# A5 (Q5) - Ram Yoogesh Gopu(20867060)

```
knitr::opts_chunk$set(echo = TRUE,
                      warning = FALSE,
                      message = FALSE,
                      fig.align = "center",
                      fig.width = 6,
                      fig.height = 5,
                      out.height = "40%")
set.seed(12314159)
library(loon.data)
library(loon)
```

```
## Loading required package: tcltk
```

```
library(gridExtra)

codeDirectory <- "./code"
imageDirectory <- "./img"
dataDirectory <- "./data"
path_concat <- function(path1, ..., sep="/") paste(path1, ..., sep = sep)
source("graphicalTests.R")
source("numericalTests.R")
source("generateData.R")
```

**Full dataset is then read in as**

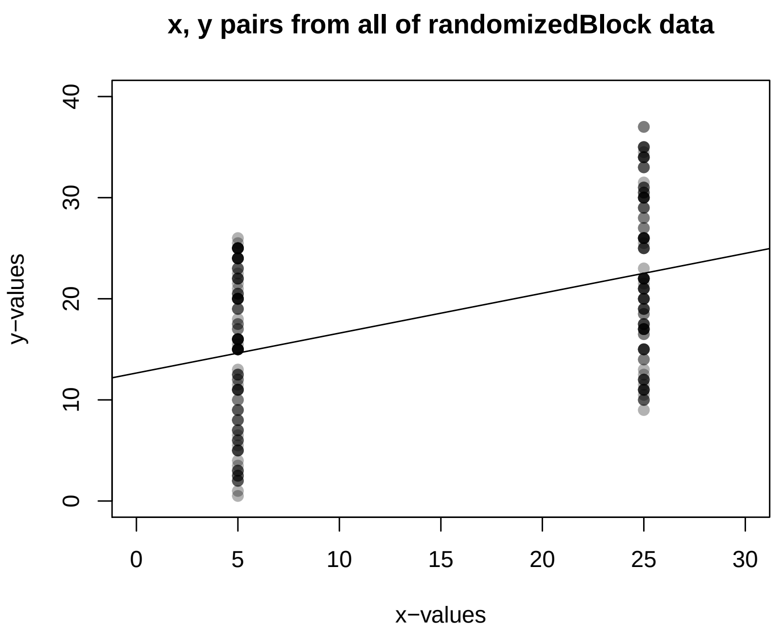
```
labData <- read.csv("labData.csv")
```

## (A)

```
randomizedBlock <- labData[labData$type == "randomizedBlock", ]
```

## (B)

```
plot(randomizedBlock$x, randomizedBlock$y, xlim = c(0, 30), ylim = c(0, 40), pch = 19, col = adjustcolo
Bmod <- lm(y~x, randomizedBlock)
abline(Bmod)
```




```r
print("Value of slope estimate is")
```

```
## [1] "Value of slope estimate is"
```

```r
print(Bmod$coefficients[2])
```

```
##         x
## 0.3944444
```

# (C)

## (i)

```r
rand1 <- randomizedBlock[randomizedBlock$rep == 1, ]
rand2 <- randomizedBlock[randomizedBlock$rep == 2, ]
```

## (ii)

```r
# defining the vectors and arrays
slopes1 <- c()
betas2 <- c()
first_t <- array()
second_t <- array()
```

```r
for (value in 1:18)
{
  first_t[value] <- lm(y~x, rand1[rand1$team == value, ])
  slopes1 <- c(slopes1, first_t[[value]][2])

  second_t[value] <- lm(y~x, rand2[rand2$team == value, ])
  betas2 <- c(betas2, second_t[[value]][2])
}
print("Average slope for Rep 1")
```

```
## [1] "Average slope for Rep 1"
```

```r
print(mean(slopes1))
```

```
## [1] 0.4305556
```

```r
print("Average slope for Rep 2")
```

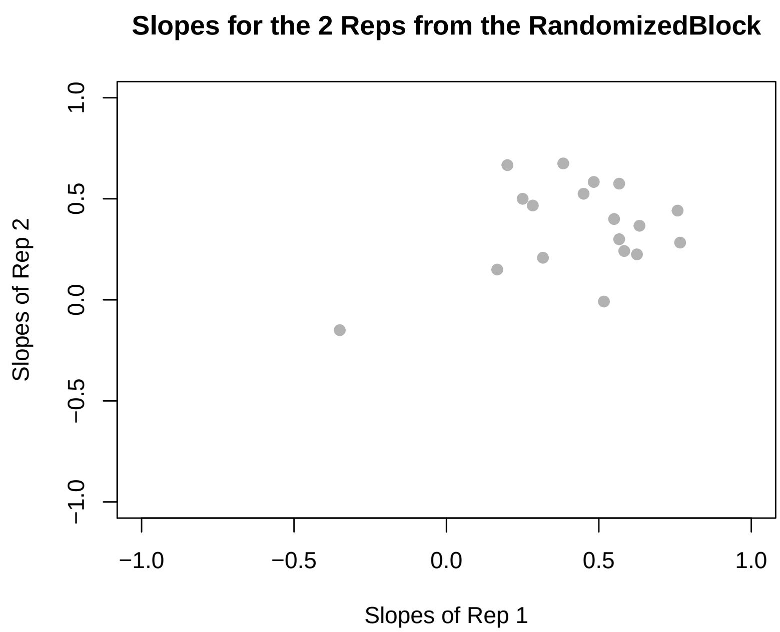
```
## [1] "Average slope for Rep 2"
```

```r
print(mean(betas2))
```

```
## [1] 0.3583333
```

### (ii)

```r
plot(slopes1, betas2, xlim = c(-1, 1), ylim = c(-1, 1), pch = 19, col = adjustcolor("black", 0.3), main
```



## (iii)

```r
sampobs <- data.frame(x = slopes1, y = betas2)
numericalTest(sampobs, generateFn = mixCoords, discrepancyFn = slopeDiscrepancy)
```

```
## [1] 0.2225
```

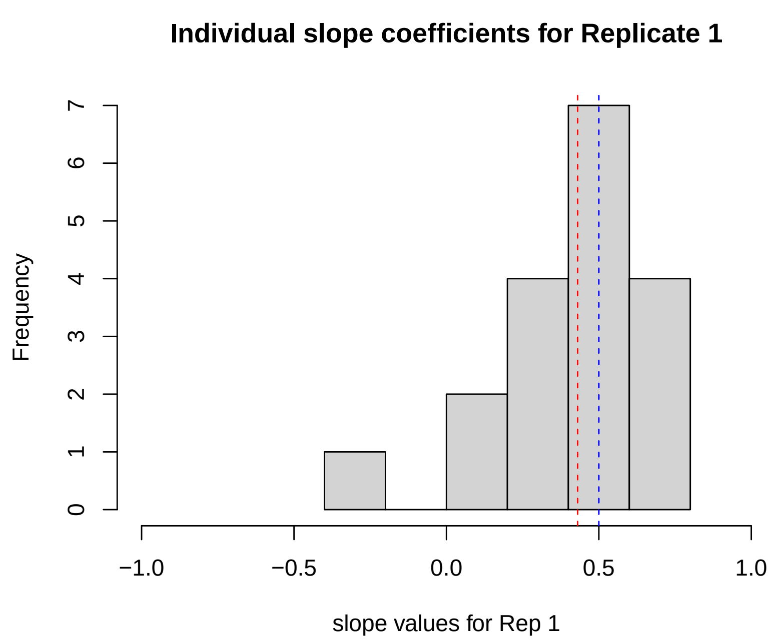
```r
numericalTest(sampobs, generateFn = mixCoords, discrepancyFn = correlationDiscrepancy)
```

```
## [1] 0.2225
```

**So from the outputs we can infer that the p-value is 0.2135 (slopeDicrepancy) (greater than 0.05) which is not strong enough against the null hypothesis. Also, the p-value is 0.22 (correlationDiscrepancy) (greater than 0.05) which is noy strong enough against the null hypothesis and the correlation cofficient must be a non-zero.**

## (iv)

```r
hist (slopes1, xlim = c(-1, 1), col = "lightgrey", main = "Individual slope coefficients for Replicate 
abline(v = mean(slopes1), col = "red", lty =2)
abline(v = 0.5, col = "blue", lty = 2)
```




```r
print("Average of slope estimates")
```

```
## [1] "Average of slope estimates"
```

```r
print(mean(slopes1))
```

```
## [1] 0.4305556
```

```r
print("Standard deviation of slope estimates")
```

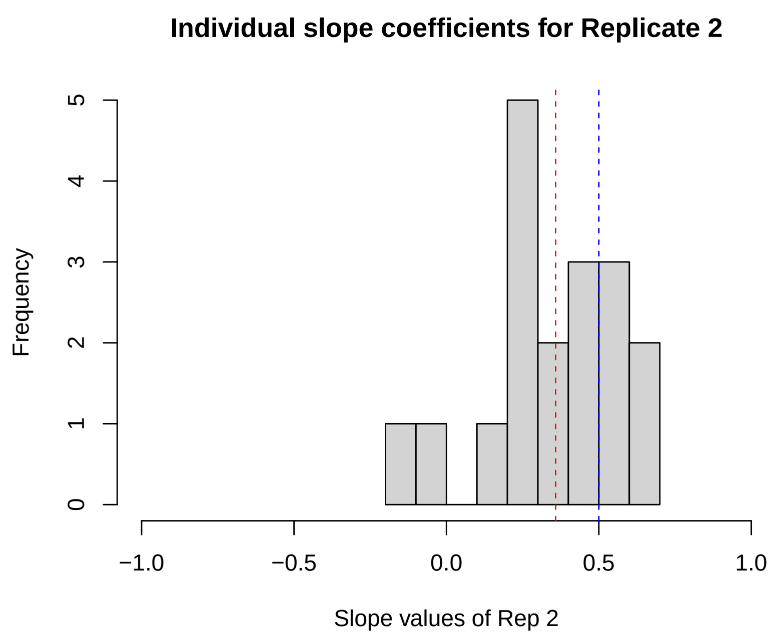
```
## [1] "Standard deviation of slope estimates"
```

```r
print(sd(slopes1))
```

```
## [1] 0.2641425
```

### (v)

```r
hist(betas2, xlim = c(-1, 1), col = "lightgrey", main = "Individual slope coefficients for Replicate 2"
abline(v = mean(betas2), col = "red", lty = 2)
abline(v = 0.5, col = "blue", lty = 2)
```




```r
print("Average of slope estimates")
```

```
## [1] "Average of slope estimates"
```

```r
print(mean(betas2))
```

```
## [1] 0.3583333
```

```r
print("Standard deviation of slope estimates")
```

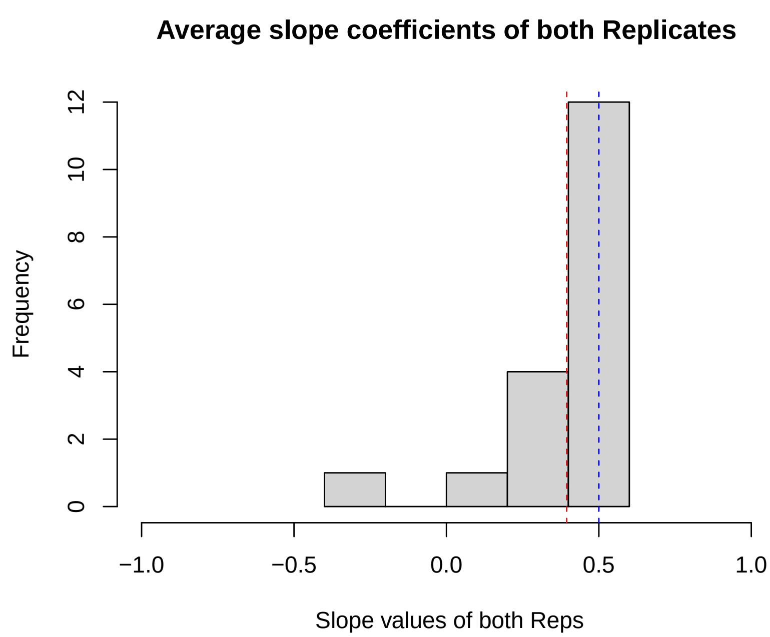
```
## [1] "Standard deviation of slope estimates"
```

```r
print(sd(betas2))
```

```
## [1] 0.2249728
```

# (Vi)

```r
avg_reps <- (slopes1 + betas2) / 2
hist(avg_reps, xlim = c(-1, 1), col = "lightgrey", main = "Average slope coefficients of both Replicate
abline(v = mean(avg_reps), col = "red", lty = 2)
abline(v = 0.5, col = "blue", lty = 2)
```




```r
print("Average of slope estimates for both reps")
```

```
## [1] "Average of slope estimates for both reps"
```

```r
print(mean(avg_reps))
```

```
## [1] 0.3944444
```

```r
print("Standard deviation of slope estimates for both reps")
```

```
## [1] "Standard deviation of slope estimates for both reps"
```

```r
print(sd(avg_reps))
```

```
## [1] 0.1981275
```

# (E)

**From the study we can infer that the average slope estimates are closer to the true slope values. Hence, I would conclude that the quality of team slope estimates in randomizedBlock study is good and better than randomized and observational design.**

# (F)

**From the output, we know that the average of replicates is 0.39444 which is quite close to the true value 0.5. This gives a better understanding of slope estimates instead of viewing a single repetition on a individual basis. Hence, this is more preferrable.**

# (G)

**Since Z is a lurking variable for the above problem, it is clear that it has a fixed value which fixes (hyperplane). This has an effect on the values of y. Also, there would be different y values when we change Z. Furthermore, the hyperplane and the height of the markers imposes a constraint on y.**