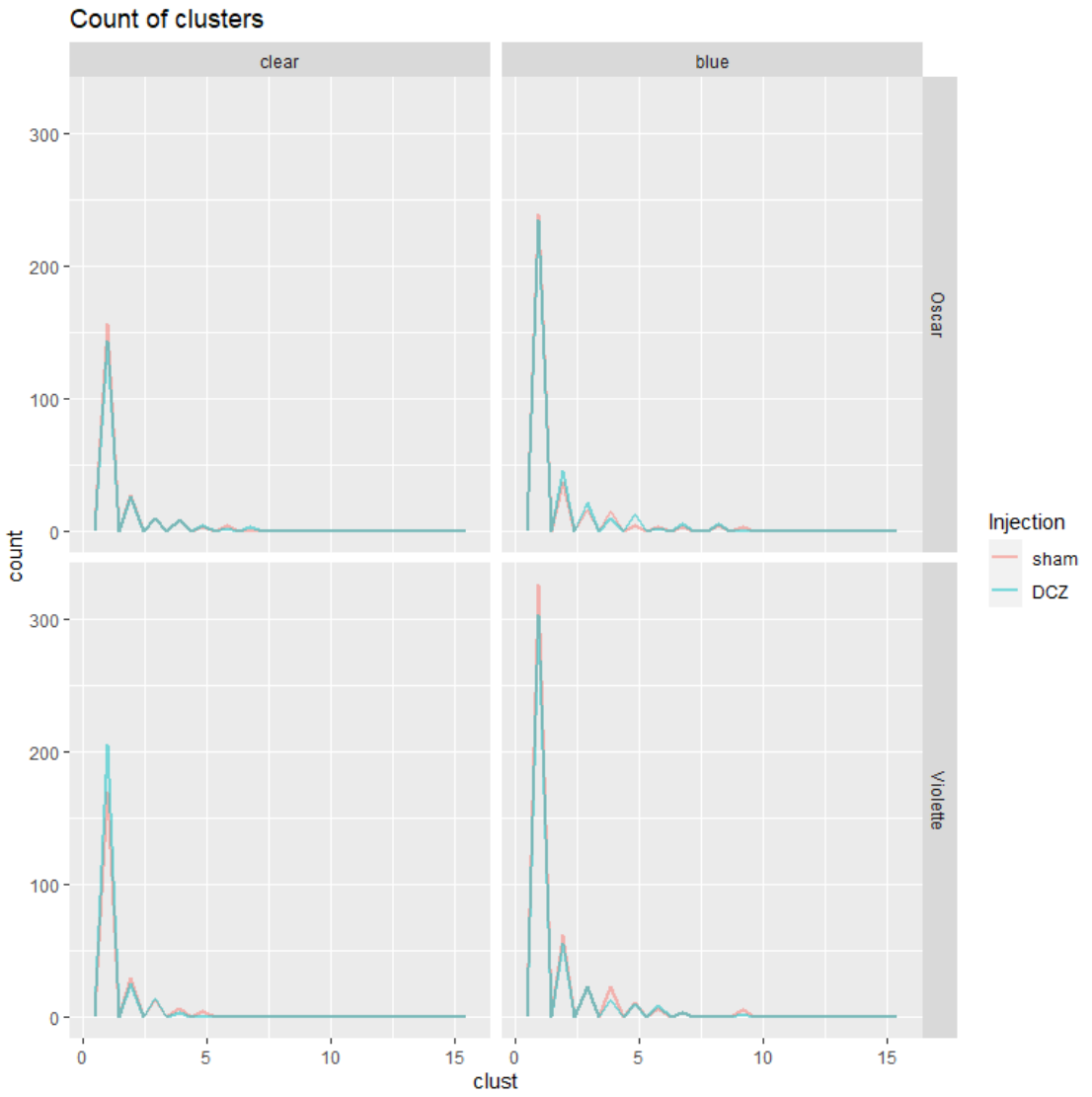
```
## D = 0.016164, p-value = 1
## alternative hypothesis: two-sided

##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  data4B$distloc[data4B$singe == "Violette" & data4B$portes ==
"clear" & data4B$Injection == "DCZ"] and data4B$distloc[data4B$singe ==
"Violette" & data4B$portes == "clear" & data4B$Injection == "sham"]
## D = 0.026996, p-value = 0.9998
## alternative hypothesis: two-sided

##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  data4B$distloc[data4B$singe == "Violette" & data4B$portes == "blue"
& data4B$Injection == "DCZ"] and data4B$distloc[data4B$singe == "Violette" &
data4B$portes == "blue" & data4B$Injection == "sham"]
## D = 0.025571, p-value = 0.9694
## alternative hypothesis: two-sided
```

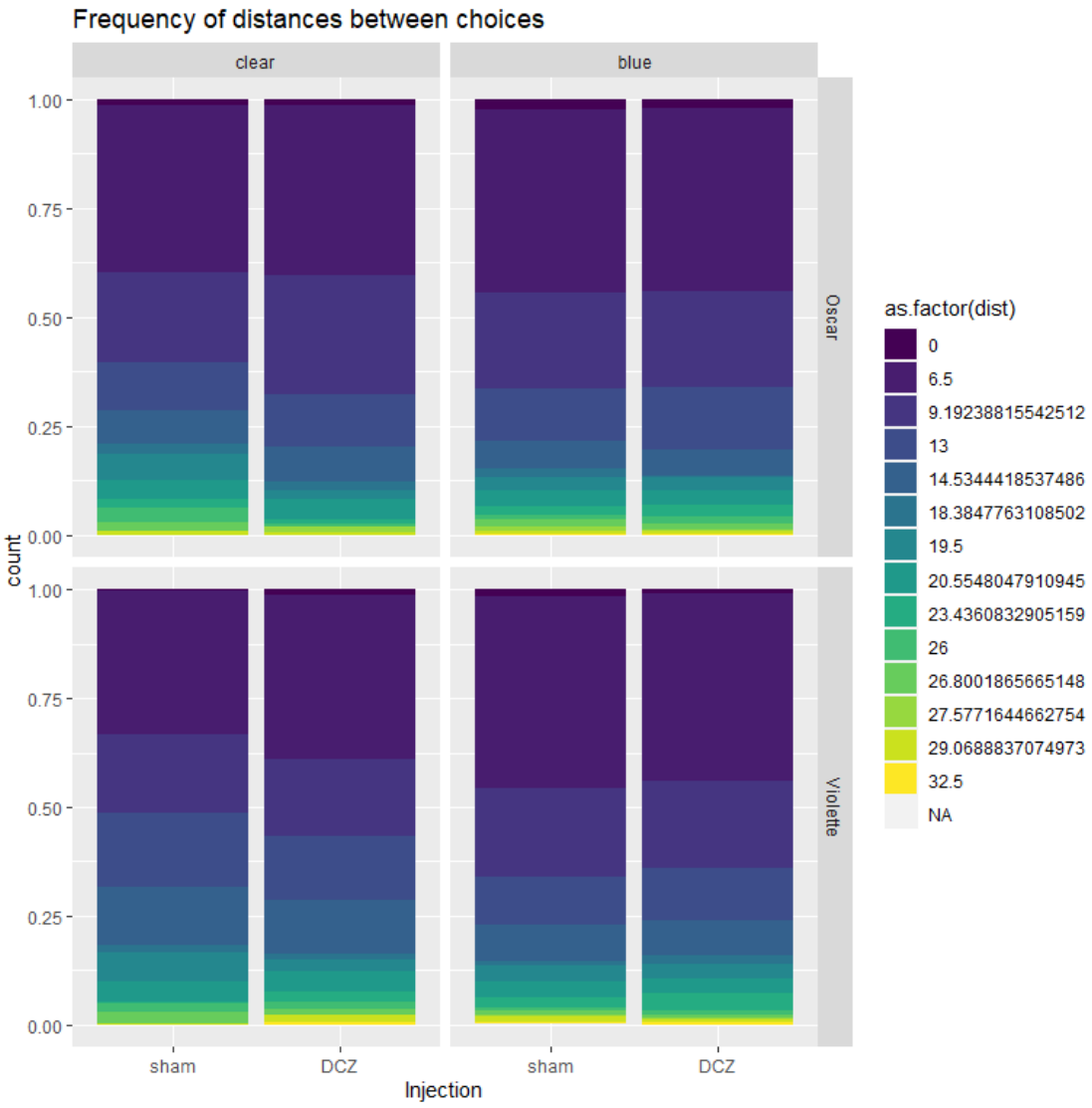
There is *no* difference in distribution of distances between sham and DCZ.

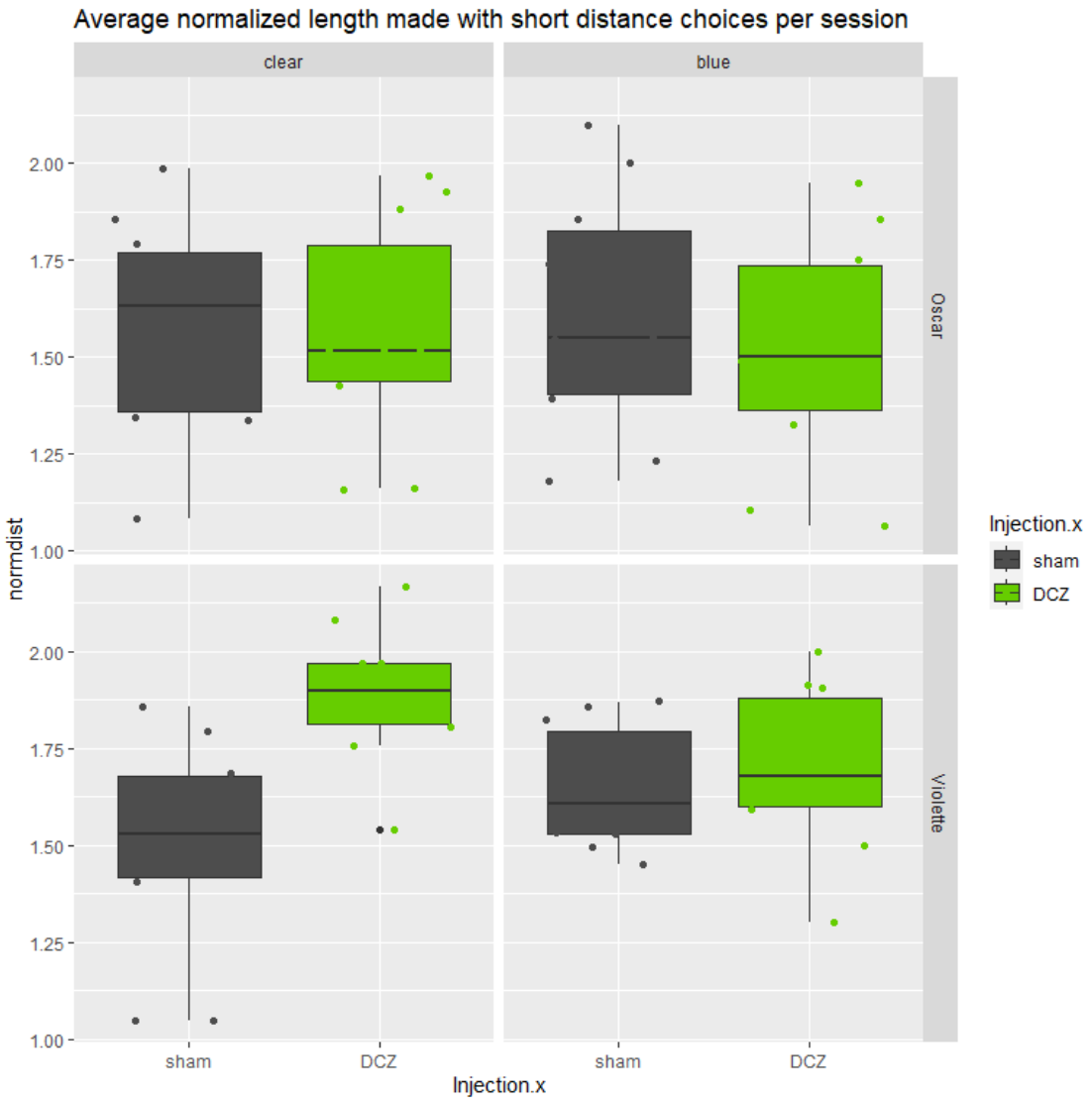
##Clusters of small distances

We look at clustering in the sense of succession of choices at small distances.


clear
blue
Oscar
Violette
density
0.0
0.2
0.4
0
10
20
30
distloc
Injection
sham
DCZ

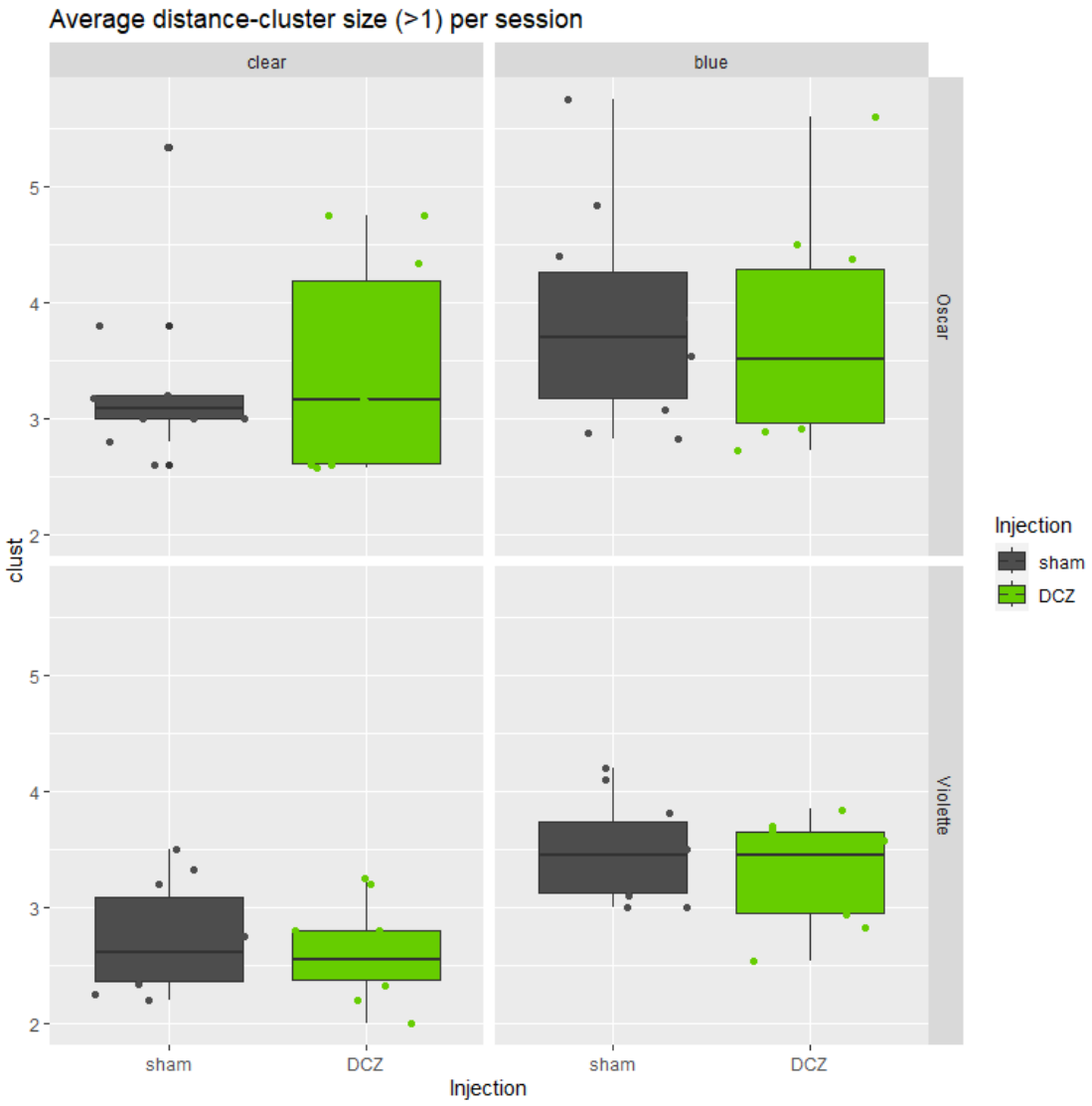

Count of clusters
clear
blue
Oscar
Violette
count
clust
0
100
200
300
0
5
10
15
Injection
sham
DCZ


Frequency of distances between choices
clear
blue
Oscar
Violette
count
1.00
0.75
0.50
0.25
0.00
sham
DCZ
Injection
as.factor(dist)
0
6.5
9.19238815542512
13
14.5344418537486
18.3847763108502
19.5
20.5548047910945
23.4360832905159
26
26.8001865665148
27.5771644662754
29.0688837074973
32.5
NA



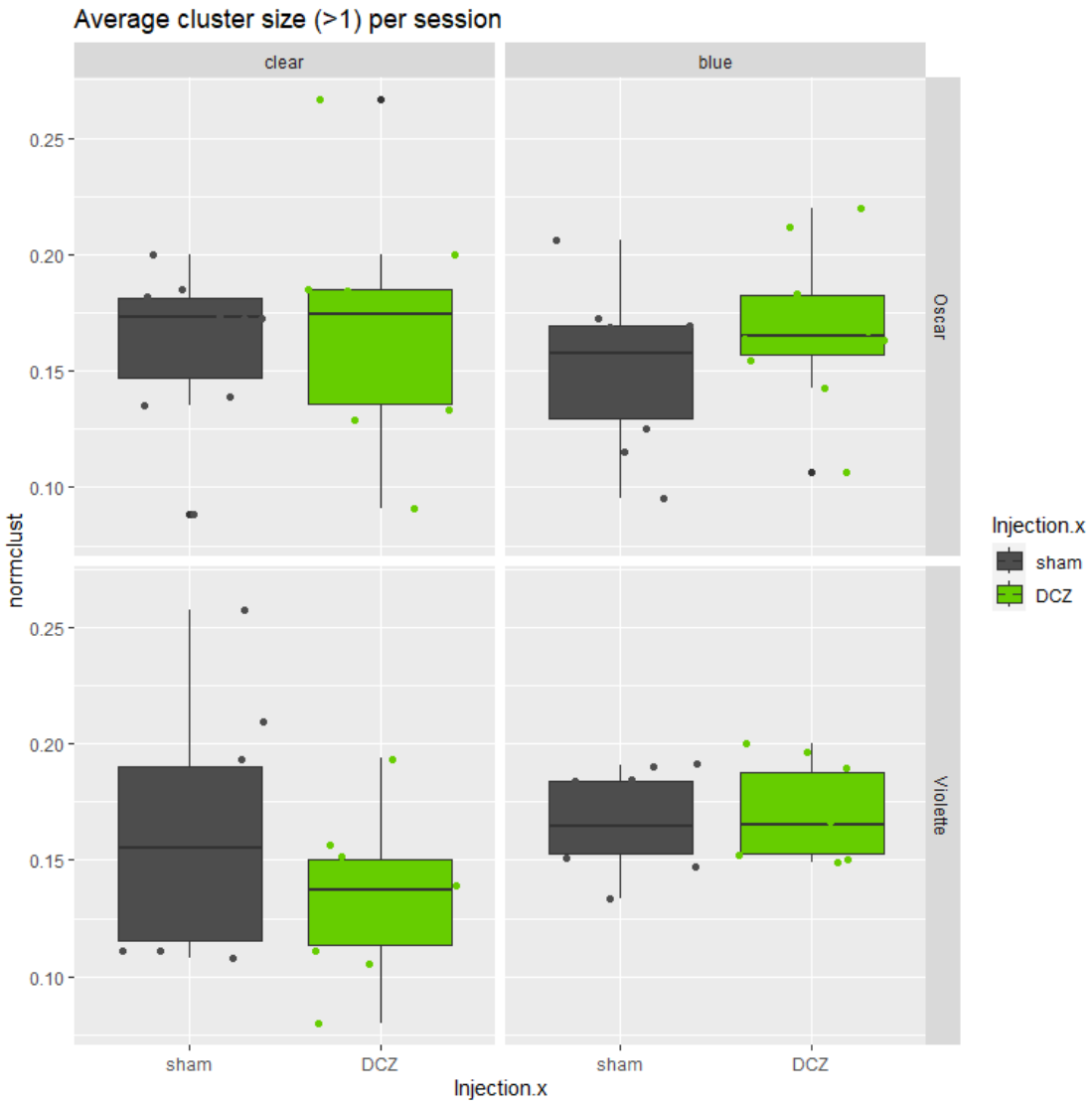

```
##
## Call:
## lm(formula = normdist ~ portes.x * Injection.x, data = subset(newdata2,
##     singe.x == "Violette"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45157 -0.10870 -0.00319  0.17719  0.35719
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    1.49995    0.06686  22.433   <2e-16 ***
## portes.xblue                   0.14871    0.09456   1.573   0.1246
## Injection.xDCZ                 0.39153    0.09456   4.141   0.0002 ***
## portes.xblue:Injection.xDCZ -0.34174    0.13373  -2.555   0.0150 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.2114 on 36 degrees of freedom
## Multiple R-squared:  0.3275, Adjusted R-squared:  0.2715
## F-statistic: 5.844 on 3 and 36 DF,  p-value: 0.002328

##
## Call:
## lm(formula = normdist ~ portes.x * Injection.x, data = subset(newdata2,
##     singe.x == "Oscar"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49314 -0.19958 -0.03765  0.23409  0.49586
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.57648    0.09336  16.886   <2e-16 ***
## portes.xblue                  0.02767    0.13203   0.210    0.835
## Injection.xDCZ               -0.02216    0.13203  -0.168    0.868
## portes.xblue:Injection.xDCZ -0.05978    0.18672  -0.320    0.751
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2952 on 36 degrees of freedom
## Multiple R-squared:  0.01136,    Adjusted R-squared:  -0.07102
## F-statistic: 0.1379 on 3 and 36 DF,  p-value: 0.9367
```




```
##
## Call:
## glm(formula = clust ~ portes * Injection, family = "poisson",
##     data = subset(distClustyClust.session, singe == "Violette"))
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                1.00186    0.19162   5.228 1.71e-07 ***
## portesblue                 0.24704    0.25574   0.966    0.334
## InjectionDCZ              -0.03932    0.27370  -0.144    0.886
## portesblue:InjectionDCZ -0.01581    0.36593  -0.043    0.966
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 4.0913  on 39  degrees of freedom
```

```
## Residual deviance: 2.2949  on 36  degrees of freedom
## AIC: Inf
##
## Number of Fisher Scoring iterations: 4

##
## Call:
## glm(formula = clust ~ portes * Injection, family = "poisson",
##     data = subset(distClustyClust.session, singe == "Oscar"))
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               1.19594    0.17390   6.877 6.11e-12 ***
## portesblue                0.15215    0.23710   0.642    0.521
## InjectionDCZ              0.03631    0.24373   0.149    0.882
## portesblue:InjectionDCZ -0.07283    0.33514  -0.217    0.828
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 7.9931  on 39  degrees of freedom
## Residual deviance: 7.4677  on 36  degrees of freedom
## AIC: Inf
##
## Number of Fisher Scoring iterations: 4
```




```
##
## Call:
## lm(formula = normclust ~ portes.x * Injection.x, data = subset(newdata2,
##     singe.x == "Violette"))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.053998 -0.020574 -0.001252  0.018145  0.096316
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.160827   0.010352  15.536   <2e-16 ***
## portes.xblue                 0.005941   0.014640   0.406   0.6873
## Injection.xDCZ              -0.026829   0.014640  -1.833   0.0751 .
## portes.xblue:Injection.xDCZ  0.030368   0.020703   1.467   0.1511
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.03273 on 36 degrees of freedom
## Multiple R-squared:  0.174,  Adjusted R-squared:  0.1051
## F-statistic: 2.527 on 3 and 36 DF,  p-value: 0.07275

##
## Call:
## lm(formula = normclust ~ portes.x * Injection.x, data = subset(newdata2,
##     singe.x == "Oscar"))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.077241 -0.025476  0.009756  0.017903  0.098517
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.162657   0.011718  13.881 5.05e-16 ***
## portes.xblue                -0.011630   0.016572  -0.702    0.487
## Injection.xDCZ               0.005492   0.016572   0.331    0.742
## portes.xblue:Injection.xDCZ  0.012754   0.023436   0.544    0.590
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03706 on 36 degrees of freedom
## Multiple R-squared:  0.04059,    Adjusted R-squared:  -0.03936
## F-statistic: 0.5077 on 3 and 36 DF,  p-value: 0.6795
```

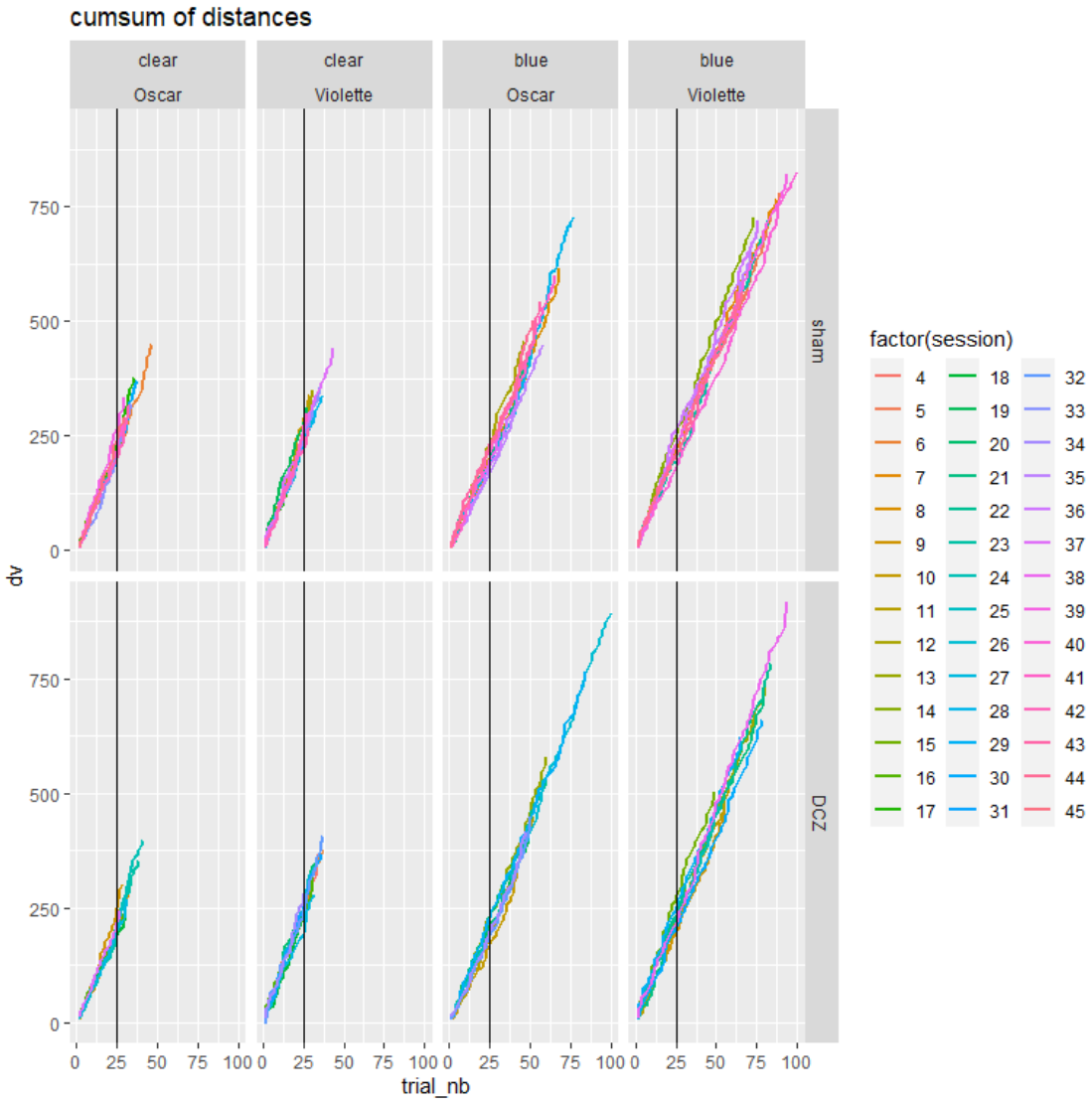
After counting and normalizing we don't find any difference in the clustering of distances between choices: no difference in the number of choices Hence the strategy of the animal, in terms of clustering choices with small distances, don't seem to differ.

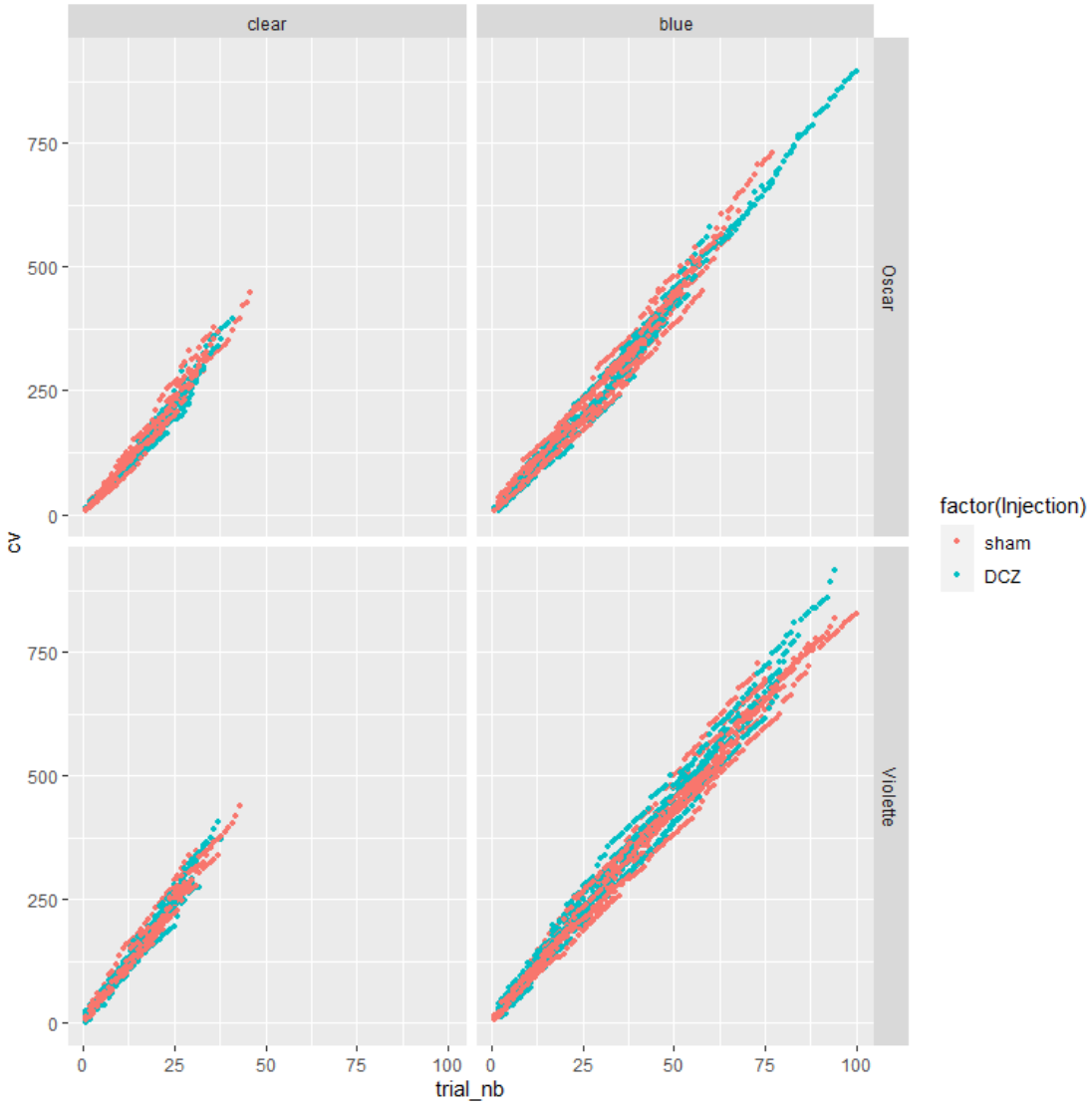
Note that surprisingly only VIOLET shows the effect in which the number of cluster with short distances is larger in Blue than in Clear conditions. OSCAR does not seem to change his strategy in that regard between the 2…

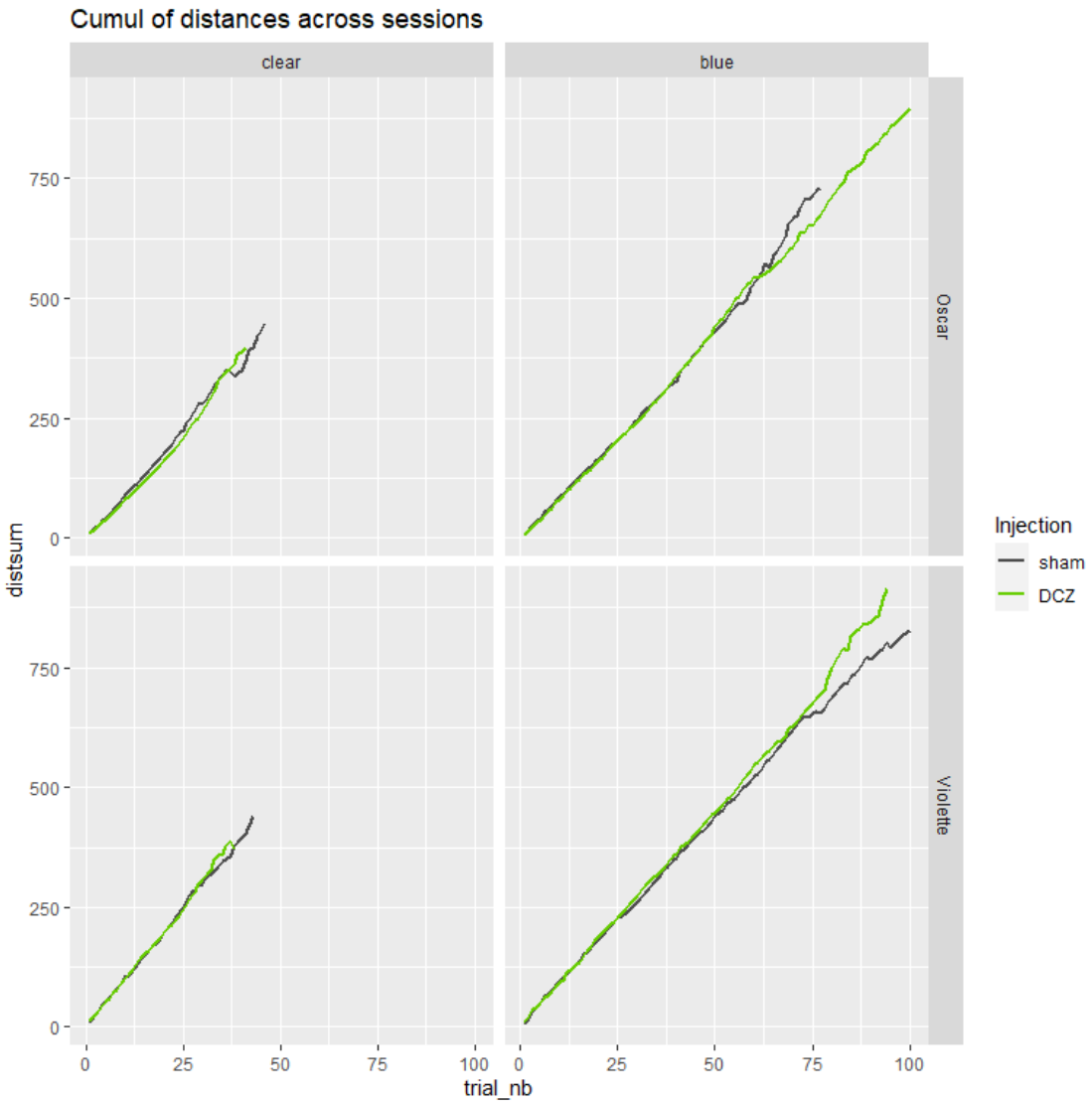
However, this is surprising because analyses and figures above suggest that under rDCZ in opaque animals get rewards faster and that it could be accompanied by more short distances between 2 successive choices. !! NEED to investigate further!!

##Cumsum distance - trajectory

Test whether the speed of getting rewards changes between conditions and under DCZ vs sham.


cumsum of distances
clear
Oscar
clear
Violette
blue
Oscar
blue
Violette
sham
DCZ
dv
trial_nb
0
250
500
750
0
25
50
75
100
factor(session)
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45


clear
blue
Oscar
Violette
cv
trial_nb
0
250
500
750
0
25
50
75
100
factor(Injection)
sham
DCZ

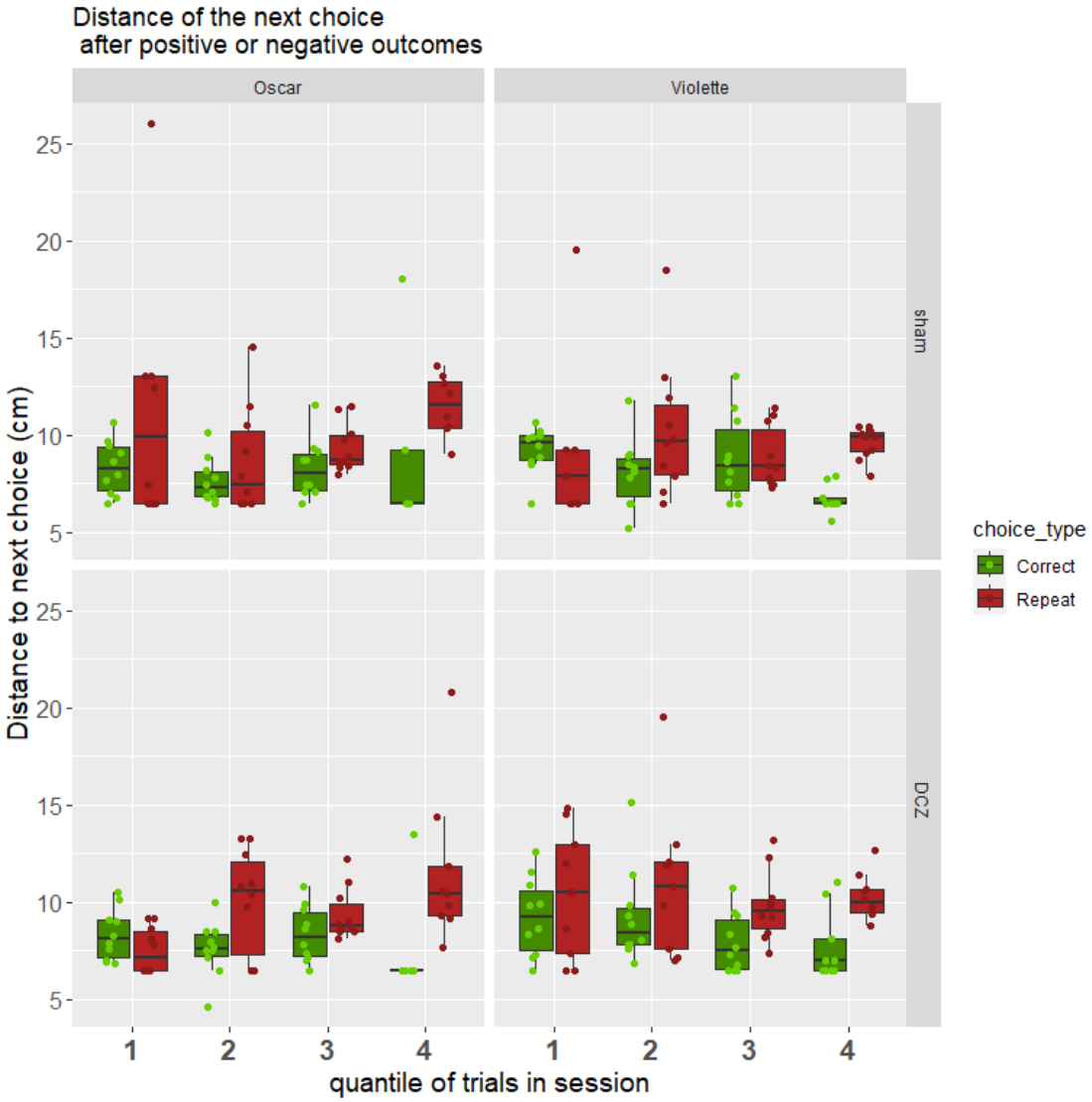


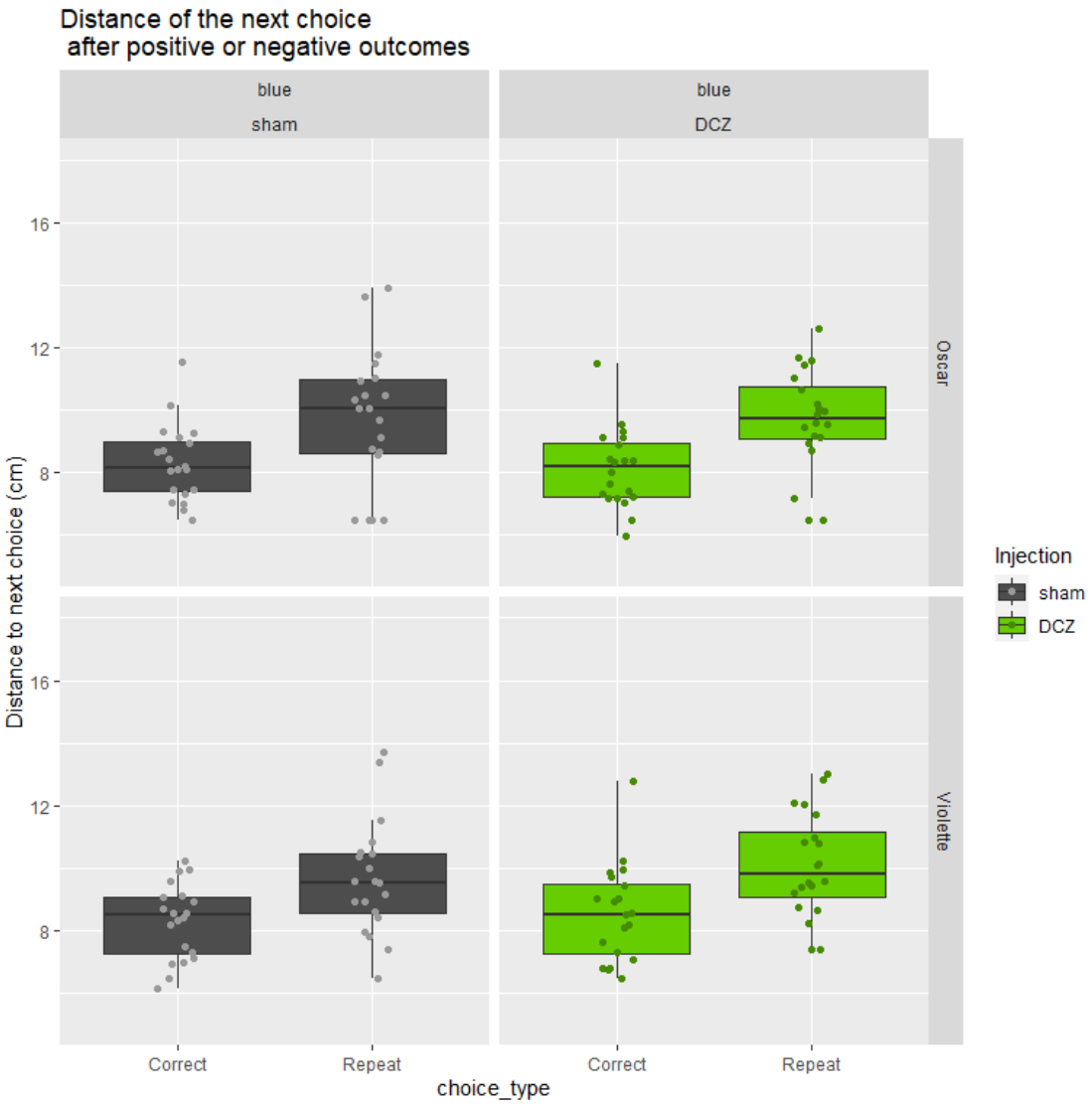

```
##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data:  cumd$distsum[cumd$portes == "blue" & cumd$singe == "Oscar" &
cumd$Injection == "sham"] and cumd$distsum[cumd$portes == "blue" & cumd$singe
== "Oscar" & cumd$Injection == "DCZ"]
## D = 0.20325, p-value = 0.04617
## alternative hypothesis: two-sided

##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data:  cumd$distsum[cum$portes == "blue" & cumd$singe == "Violette" &
cumd$Injection == "sham"] and cumd$distsum[cumd$portes == "blue" & cumd$singe
== "Violette" & cumd$Injection == "DCZ"]
## D = 0.086383, p-value = 0.8205
## alternative hypothesis: two-sided
```

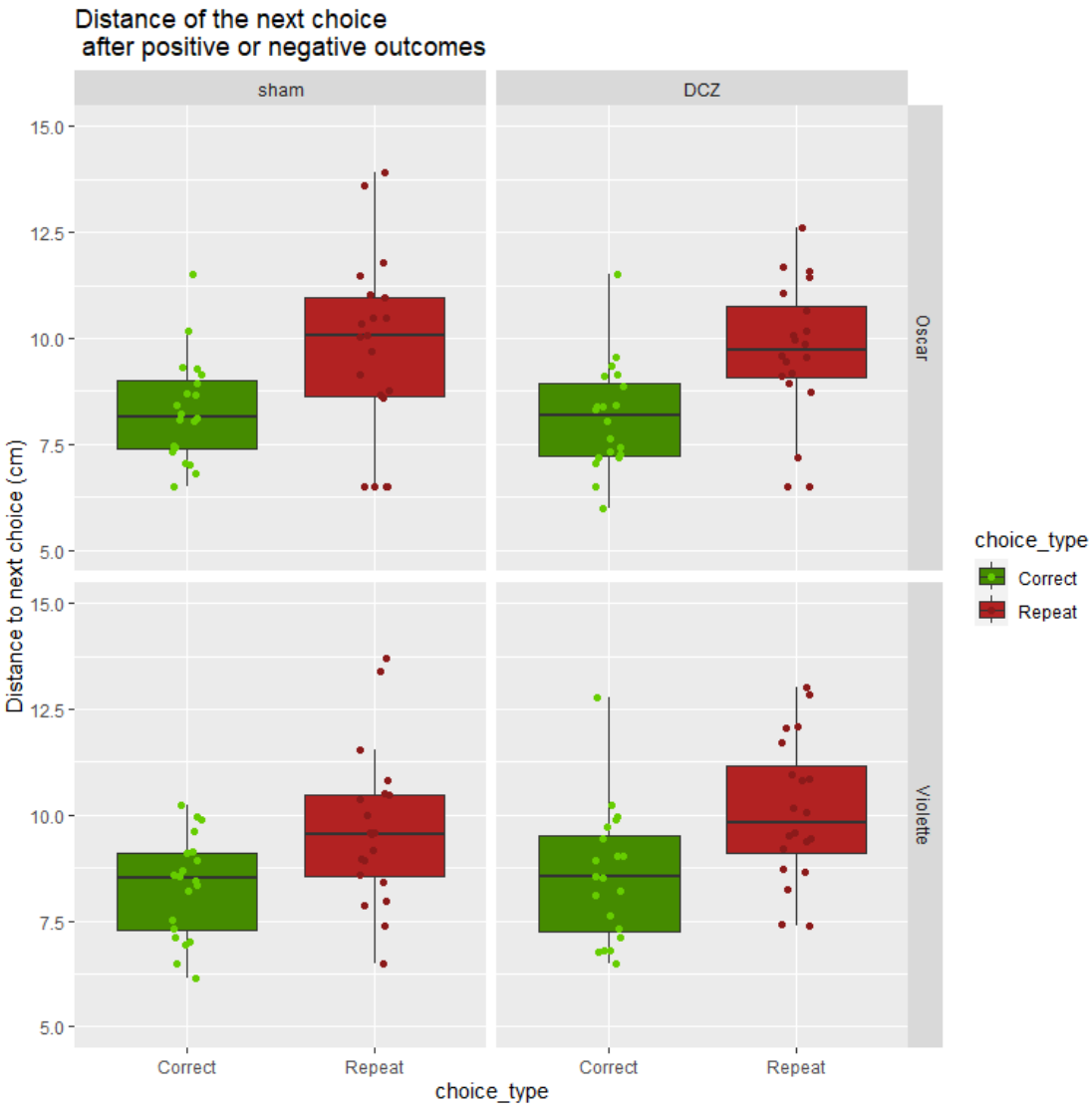
##Post-error reaction vs Reward

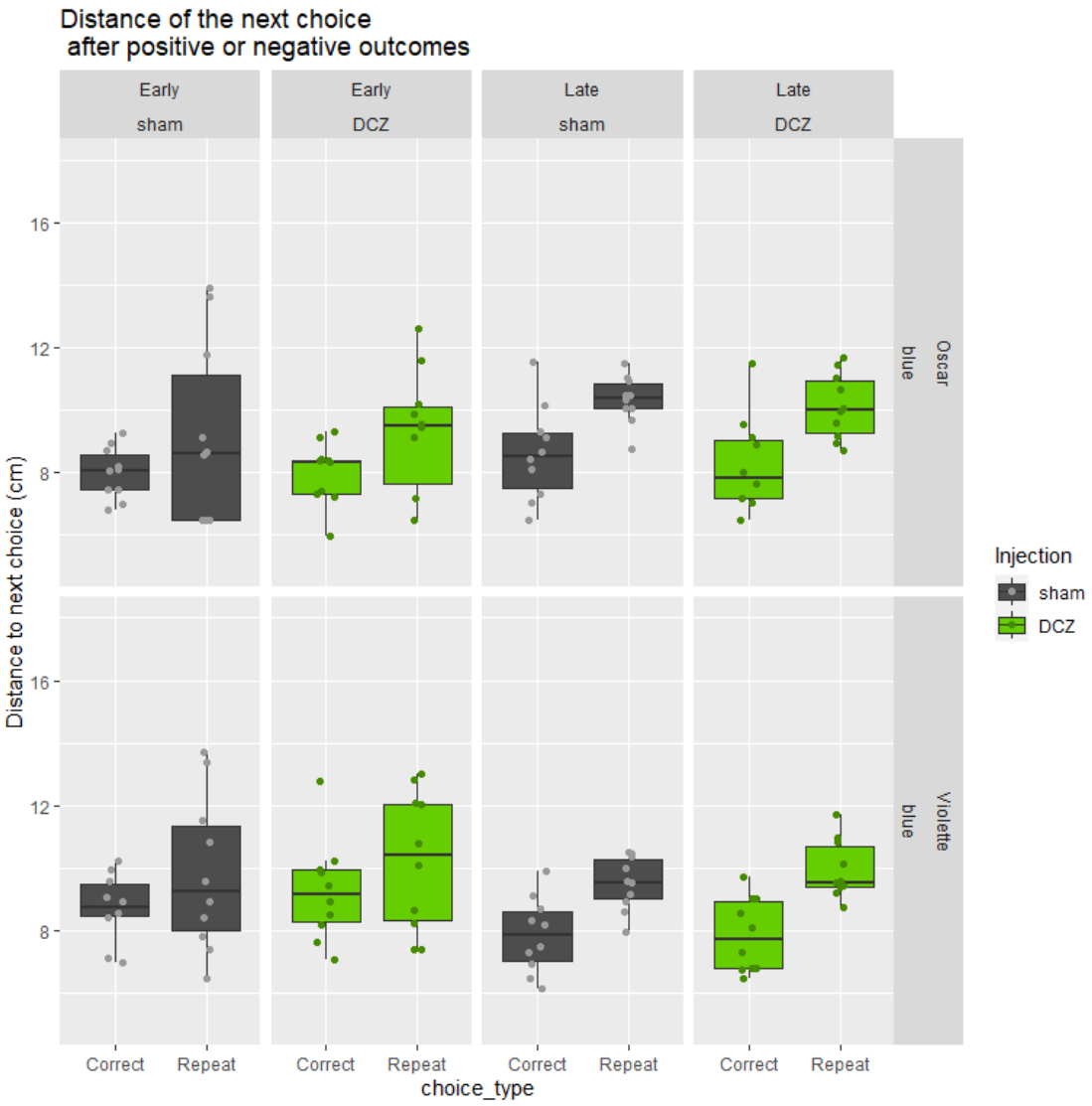
Test whether the choices made after no rewards (e.g. next distance of choice: close or far). We hypothesize that in the blue condition (the animal doesn't see rewards) the distance after negative outcomes (for a repeat) is larger than after a correct rewarded response, because when rewarded the animal will stay 'in the patch' i.e. close to where he got the reward.

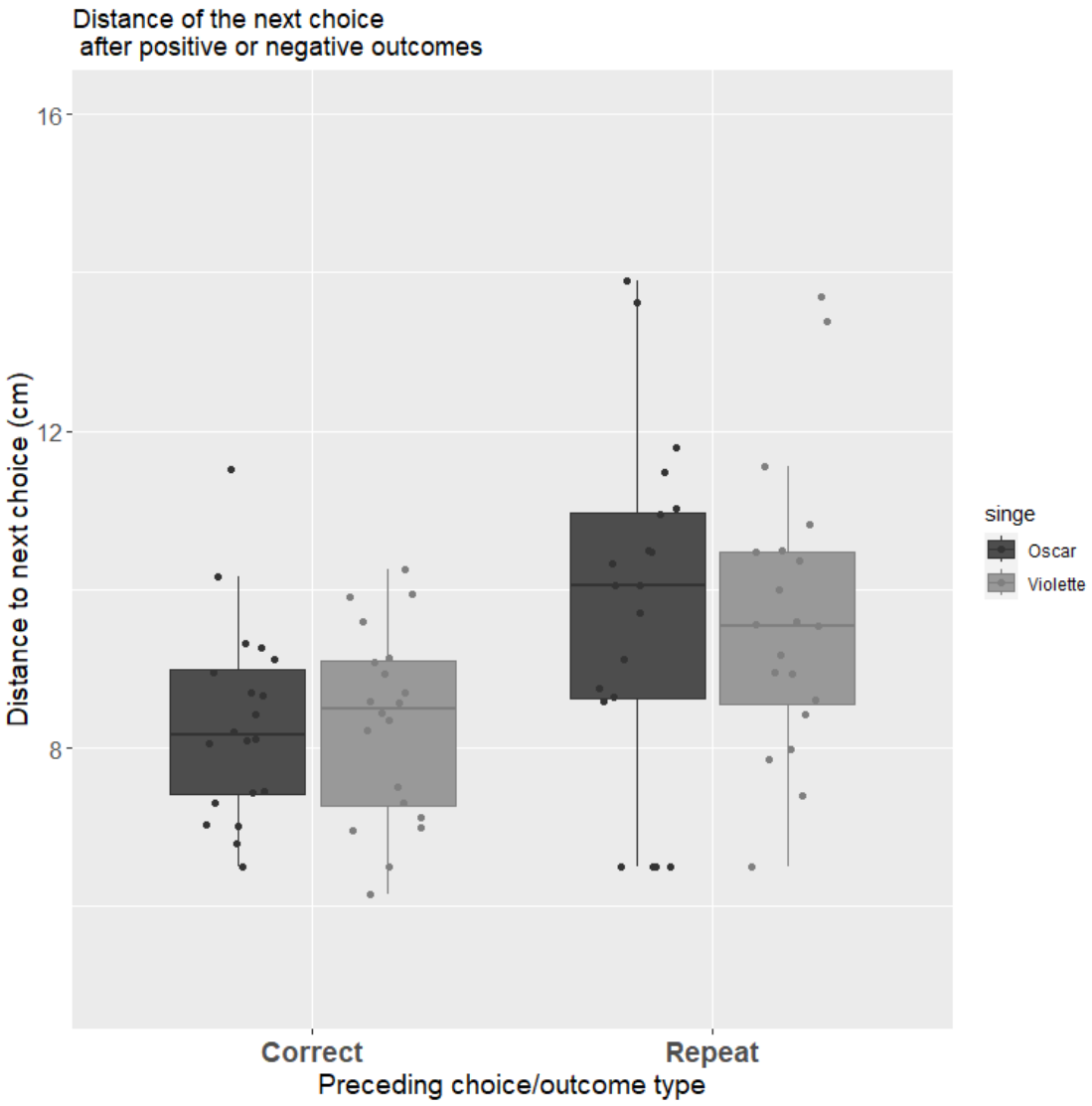
We remove Clear conditions, and misses because it's not appropriate.

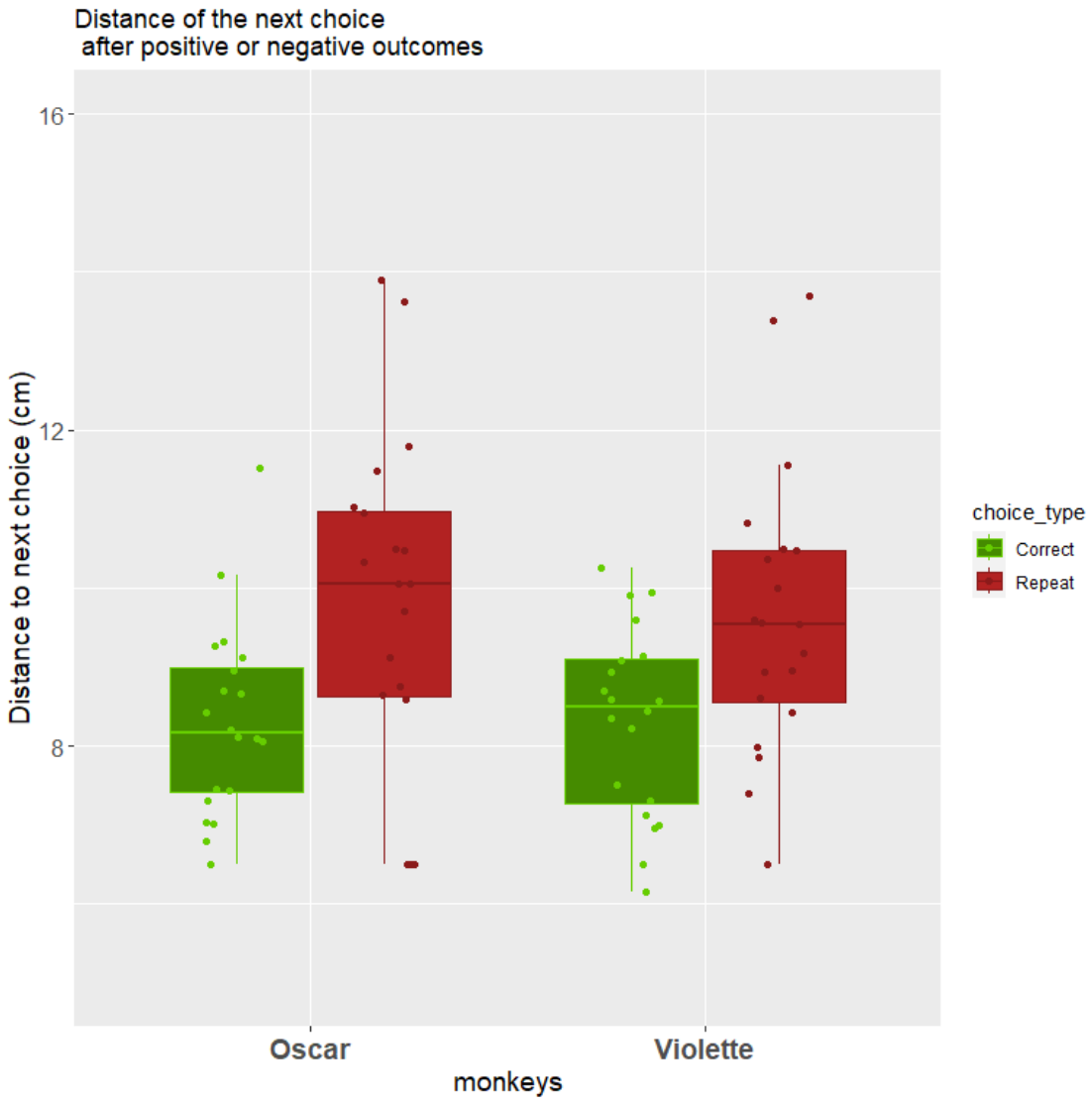

Distance of the next choice
after positive or negative outcomes
Oscar
Violette
sham
DCZ
Distance to next choice (cm)
25
20
15
10
5
1
2
3
4
quantile of trials in session
choice_type
Correct
Repeat


Distance of the next choice
after positive or negative outcomes
blue
sham
blue
DCZ
Oscar
Violette
Distance to next choice (cm)
16
12
8
Correct
Repeat
choice_type
Injection
sham
DCZ


Distance of the next choice
after positive or negative outcomes
sham
DCZ
Oscar
Violette
Distance to next choice (cm)
15.0
12.5
10.0
7.5
5.0
Correct
Repeat
choice_type
choice_type
Correct
Repeat


Distance of the next choice
after positive or negative outcomes
Early
sham
Early
DCZ
Late
sham
Late
DCZ
Oscar
blue
Violette
blue
Distance to next choice (cm)
16
12
8
Correct
Repeat
choice_type
Injection
sham
DCZ


Distance of the next choice
after positive or negative outcomes
16
12
8
Distance to next choice (cm)
Correct
Repeat
Preceding choice/outcome type
singe
Oscar
Violette




```
##
## Call:
## lm(formula = nextdist ~ choice_type * Injection, data = mean.nextdist)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3969 -1.0720 -0.0117  0.8434  4.4216
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     8.30804    0.25024  33.200  < 2e-16 ***
## choice_typeRepeat               1.39812    0.35389   3.951 0.000118 ***
## InjectionDCZ                    0.03725    0.35389   0.105 0.916306
## choice_typeRepeat:InjectionDCZ  0.15344    0.50048   0.307 0.759573
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.583 on 156 degrees of freedom
## Multiple R-squared:  0.1834, Adjusted R-squared:  0.1677
## F-statistic: 11.68 on 3 and 156 DF,  p-value: 6.033e-07

##
## Call:
## lm(formula = nextdist ~ choice_type, data = subset(mean.nextdist,
##     singe == "Violette" & Injection == "sham"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1638 -1.0962 -0.0216  0.8076  4.0347
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         8.3096     0.3429  24.231  < 2e-16 ***
## choice_typeRepeat   1.3542     0.4850   2.792  0.00815 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.534 on 38 degrees of freedom
## Multiple R-squared:  0.1703, Adjusted R-squared:  0.1484
## F-statistic: 7.797 on 1 and 38 DF,  p-value: 0.008146

##
## Call:
## lm(formula = nextdist ~ choice_type, data = subset(mean.nextdist,
##     singe == "Oscar" & Injection == "sham"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2485 -1.0249  0.0325  0.8563  4.1490
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         8.3065     0.3953   21.02   <2e-16 ***
## choice_typeRepeat   1.4420     0.5590    2.58   0.0139 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.768 on 38 degrees of freedom
## Multiple R-squared:  0.149,  Adjusted R-squared:  0.1266
## F-statistic: 6.655 on 1 and 38 DF,  p-value: 0.01388

##
## Call:
## lm(formula = nextdist ~ choice_type * phase, data = subset(mean.nextdist,
##     singe == "Violette" & Injection == "sham"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

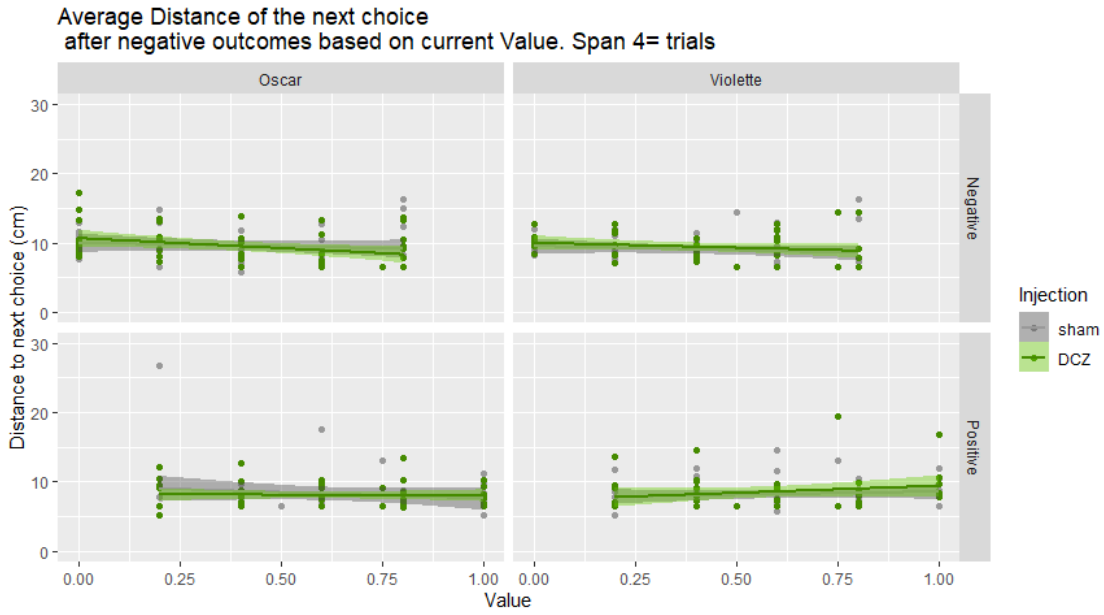
```
## -3.3106 -0.9186 -0.0769  0.8780  3.8880
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    8.7493     0.4861  17.997   <2e-16 ***
## choice_typeRepeat              1.0613     0.6875   1.544    0.131
## phaseLate                     -0.8794     0.6875  -1.279    0.209
## choice_typeRepeat:phaseLate    0.5859     0.9723   0.603    0.551
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.537 on 36 degrees of freedom
## Multiple R-squared:  0.2101, Adjusted R-squared:  0.1443
## F-statistic: 3.193 on 3 and 36 DF,  p-value: 0.035

##
## Call:
## lm(formula = nextdist ~ choice_type * phase, data = subset(mean.nextdist,
##     singe == "Oscar" & Injection == "sham"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6649 -0.7182 -0.0217  0.6930  4.7325
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    7.9959     0.5528  14.466   <2e-16 ***
## choice_typeRepeat              1.1690     0.7817   1.495    0.144
## phaseLate                      0.6212     0.7817   0.795    0.432
## choice_typeRepeat:phaseLate    0.5460     1.1055   0.494    0.624
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.748 on 36 degrees of freedom
## Multiple R-squared:  0.2117, Adjusted R-squared:  0.146
## F-statistic: 3.222 on 3 and 36 DF,  p-value: 0.0339

##
## Call:
## lm(formula = nextdist ~ choice_type * Injection, data =
subset(mean.nextdist,
##     singe == "Violette"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1638 -1.2054 -0.0372  0.8421  4.2027
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      8.3096     0.3507  23.692  < 2e-16 ***
```

```
## choice_typeRepeat                1.3542     0.4960   2.730  0.00786 **
## InjectionDCZ                     0.2546     0.4960   0.513  0.60923
## choice_typeRepeat:InjectionDCZ   0.1853     0.7015   0.264  0.79232
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.569 on 76 degrees of freedom
## Multiple R-squared:  0.1921, Adjusted R-squared:  0.1602
## F-statistic: 6.023 on 3 and 76 DF,  p-value: 0.0009774

##
## Call:
## lm(formula = nextdist ~ choice_type * Injection, data =
subset(mean.nextdist,
##     singe == "Violette"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1638 -1.2054 -0.0372  0.8421  4.2027
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      8.3096     0.3507  23.692  < 2e-16 ***
## choice_typeRepeat                1.3542     0.4960   2.730  0.00786 **
## InjectionDCZ                     0.2546     0.4960   0.513  0.60923
## choice_typeRepeat:InjectionDCZ   0.1853     0.7015   0.264  0.79232
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.569 on 76 degrees of freedom
## Multiple R-squared:  0.1921, Adjusted R-squared:  0.1602
## F-statistic: 6.023 on 3 and 76 DF,  p-value: 0.0009774
```

We observe a post-error effect: after a repeat (negative outcome) the next choice is further away than after a rewarded choice.The effect is however not significant for OSCAR, and there is no DCZ effect.

Other option is to check whether the probability to do a short vs. long shift depends on the previous reward, using logistic regressions.

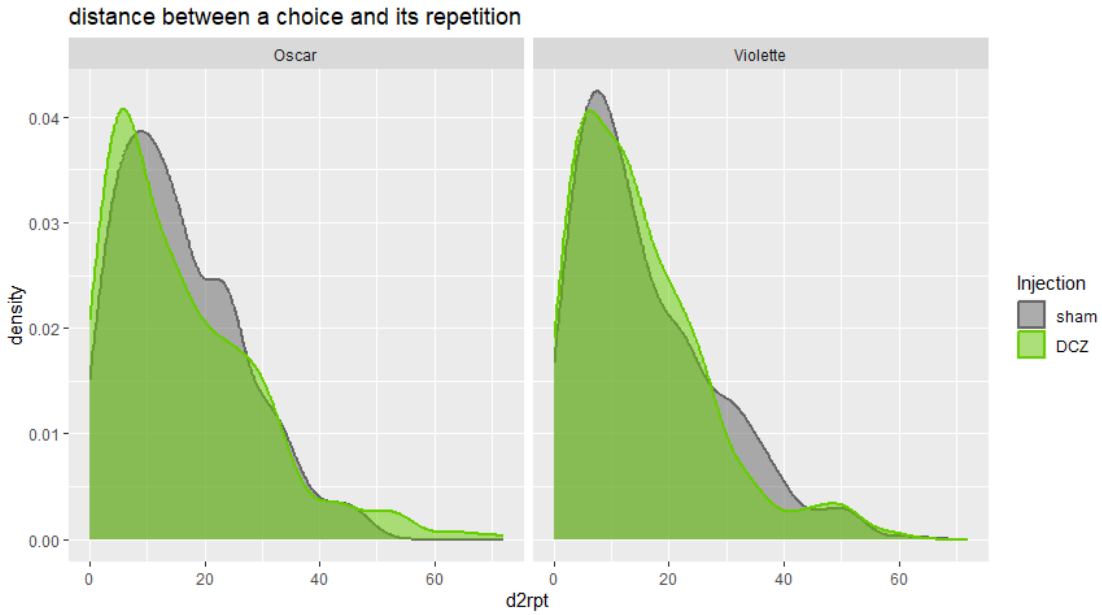
Now how does this work with successive trials : does the cumulative outcome (e.g. average outcome in 5 last choices) impact the distance of leave after a negative outcome? The hypothesis is that there should be a threshold to “leave the patch” in terms of average reward encountered. We don’t look first at average values but at successions of rwd: ..010, .0110, 01110, 11110




```
##
## Call:
## lm(formula = distNeg ~ Value * Injection, data =
subset(MeanValueNeg.nextdist,
##     singe == "Violette" & Feedback == "Negative"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8031 -1.4726 -0.3931  0.7742  7.6836
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          9.5532     0.5337  17.901   <2e-16 ***
## Value               -1.2336     1.1075  -1.114    0.268
## InjectionDCZ         0.4517     0.7534   0.599    0.550
## Value:InjectionDCZ  -0.1700     1.5453  -0.110    0.913
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.173 on 99 degrees of freedom
## Multiple R-squared:  0.03566,    Adjusted R-squared:  0.006433
## F-statistic:  1.22 on 3 and 99 DF,  p-value: 0.3065

##
## Call:
## lm(formula = distNeg ~ Value * Injection, data =
subset(MeanValueNeg.nextdist,
##     singe == "Oscar" & Feedback == "Negative"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -3.7251 -2.0081 -0.5137  1.0987  7.0839
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         10.0148     0.6399  15.650   <2e-16 ***
## Value               -1.0609     1.2931  -0.820    0.414
## InjectionDCZ         0.6689     0.8906   0.751    0.454
## Value:InjectionDCZ  -1.8439     1.7971  -1.026    0.307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.533 on 96 degrees of freedom
## Multiple R-squared:  0.0599, Adjusted R-squared:  0.03052
## F-statistic: 2.039 on 3 and 96 DF,  p-value: 0.1136
```

##Distance to repeat




```
##
## Call:
## glm(formula = d2rpt ~ Injection * portes, family = "poisson",
##     data = subset(stats.repeat, singe == "Oscar"))
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               2.66303    0.03627  73.415  < 2e-16 ***
## InjectionDCZ              0.10243    0.05241   1.955  0.05063 .
## portesblue                0.12373    0.03923   3.154  0.00161 **
## InjectionDCZ:portesblue -0.11461    0.05629  -2.036  0.04176 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 5972.2  on 685  degrees of freedom
## Residual deviance: 5961.8  on 682  degrees of freedom
## AIC: 8905.7
##
## Number of Fisher Scoring iterations: 5

##
## Call:
## glm(formula = d2rpt ~ Injection * portes, family = "poisson",
##     data = subset(stats.repeat, singe == "Violette"))
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               2.84620    0.03592  79.235   <2e-16 ***
## InjectionDCZ             -0.15949    0.04925  -3.238   0.0012 **
## portesblue               -0.08280    0.03785  -2.188   0.0287 *
## InjectionDCZ:portesblue   0.11478    0.05238   2.191   0.0284 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 7929.5  on 922  degrees of freedom
## Residual deviance: 7912.5  on 919  degrees of freedom
## AIC: 11848
##
## Number of Fisher Scoring iterations: 5

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: glm(formula = d2rpt ~ -1 + BV, family = "poisson", data = subset(d,
##     singe == "Oscar"))
##
## Linear Hypotheses:
##                               Estimate Std. Error z value Pr(>|z|)
## DCZ.clear - sham.clear == 0  0.102435   0.052406   1.955  0.19096
## sham.blue - sham.clear == 0  0.123728   0.039231   3.154  0.00789 **
## DCZ.blue - sham.clear == 0   0.111557   0.038924   2.866  0.01974 *
## sham.blue - DCZ.clear == 0   0.021293   0.040668   0.524  0.94990
## DCZ.blue - DCZ.clear == 0    0.009122   0.040372   0.226  0.99560
## DCZ.blue - sham.blue == 0   -0.012171   0.020556  -0.592  0.92968
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: glm(formula = d2rpt ~ -1 + BV, family = "poisson", data = subset(d,
##     singe == "Violette"))
##
## Linear Hypotheses:
##                               Estimate Std. Error z value Pr(>|z|)
## DCZ.clear - sham.clear == 0 -0.15949    0.04925  -3.238  0.00591 **
## sham.blue - sham.clear == 0 -0.08280    0.03785  -2.188  0.11501
## DCZ.blue - sham.clear == 0  -0.12751    0.03830  -3.330  0.00416 **
## sham.blue - DCZ.clear == 0   0.07668    0.03574   2.146  0.12648
## DCZ.blue - DCZ.clear == 0    0.03198    0.03621   0.883  0.80046
## DCZ.blue - sham.blue == 0   -0.04471    0.01784  -2.505  0.05301 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Regarding distances between a choice and a repeat, DCZ effects are absent in VIOLET but present in OSCAR. For OSCAR, the distances are longer under DCZ in blue conditions but much shorter in DCZ than sham in clear conditions.