20092603

A1, B5, C2

A1, 20092603

For the sequence of independent Bernoulli trials we will denote the probability of success as "p", the number of trials as "x", the number of fails as "r".

The difference between the negative binomial distribution and the functions dnbinom and pnbinom is that the latter account for the total number of fails instead for the total number of trials‘. Since we want to end up with 2 successes, the total number of trials will be the the output of the function +2.

- First, we will define the parameters for the negative binomial distribution.
- Then, we will compute the probabilities for several numbers of total trials and the 99.99% percentile of the distribution.
- We will then proceed to plot the PMF and CDF to get an intuition on the data.
- Finally, we will present the PMF and CDF in tabular form.

```
number_successes=2 #number of successes for the negative binomial function
fails = 0:10 #values of fails for which I want the probabilities
#   This can also be interpreted as total trials from 2 to 12
prob_success = 0.75 # we define the probability of success

probs = dnbinom(fails, number_successes, prob = prob_success)
#   probabilities from 0 to 10 failures
#   until we get two successes, with prob of success 0.75
#   or probabilities until two successes for total number of trials 2 to 12

#   We now want to find the 99.99% quantile of the negative binomial distribution.
#   We use qnbinom with parameters: quantile, no. of succeesses and probability
#   of success.
quant = 0.9999 # defining the 99.99% quantile

cat("The 99.99% percentile of the negative binomial distribution with r=",
    number_successes,"p=",prob_success , "is",
    qnbinom(quant,number_successes, prob_success)+2, "trials.")
```

```
## The 99.99% percentile of the negative binomial distribution with r= 2 p= 0.75 is 10 trials.
```

```
cum_probs = cumsum(probs) # calculating CDF

#   creating a data frame with the number of trials, their respective probabilities and
#   cumulative probabilities for tabular illustration.
neg_binom = data.frame(2:12,probs,cum_probs)
```

```r
#   Assigning column names
colnames(neg_binom) = c("Number of Trials", "Probabilities", "Cumulative Probabilities")

#   Using kable function to better illustrate the data in tabular form
knitr::kable(neg_binom[1:8,], align="c",digits=3,  #3 decimals
             caption = "The values of the PMF and CDF rounded to 3 decimals")
```

Table 1: The values of the PMF and CDF rounded to 3 decimals

Number of Trials	Probabilities	Cumulative Probabilities
2	0.562	0.562
3	0.281	0.844
4	0.105	0.949
5	0.035	0.984
6	0.011	0.995
7	0.003	0.999
8	0.001	1.000
9	0.000	1.000

At the precision of 3 decimals, we would conclude that the 99.99% percentile will be at 8 trials instead of 10.

B5, 20092603

1. Read the data into R.
2. Create a frequency table of the Premiership results (i.e. a table showing the numbers of matches with x home goals and y away goals, for x=0, 1, ... and y=0, 1, ...).
3. Create a scatterplot of the results. In the comments to your script, explain what is the problem with this graph.
4. Create a sunflowerplot for the data. Assign the output from sunflowerplot to an object.
5. Using the results from part (4) and the function symbols, create a plot where the size of each symbol corresponds to the frequency of each result. Colour code the symbols so that each colour corresponds to intervals of 0 - 9 matches, 10 - 19 matches, etc.

```r
#   1. Read the data
PLgoals = read.table("premgoals.dat", header=T)
#   examine the structure
str(PLgoals)
```

```
## 'data.frame':    380 obs. of  2 variables:
##  $ Home: int  2 0 1 0 2 1 1 5 5 0 ...
##  $ Away: int  1 2 1 1 1 0 1 1 1 2 ...
```

```r
#   2. Frequency table of home and away goals
table(x=PLgoals$Home, y=PLgoals$Away)
```

```
##    y
## x    0  1  2  3  4  5
##   0 41 28 17  7  5  2
##   1 35 42 28  6  4  0
```

```
##   2 26 34 19  5  0  0
##   3 12 16 12  5  3  0
##   4  8  7  5  2  1  0
##   5  2  3  2  1  0  0
##   6  0  1  1  0  0  0
```

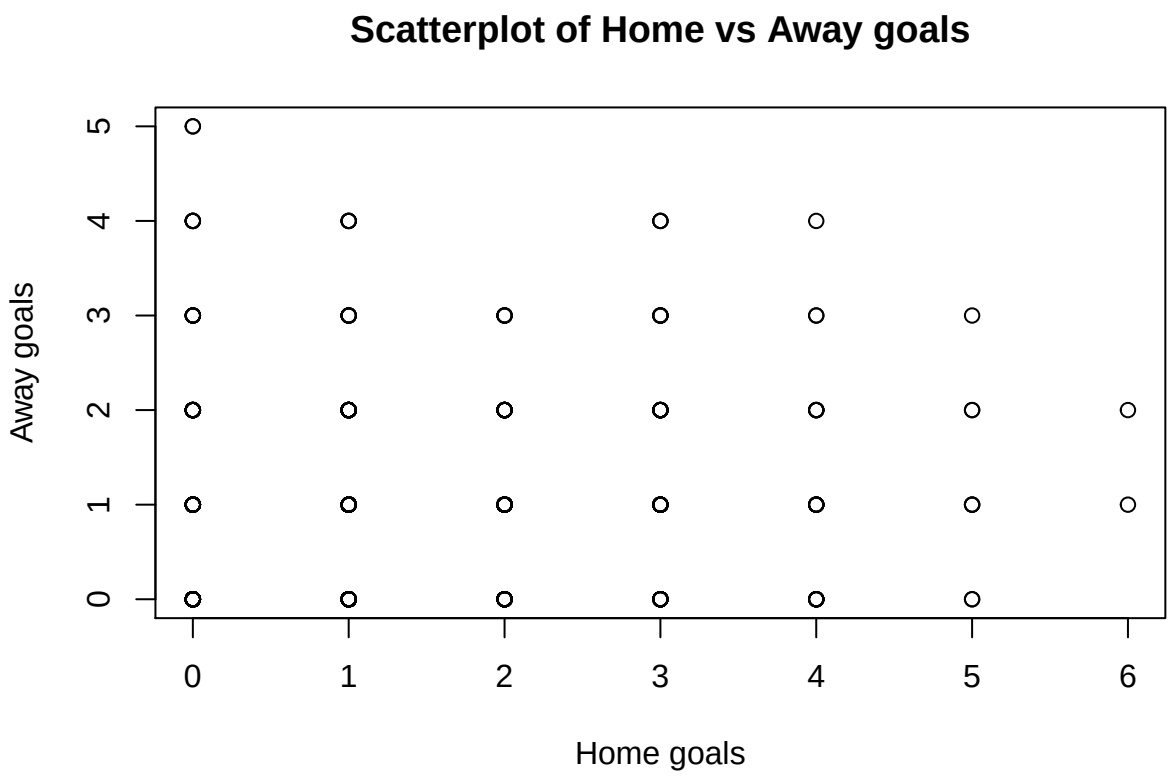
```
t = table(PLgoals$Home, PLgoals$Away)

#   Using kable function for better format
knitr::kable(t , align = "c",
             caption = "Frequency Table of the home vs away goals")
```

Table 2: Frequency Table of the home vs away goals

	0	1	2	3	4	5
0	41	28	17	7	5	2
1	35	42	28	6	4	0
2	26	34	19	5	0	0
3	12	16	12	5	3	0
4	8	7	5	2	1	0
5	2	3	2	1	0	0
6	0	1	1	0	0	0

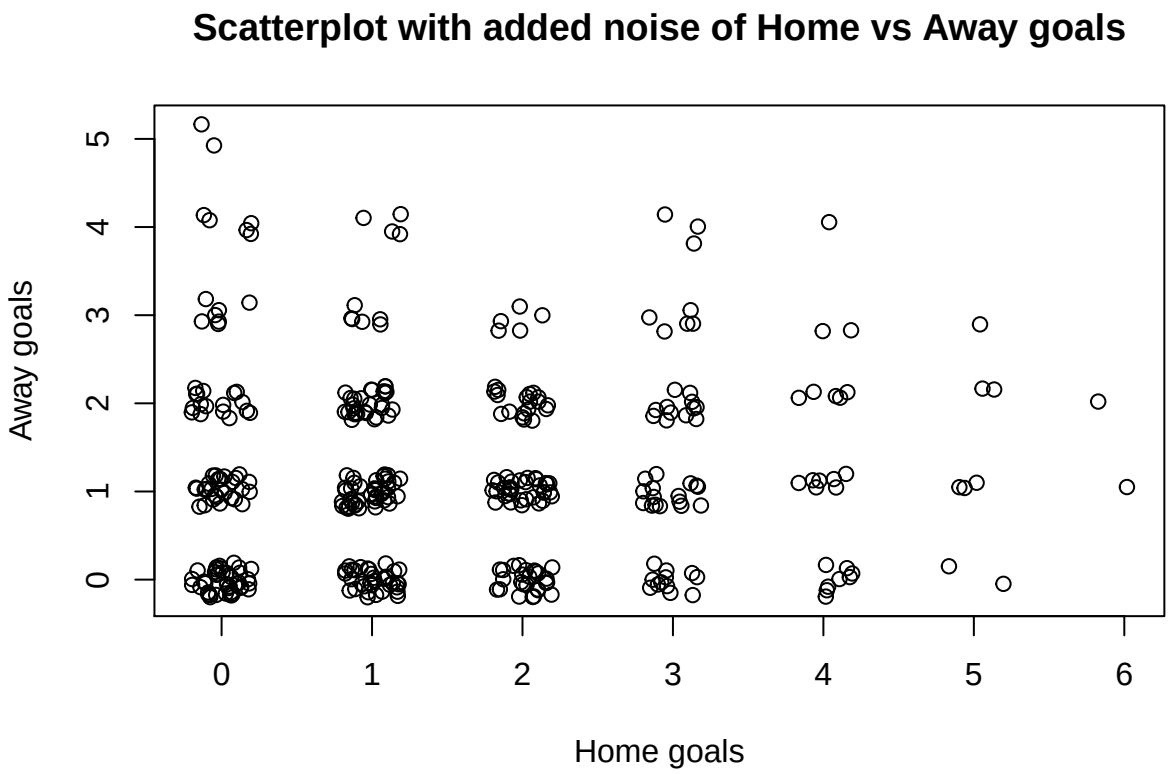
```
#   3. scatterplot of home vs away goals
plot(x=PLgoals$Home, y=PLgoals$Away, main="Scatterplot of Home vs Away goals", xlab="Home goals",
  ylab="Away goals"     )
```



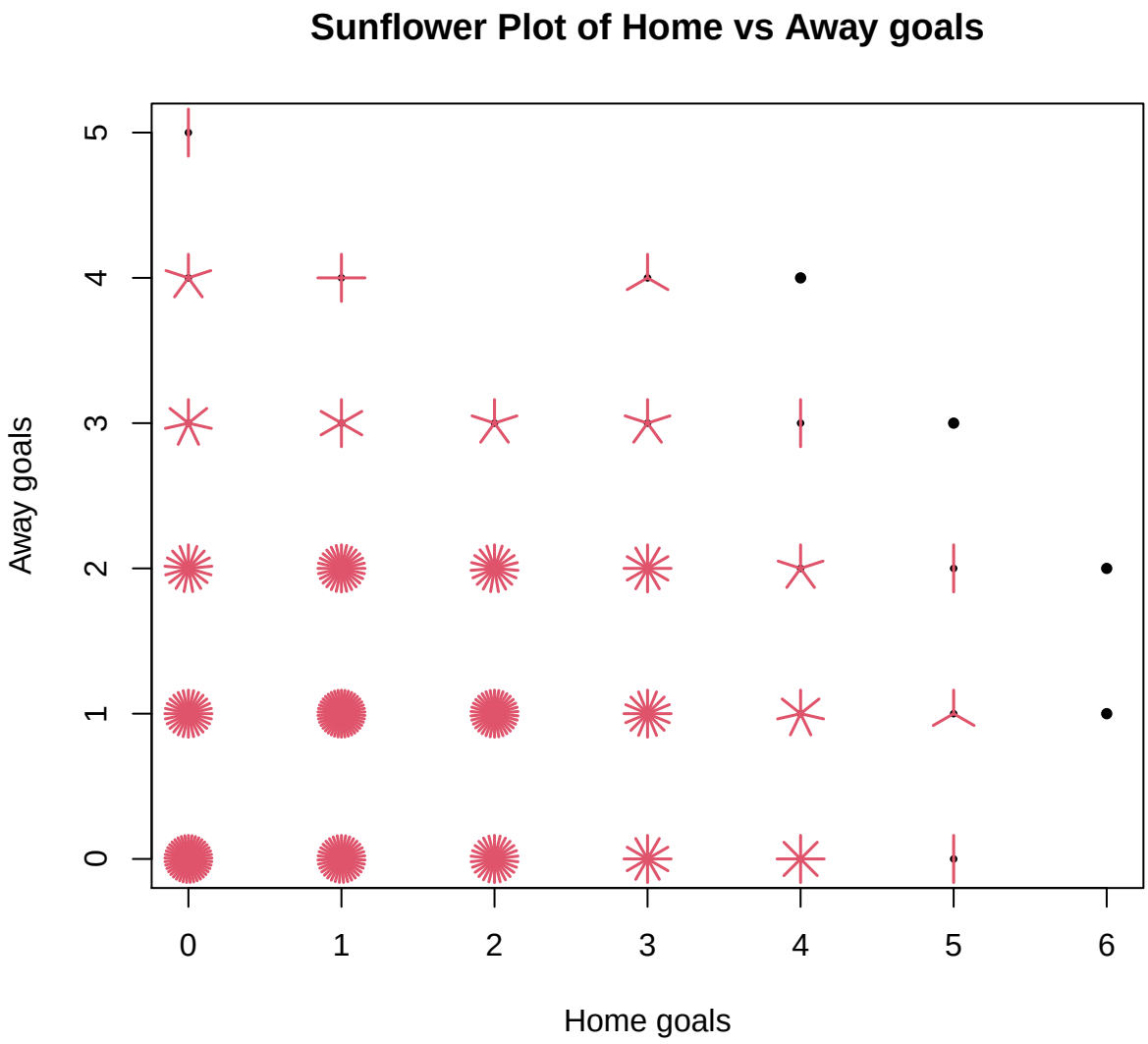
The variables take only integer values which leads to having just one point displayed for each pair (i,j) instead for the actual number of such pairs. This means that even if we encounter one pair once or multiple times we wouldn't be able to tell by the scatterplot.

The first way that I thought of to solve or at least examine this problem is to add a little noise to the data in the previous scatterplot. Then we can see that each point of the previous plot now corresponds to a cluster of points.

```
#   scatter plot with added noise to reveal clusters of points
plot(y=jitter(PLgoals$Away),x=jitter(PLgoals$Home),
     main="Scatterplot with added noise of Home vs Away goals", xlab="Home goals",
     ylab="Away goals" )
```




```
#   4. Creating a sunflower plot and assigning it to sun_plot
sun_plot = sunflowerplot(x=PLgoals$Home,y=PLgoals$Away,
                         main="Sunflower Plot of Home vs Away goals",
                         xlab="Home goals", ylab="Away goals" )
```



Of course a better way to visualize the data would be a sunflower plot compared to the scatterplot.

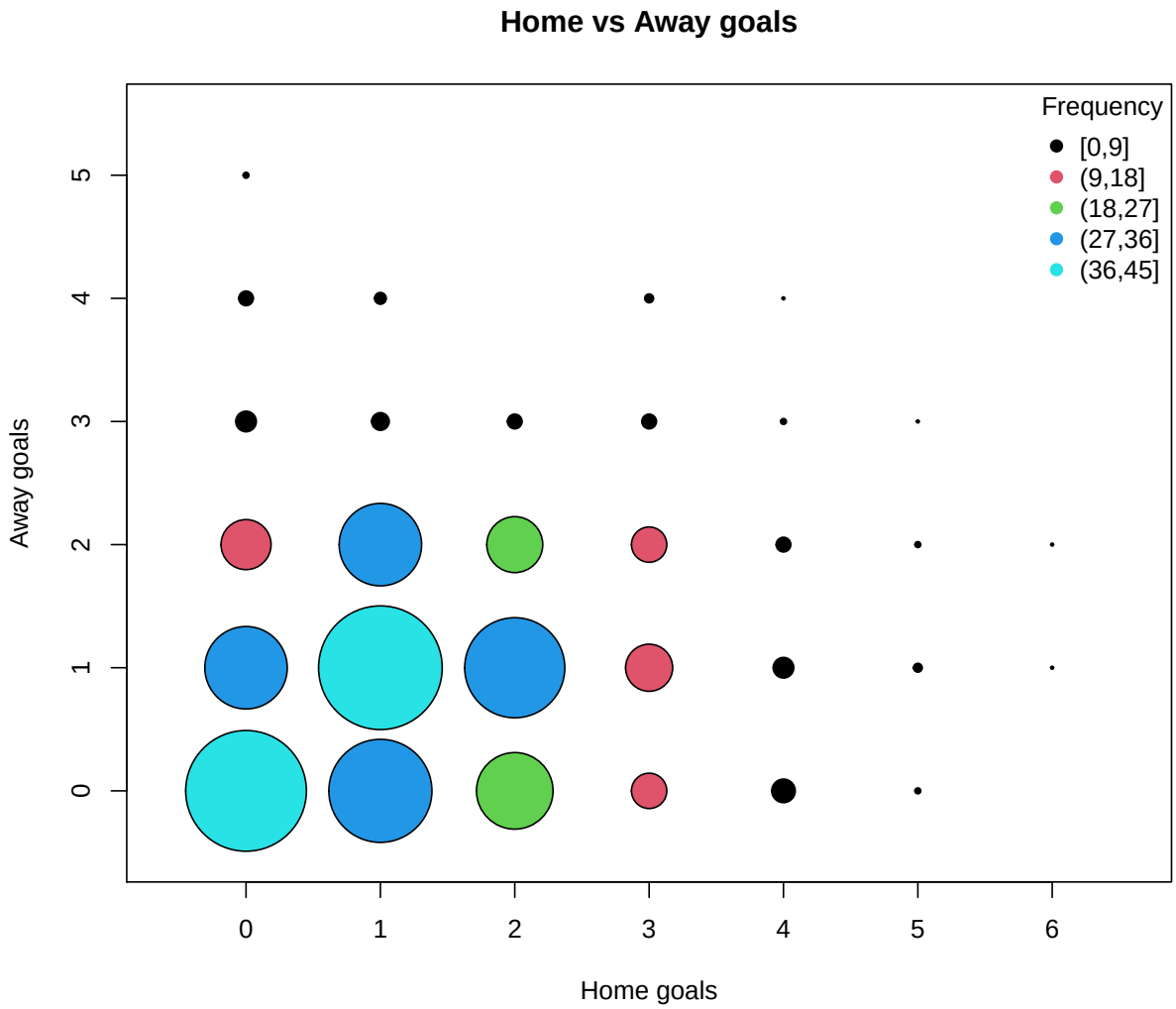
```
#   5. Creating a new plot where each circle's will correspong to the frequency of that point
#   We will create bins of the frequencies and each bin will have a different colour
#   on the plot. The bins will be 0-9,10-19,...until the max value of freq

#   Storing the frequencies of the (i,j) pairs to use later
freq = sun_plot$number

#   Creating the bins needed for the colours
bins = cut(freq, breaks = seq(0,max(freq)+9,9), include.lowest = T)

#   Using symbols function to create the plot. The biggest circle will have a radius
#   of 0.4 inches and will be coloured as discussed previously.
symbols(x=sun_plot$x,y=sun_plot$y, circles  = freq, inches = 0.4,
        fg="black",bg=bins, xlab = "Home goals", ylab="Away goals",
        main= "Home vs Away goals")
```

```r
#   A legend is needed to explain the colour notation
legend("topright", legend = levels(bins), title="Frequency",
       col = 1:length(levels(bins)), pch = 19, bty = "n")
```



We can now clearly see which pairs occur more frequently and which do not, both by the different size and colour.

C2, 20092603

1. Graphically, compare the two groups of patients at each time point. Also, compare the differences in plasma concentration between the two times. If you require more than one plot for these comparisons, all plots should appear on the same page.
2. Use the most appropriate t tests to assess the strength of evidence for differences between the two groups as regards (i) plasma concentrations at each time point (ii) the change in plasma concentration between time points.
3. Repeat the analysis of part (2) using nonparametric methods.

4. Briefly, summarise your findings.

```
#   Importing ggplot for the plots and tidyr
library(ggplot2)
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.0.3
```

```
#   Reading the data and looking at their structure
blood = read.table("vitaminc.dat", header = T)
str(blood)
```

```
## 'data.frame':    35 obs. of  3 variables:
##  $ Schiz  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Plasma0: num  1.27 0.09 1.64 0.23 0.18 0.12 0.85 0.69 0.78 0.63 ...
##  $ Plasma2: num  2 0.41 2.37 0.41 0.79 0.94 1.72 1.75 1.6 1.8 ...
```

```
#   1.
#   Adding a new column with the difference for each patient between the two measurements
blood2 = data.frame(blood, diff = blood$Plasma2-blood$Plasma0)
#   We should note that the differences are all positive which means we have an increase
#   in the concentration. To be clearer we will refer to this difference as "increase"

#   Transforming the wide data.frame blood2 to a long one by using the gather function
#   from the tidyr package
blood_long2 = gather(blood2,time,Plasma,Plasma0:diff,factor_key = T)

#   Examining its structure
str(blood_long2)
```

```
## 'data.frame':    105 obs. of  3 variables:
##  $ Schiz : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ time  : Factor w/ 3 levels "Plasma0","Plasma2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Plasma: num  1.27 0.09 1.64 0.23 0.18 0.12 0.85 0.69 0.78 0.63 ...
```

```
#   Making the groups factor
blood_long2$Schiz = factor(blood_long2$Schiz)
#   Changing the column names
colnames(blood_long2) = c("Group", "Time","Plasma")
#   Changing the levels' labels
levels(blood_long2$Time) = c("Initial Measurement","Second Measurement", "Increase")
levels(blood_long2$Group) = c("Control", "Schizophrenic")
```

```
#   Using ggplot2 package to produce violin plots to have a better understanding of
#   how the points are distributed in each group at each time point
#   We will compare the two groups before the treatment, after the treatment and
#   also their increases in the plasma concentrations
ggplot(blood_long2, aes(x=Group, y=Plasma)) +
  geom_violin(trim = F, aes(fill=Group))+ #violin plot with dotplot
  geom_dotplot(binaxis='y', stackdir='center', dotsize = 0.5, aes(fill=Group))+
  facet_wrap(~Time)+ #create 3 plots for the first and second measurement and the increases
```

```
  theme_bw()+ # a rather minimalistic theme
  # each violin plot will display its mean value just for reference, will be tested
  #later for significant differences
  stat_summary(geom = "point",fun = "mean", col = "black",size =3,shape = 23,fill = "grey")+
  theme(legend.position="bottom")+
  ylab("Plasma Concentration (mg/DL)")+
  xlab("")+
  ggtitle("Violin plots for each Group at each time point and the Increase between measurements",
          subtitle="The grey rhombus denotes the mean")
```

```
## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```

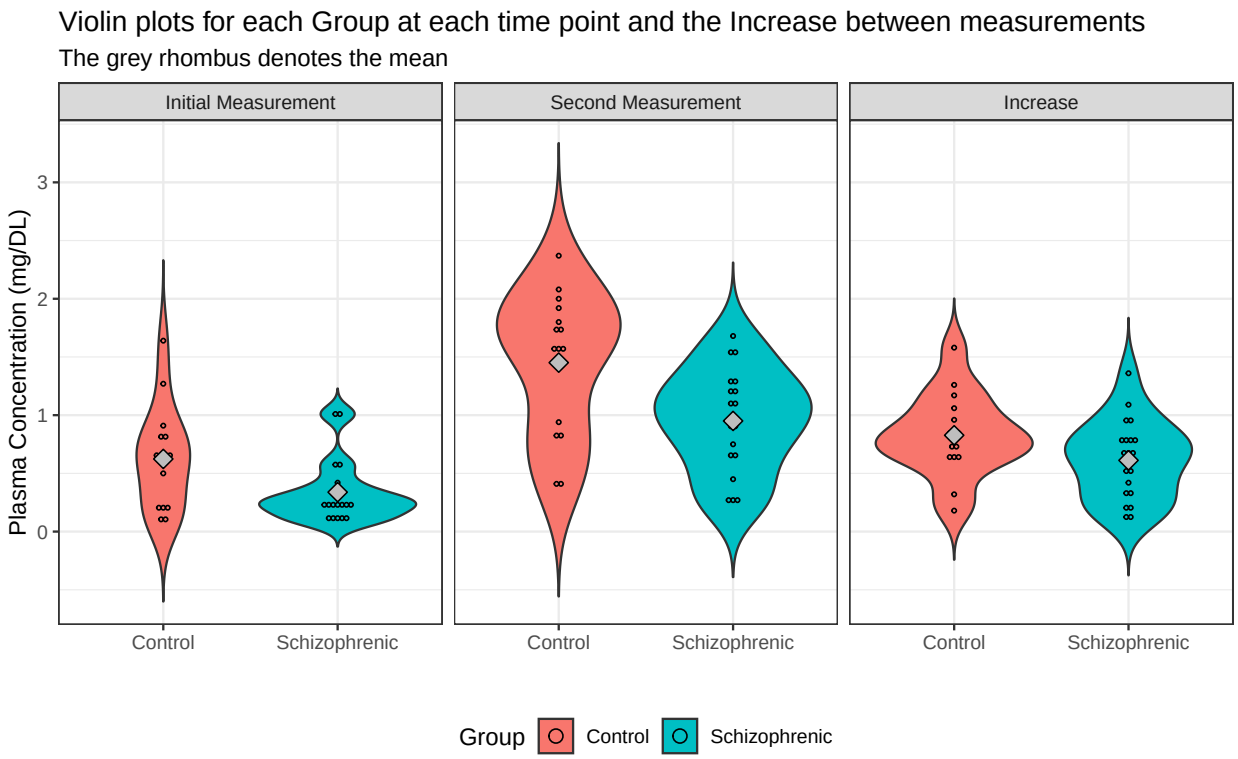


Figure 1: Violin plots for each Group at each time point and the difference between measurements for each group (The grey rhombus denotes the mean)

```
#   2. t-tests
#   All tests will be interpreted at significance level a=5%

#   two-sample t-test for the difference between groups at the first measurement
#   assuming equal variances
t1 = t.test(blood$Plasma0[blood$Schiz==1],blood$Plasma0[blood$Schiz==2],
       alternative = "two.sided", paired=F, var.equal = T)
t1
```

```
##
##  Two Sample t-test
```

```
##
## data:  blood$Plasma0[blood$Schiz == 1] and blood$Plasma0[blood$Schiz == 2]
## t = 2.3773, df = 33, p-value = 0.02339
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.04117026 0.52982974
## sample estimates:
## mean of x mean of y
##    0.6240    0.3385
```

```
#   p-value=0.023<0.05 hence the t-test suggests that there's a significant difference
#   between the measurements of the two groups to begin with.

#   two-sample t-test for the difference between groups at the second measurement
#   assuming equal variances
t2 = t.test(blood$Plasma2[blood$Schiz==1],blood$Plasma2[blood$Schiz==2],
       alternative = "two.sided", paired=F, var.equal = T)
t2
```

```
##
##  Two Sample t-test
##
## data:  blood$Plasma2[blood$Schiz == 1] and blood$Plasma2[blood$Schiz == 2]
## t = 2.8321, df = 33, p-value = 0.007821
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1410486 0.8606180
## sample estimates:
## mean of x mean of y
##  1.451333  0.950500
```

```
#   p-value=0.007<0.05 hence the t-test suggests that there's a significant difference
#   between the measurements of the two groups at the time of second measurement.
```

```
#   two-sample t-test for the difference in the increases of the concentrations of
#   plasma for the two groups
t3 = t.test(blood2$diff[blood2$Schiz==1],blood2$diff[blood2$Schiz==2],
       alternative = "two.sided", paired=F, var.equal = T)
t3
```

```
##
##  Two Sample t-test
##
## data:  blood2$diff[blood2$Schiz == 1] and blood2$diff[blood2$Schiz == 2]
## t = 1.831, df = 33, p-value = 0.07614
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.02393077  0.45459744
## sample estimates:
## mean of x mean of y
## 0.8273333 0.6120000
```

```r
#   p-value =0.076>0.05 hence we should accept the null hypothesis
#   that the increases of the concentrations observed are not significantly
#   different between the two groups.
```

```r
#   3. Non parametric tests
#   Mann-Whitney test for the difference between groups at the first measurement
np1 = wilcox.test(blood$Plasma0[blood$Schiz==1],blood$Plasma0[blood$Schiz==2],
            alternative = "two.sided", paired=F)
```

```
## Warning in wilcox.test.default(blood$Plasma0[blood$Schiz == 1],
## blood$Plasma0[blood$Schiz == : cannot compute exact p-value with ties
```

```r
np1
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  blood$Plasma0[blood$Schiz == 1] and blood$Plasma0[blood$Schiz == 2]
## W = 204, p-value = 0.07439
## alternative hypothesis: true location shift is not equal to 0
```

```r
#   The p-value=0.074>0.05 hence we accept the null hypotheses that there is no
#   significant difference between the measurements of the two groups at the time
#   of the first measurement.

#   Mann-Whitney test for the difference between groups at the second measurement
np2 = wilcox.test(blood$Plasma2[blood$Schiz==1],blood$Plasma2[blood$Schiz==2],
            alternative = "two.sided", paired=F)
```

```
## Warning in wilcox.test.default(blood$Plasma2[blood$Schiz == 1],
## blood$Plasma2[blood$Schiz == : cannot compute exact p-value with ties
```

```r
np2
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  blood$Plasma2[blood$Schiz == 1] and blood$Plasma2[blood$Schiz == 2]
## W = 225, p-value = 0.01298
## alternative hypothesis: true location shift is not equal to 0
```

```r
#   The p-value=0.0129<0.05 hence we reject the null hypotheses that there is no
#   significant difference between the measurements of the two groups at the time
#   of the second measurement.

#   Mann-Whitney test for the difference in the increases of the concentrations of
#   plasma for the two groups
np3 = wilcox.test(blood2$diff[blood2$Schiz==1],blood2$diff[blood2$Schiz==2],
            alternative = "two.sided", paired=F)
```

```
## Warning in wilcox.test.default(blood2$diff[blood2$Schiz == 1],
## blood2$diff[blood2$Schiz == : cannot compute exact p-value with ties
```

```
np3
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  blood2$diff[blood2$Schiz == 1] and blood2$diff[blood2$Schiz == 2]
## W = 199, p-value = 0.1059
## alternative hypothesis: true location shift is not equal to 0
```

```
#   The p-value=0.1059>0.05 hence we accept the null hypotheses that there is no
#   significant difference between the increses of the concentration of plasma in
#   the two groups.
```

4. At the significance level of 5%, the parametric tests suggest that there is a significant difference in measurements between the two groups, both before and after the treatment. However, if we consider the difference in measurements for each individual and then test for a significant difference between the two Groups we deduce that there is no significant difference. This suggests that the treatment works in a similar way for both Groups. The non parametric tests agree for the the second and third case, but suggest that there is no significant difference between the measurements of the two groups at the start.

All the p-values are summarised in the following table.

```
#  Putting the p-values of the parametric and non parametric tests into a matrix
tab = matrix(c(t1$p.value,t2$p.value, t3$p.value, np1$p.value,np2$p.value,np3$p.value),
             nrow = 3,byrow = F)
#   renaming rows and columns
colnames(tab) = c("Parametric","Non-parametric")
rownames(tab) = c("Diff. between Groups at start", "Diff. between Groups after 2 hrs",
                  "Diff. in increases between Groups")
#   printing a summary table of the p-values
knitr::kable(tab, align="c", digits=3,caption = "p-values of parametric and non-parametric tests")
```

Table 3: p-values of parametric and non-parametric tests

	Parametric	Non-parametric
Diff. between Groups at start	0.023	0.074
Diff. between Groups after 2 hrs	0.008	0.013
Diff. in increases between Groups	0.076	0.106