Project 2

Nikant Yadav

2024-06-02

COLLECTING AND ANALYSING DATA FROM EXPERIMENTS

For this project, we first install the following two packages - * tidyverse, to help with data manipulation * readxl, to import an Excel spreadsheet * knitr, to suppress extra irrelevant warnings

PART 2.2

We will now evaluate the effect of the punishment option on average contributions. We used a defualt dataset for this, which is in the form of a excel spreadsheet containing two tables.

- The first table shows average contributions in a public goods game without punishment.
- The second table shows average contributions in a public goods game with punishment.

Importing data file to R

```
suppressWarnings({
  library(readxl)
  library(tidyverse)
  data_N <-  read_excel("datafile-excel-project-2.xlsx", range= "A2:Q12")
  data_P <- read_excel("datafile-excel-project-2.xlsx", range="A16:Q26")
})
```

The code chunk is called inside *suppressWarnings* in order to avoid extra warnings to be showcased in the R markdown report. Now as the two data frames are imported in the project, lets see how they look.

```
head(data_N)
```

```
## # A tibble: 6 × 17
##   Period Copenhagen `Dnipropetrovs'k` Minsk `St. Gallen` Muscat Samara Zurich
##    <dbl>      <dbl>             <dbl> <dbl>        <dbl>  <dbl>  <dbl>  <dbl>
## 1      1       14.1              11.0 12.8          13.7   9.54   10.8  11.1
## 2      2       14.1              12.6 12.3          12.8  11.0    11.5  12.2
## 3      3       13.7              12.1 12.6          12.4  11.5    11.7  10.8
## 4      4       12.9              11.2 12.3          10.6  10.3    11.3  10.6
## 5      5       12.3              11.3 11.8          11.0   9.83   10.3   8.52
## 6      6       11.7              10.5  9.88         10.7  10.3    10.2   7.10
## # ℹ 9 more variables: Boston <dbl>, Bonn <dbl>, Chengdu <dbl>, Seoul <dbl>,
## #   Riyadh <dbl>, Nottingham <dbl>, Athens <dbl>, Istanbul <dbl>,
## #   Melbourne <dbl>
```

```
head(data_P)
```

```
## # A tibble: 6 × 17
##   Period Copenhagen `Dnipropetrovs'k` Minsk `St. Gallen` Muscat Samara Zurich
##    <dbl>      <dbl>             <dbl> <dbl>        <dbl>  <dbl>  <dbl>  <dbl>
## 1      1       15.4              9.48  11.8         15.0   9.21   10.8   13.2
## 2      2       17.0              9.91  13.2         16.7  10.3    11.3   15.0
## 3      3       17.7             11.8   12.9         17.6  10.1    11.7   15.8
## 4      4       18.2             11.5   13.4         17.4  10      11.8   16.3
## 5      5       18.4             12.7   14.0         17.6   9.58   11.2   16.4
## 6      6       18.7             11.8   13.0         17.3   9.90   12.2   16.6
## # ℹ 9 more variables: Boston <dbl>, Bonn <dbl>, Chengdu <dbl>, Seoul <dbl>,
## #   Riyadh <dbl>, Nottingham <dbl>, Athens <dbl>, Istanbul <dbl>,
## #   Melbourne <dbl>
```

We can see that in each period, the average contribution varies across countries. The mean and variance are two ways to summarize distributions.

Calculating mean contribution in each period separately

```
period_data <- vector("list", length = nrow(data_N))

for(i in 1:nrow(data_N)) {
  temp_vector <- c()
  for(k in 2:17) {
    temp_vector <- c(temp_vector, data_N[i, k])
  }
  period_data[[i]] <- unlist(unname(temp_vector))
}
```

We create a list *period_data*, and using nested loops and a temporary vector, basically convert the rows from 2:17 for each period into separate vectors in the list. So, now the data for any period, can be accessed by directly calling the period number inside the list *period_data*.

```
mean_values <- c()
for (i in 1:nrow(data_N)) {
  mean_values[i] <- mean(period_data[[i]])
}

print(mean_values)
```

```
##  [1] 10.578313 10.628398 10.407079  9.813033  9.305433  8.454844  7.837568
##  [8]  7.376388  6.392985  4.383769
```

We create an empty vector *mean_values*. Using a loop we calculate the mean for all the periods using the *mean* function of R

Similarly, mean values for different periods of *data_P* can be calculated. The code for it that follows the same steps as above, has been written at once.

```
period_dataP <- vector("list", length = nrow(data_N))


for(i in 1:nrow(data_P)) {
  temp_vector <- c()
  for(k in 2:17) {
    temp_vector <- c(temp_vector, data_P[i, k])
  }
  period_dataP[[i]] <- unlist(unname(temp_vector))
}

mean_valuesP <- c()
for (i in 1:nrow(data_P)) {
  mean_valuesP[i] <- mean(period_dataP[[i]])
}

print(mean_valuesP)
```

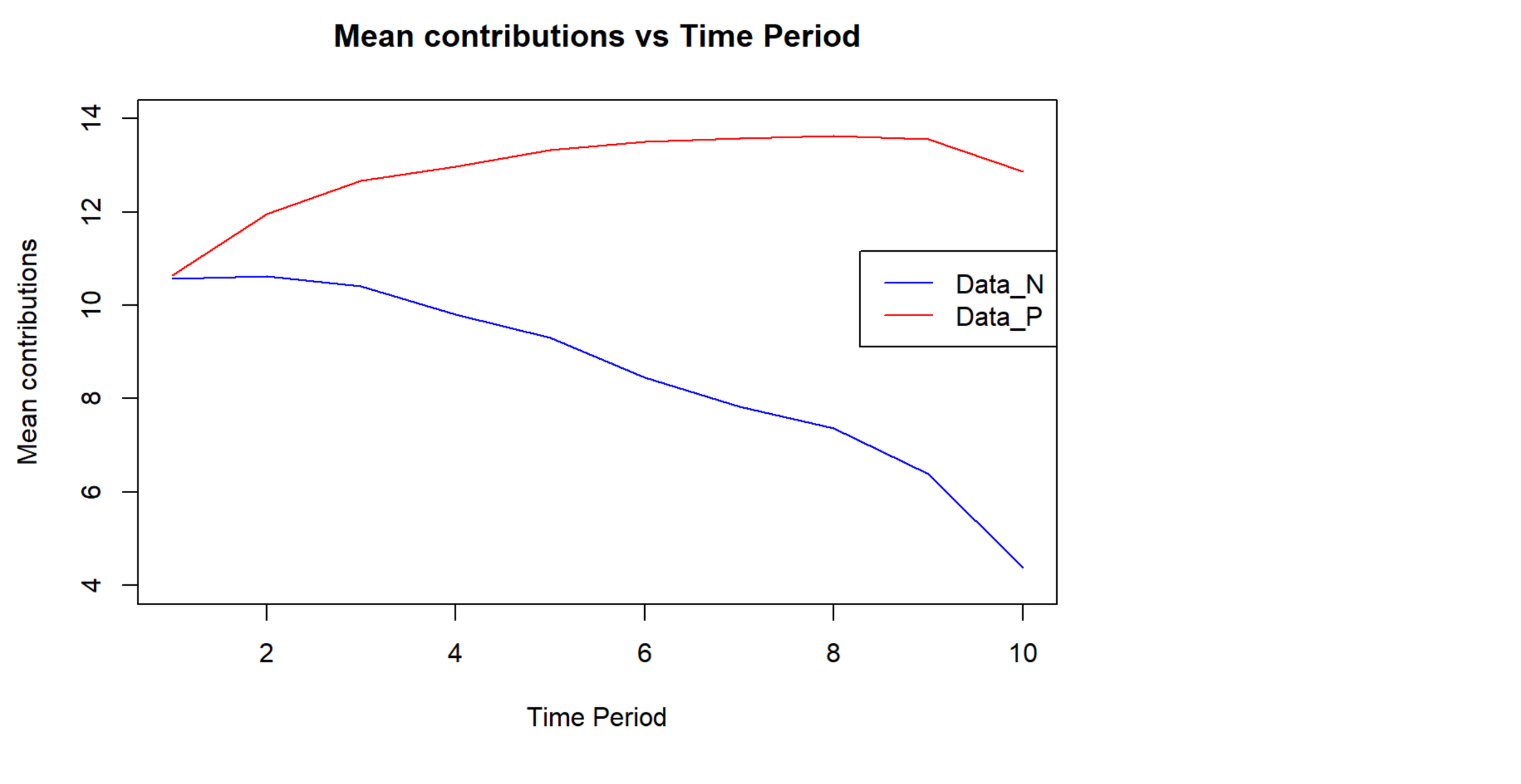
```
##  [1] 10.63876 11.95479 12.66434 12.96666 13.33164 13.50224 13.57468 13.63554
##  [9] 13.56955 12.86988
```

Plotting line chart of mean contribution on vertical axis and time period on horizontal axis

```
plot(1:nrow(data_N), mean_values, type = "l", xlab="Time Period",ylim = c(4, 14),  ylab="Mean contributions", main="Mean con
tributions vs Time Period", col="blue")

lines(1:nrow(data_P), mean_valuesP, type = "l", col = "red")

legend("topright", legend = c("Data_N", "Data_P"), col = c("blue", "red"), lty = 1, inset = c(0, 0.3))
```



We use the *plot* function to plot one line, and then add another line using the *line* function.

Observation

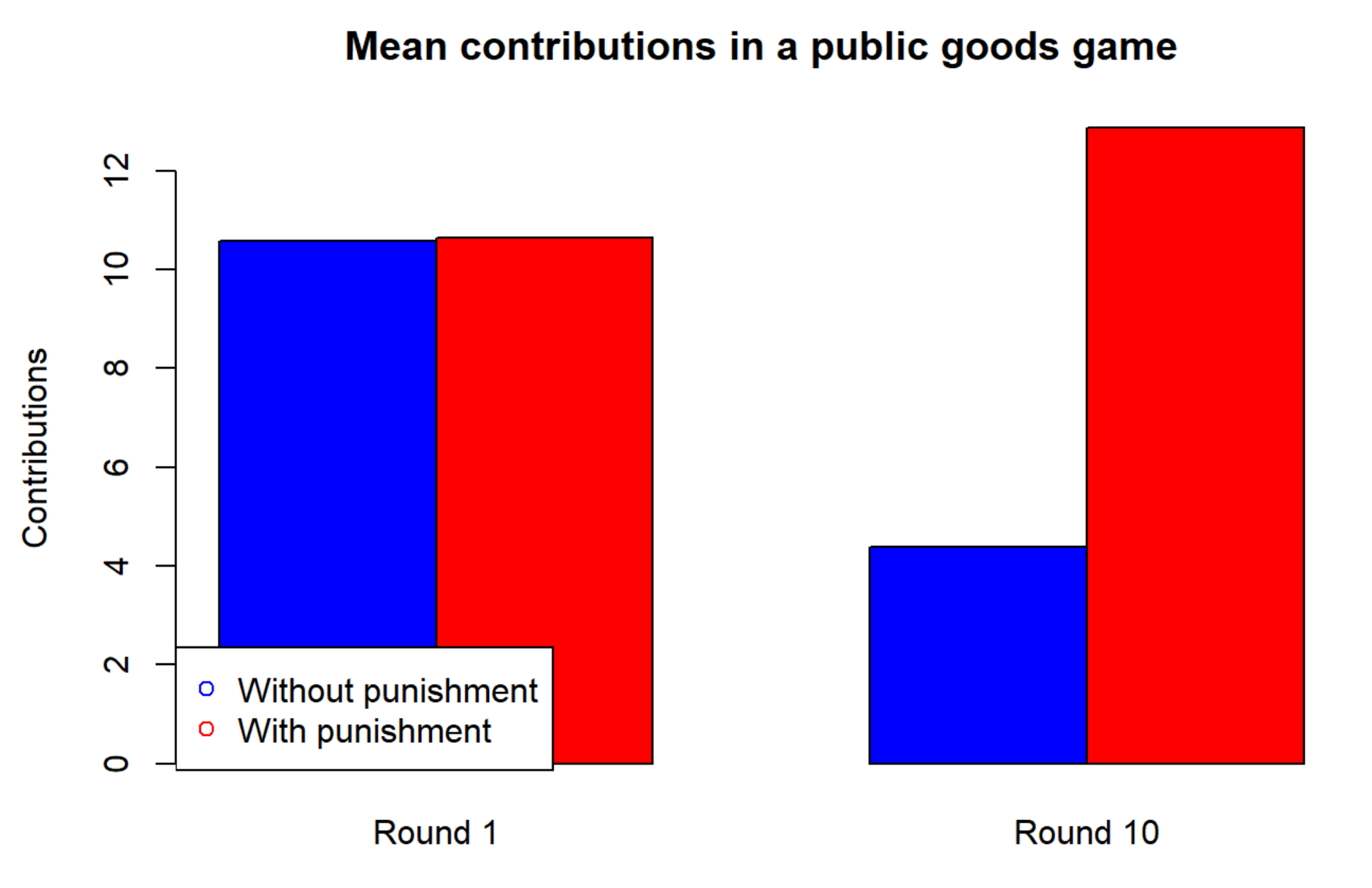
We can observe, that for experiment with punishment, the mean contribution increases whereas for experiment without punishment, the mean contribution falls steadily.

Creating column chart showing the mean contribution in the first and last period for both experiments

```
temp_d <- c(mean_values[1], mean_values[10], mean_valuesP[1], mean_valuesP[10])
temp <- matrix(temp_d, nrow=2, ncol=2, byrow=TRUE)

barplot(temp, main="Mean contributions in a public goods game", ylab="Contributions", beside = TRUE, col = c("Blue", "Red"),
names.arg = c("Round 1", "Round 10"))

legend("bottomleft", pch = 1, col = c("Blue", "Red"), c("Without punishment", "With punishment"))
```



We first create a temporary matrix *temp_d* containing the mean contributions of 1st and 10th period of both experiment with and without punishment. Then using the *barplot* function, two column charts for round(period) 1 and round(period) 10 are plotted, where the blue color denotes **without punishment** and red color denotes **with punishment**.

Observation

It can be observed easily, that for round 1, the mean contribution for both experiment is almost same. But as it goes to round 10, there is a vast difference in the mean contribution, with a much larger mean contribution been made in the experiment without punishment

We also need to know how 'spread out' the data is in order to get a clearer picture and make comparisons between distributions. The variance is one way to measure spread: the higher the variance, the more spread out the data is.

Calculating standard deviation for Periods 1 and 10 separately, for both experiments.

```
sdN1 <- sd(period_data[[1]])
sdN10 <- sd(period_data[[10]])

sdP1 <- sd(period_dataP[[1]])
sdP10 <- sd(period_dataP[[10]])

print(sdN1)
```

```
## [1] 2.020724
```

```
print(sdN10)
```

```
## [1] 2.187126
```

```
print(sdP1)
```

```
## [1] 3.207258
```

```
print(sdP10)
```

```
## [1] 3.89802
```

We simply use the *sd* function of R to easily calculate the standard deviation of period 1 and 10 using the existing vectors containing the data.

Thumb Rule: It means that most of the data (95% if there are many observations) will be less than two standard deviations away from the mean.

For experiment without punishment, in round 1, mean = 10.58. We get the interval for thumb rule as * [8.6, 12.6] for Period 1, without punishment * [7.5, 13.7] for Period 1, with punishment * [2.3, 6.5] for Period 10, without punishment * [9.1, 16.7] for Period 10, with punishment.

Inspecting the data, we can see that thumb rule applies here.

Observation

Having the same mean contributions in period 1, does not mean that both the data sets are same. The standard deviation for experiment with punishment is greater meaning that the data of experiment with punishment is spread out over a wider range of values around the mean.

Calculate the maximum and minimum value for Periods 1 and 10 separately, for both experiments.

```
rangeN1 <- range(period_data[[1]])
rangeN10 <- range(period_data[[10]])

rangeP1 <- range(period_dataP[[1]])
rangeP10 <- range(period_dataP[[10]])

print(rangeN1)
```

```
## [1]  7.958333 14.102941
```

```
print(rangeN10)
```

```
## [1] 1.300000 8.681818
```

```
print(rangeP1)
```

```
## [1]  5.818182 16.017857
```

```
print(rangeP10)
```

```
## [1]  6.204545 17.511906
```

Just like we did for standard deviation calculations, we simply use the *range* function of R to easily calculate the range of period 1 and 10 using the existing vectors containing the data. Range is the interval formed by the maximum and minimum value of the data set.

Creating a table of summary statistics that displays mean, standard deviation, minimum, maximum and range for Periods 1 and 10 and for both experiments.

```
print("Table for experiment without punishment")
```

```
## [1] "Table for experiment without punishment"
```

```
tabN <- matrix(c(1, mean_values[1], sdN1, rangeN1[1], rangeN1[2], abs(rangeN1[1]-rangeN1[2]), 10, mean_values[10], sdN10, ra
ngeN10[1], rangeN10[2], abs(rangeN10[1]-rangeN10[2])), ncol=6, byrow=TRUE)
colnames(tabN) <- c('Periods', 'Mean', 'Std Deviation', 'Minimum', 'Maximum', 'Range')
tabN <- as.table(tabN)

print(tabN)
```

```
##     Periods      Mean Std Deviation   Minimum   Maximum     Range
## A  1.000000 10.578313      2.020724  7.958333 14.102941  6.144608
## B 10.000000  4.383769      2.187126  1.300000  8.681818  7.381818
```

Here, we first create a matrix with 6 columns, and then use the function *as.table* to convert that matrix into a table. Similar table can be created for experiment with punishment.

```
print("Table for experiment with punishment")
```

```
## [1] "Table for experiment with punishment"
```

```
tabP <- matrix(c(1, mean_valuesP[1], sdP1, rangeP1[1], rangeP1[2], abs(rangeP1[1]-rangeP1[2]), 10, mean_valuesP[10], sdP10,
rangeP10[1], rangeP10[2], abs(rangeP10[1]-rangeP10[2])), ncol=6, byrow=TRUE)
colnames(tabP) <- c('Periods', 'Mean', 'Std Deviation', 'Minimum', 'Maximum', 'Range')
tabP <- as.table(tabP)

print(tabP)
```

```
##     Periods      Mean Std Deviation   Minimum   Maximum     Range
## A  1.000000 10.638759      3.207258  5.818182 16.017857 10.199675
## B 10.000000 12.869879      3.898020  6.204545 17.511906 11.307360
```

Observation

The two experiments have same mean in Period 1. The standard deviation of experiment with punishment is more than the other experiment. The difference between maximum and minimum for the experiment with punishment is greater.

PART 2.3

p-value

A p-value is defined as a number that indicates how likely you are to obtain a value that is at least equal to or more than the actual observation if the null hypothesis is correct.

The null hypothesis is usually an hypothesis of "no difference" e.g. no difference between blood pressures in group A and group B.

Hypothesis Testing

The process of formulating a hypothesis about the data, calculating the p-value, and using it to assess whether what we observe is consistent with the hypothesis, is known as a hypothesis test.

- The smaller the p-value, the lower the probability that the differences we observe could have happened simply by chance, i.e. if the null hypothesis were true. The smaller the p-value, the stronger the evidence in favour of the alternative hypothesis.

Calculating the p-value and use it to assessing how likely it is that the differences we observe are due to chance.

```
t.test(x = period_data[[1]], y = period_dataP[[1]], paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  period_data[[1]] and period_dataP[[1]]
## t = -0.14996, df = 15, p-value = 0.8828
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -0.9195942  0.7987027
## sample estimates:
## mean difference
##     -0.06044576
```

We use the *t.test* function to calculate the p value, inputting two vectors containing data for period 1 of both experiments.

The paired factor is set to *TRUE* because our hypothesis is that the means for both groups are the same.

As the p-value is large *(0.88)* we can conclude that our hypothesis is correct, and it would not be unusual to observe the data that we did.

Calculating p-value for Period 10

```
t.test(x = period_data[[10]], y = period_dataP[[10]], paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  period_data[[10]] and period_dataP[[10]]
## t = -6.4806, df = 15, p-value = 1.037e-05
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -11.277166  -5.695054
## sample estimates:
## mean difference
##        -8.48611
```

- p-value for this case is **(0.00001)**

We assumed a null hypothesis. But the p-value indicated that the probability of observing a difference in sample means as large as or even larger than the one observed is 0.00001, which is very less, hence making it highly unusual and unlikely. Here our null hypothesis is not compatible with the data observed, so we reject the null hypothesis.