Code Examples for Experiment Design in Computer Science, Lecture II

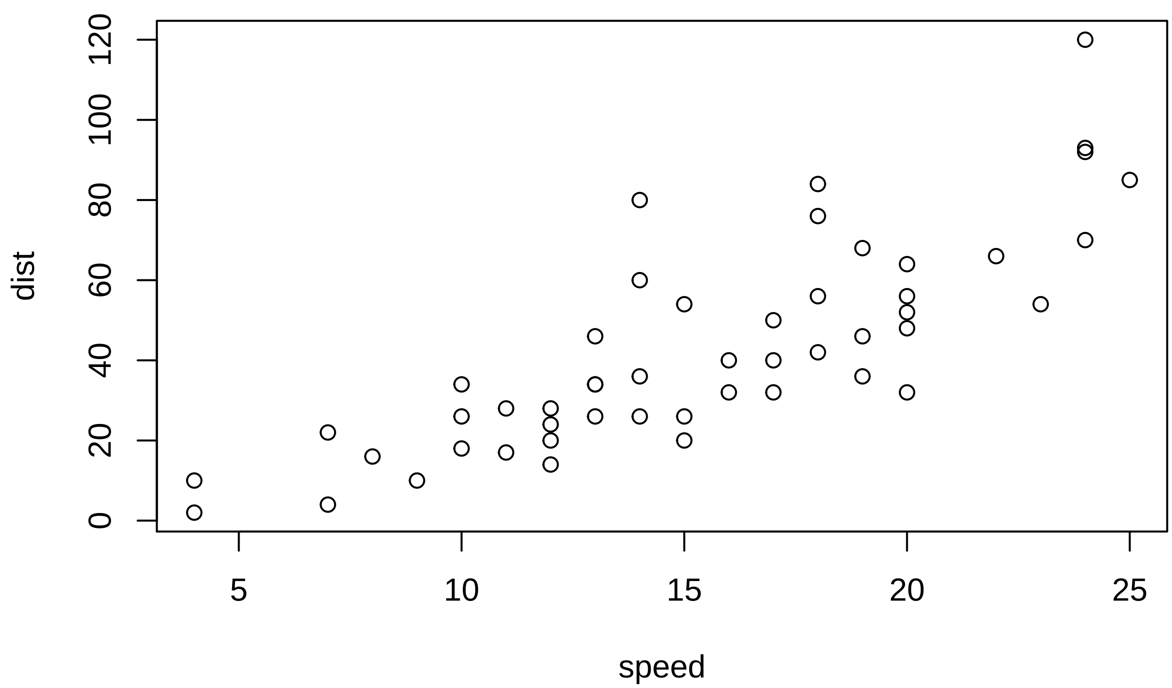
Claus Aranha

R Markdown Introduction

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
plot(cars)
```



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

Lecture II, Topic I: Point and Interval Indicators

Example I: Coaxial cable factory

In this example, we assume a hypotethical factory that produces coaxial cables. The resistance of the cables produced follow a gaussian distribution, with mean 50 and standard deviation 2.

First, let's generate a sample of 25 cables from this process:

```
# Generating the data.
set.seed(42)                    # Set PRNG seed (for reproducibility)
x<-rnorm(n=25, mean=50, sd=2)   # Draw 25 samples from N(mu=50,sigma=2)

x                               # Lists the resistances of each cable in the sample
```

```
##  [1] 52.74192 48.87060 50.72626 51.26573 50.80854 49.78775 53.02304 49.81068
##  [9] 54.03685 49.87457 52.60974 54.57329 47.22228 49.44242 49.73336 51.27190
## [17] 49.43149 44.68709 45.11907 52.64023 49.38672 46.43738 49.65617 52.42935
## [25] 53.79039
```

We are interested in estimating the mean of the process from the sample data. We can do this by using the **sample mean** statistic, which, as we saw in class, is an unbiased estimator of the population mean.

```
sample_mean <- sum(x) / length(x) # Sample mean: sum of obs / sample size
sample_mean
```

```
## [1] 50.37507
```

Of course, we can also direclty use the native **mean** function to obtain the same value: 50.3750723

We can now calculate an estimated error for the estimated mean:

```
sample_mean_error <- sd(x)/sqrt(length(x)) # sample mean error: slide 25
sample_mean_error
```

```
## [1] 0.5225459
```

The larger the sample, the smaller is the estimated error for the estimated mean:

```
csample_10 <- rnorm(n=10, mean=50, sd=2)
csample_25 <- rnorm(n=25, mean=50, sd=2)
csample_50 <- rnorm(n=50, mean=50, sd=2)

sme <- function(x) {
  sd(x) / sqrt(length(x))
}

print(c(length(csample_10), mean(csample_10), sme(csample_10)))
```

```
## [1] 10.0000000 49.8921241  0.5314076
```

```
print(c(length(csample_25), mean(csample_25), sme(csample_25)))
```

```
## [1] 25.0000000 49.5401060  0.4374049
```

```
print(c(length(csample_50), mean(csample_50), sme(csample_50)))
```

```
## [1] 50.0000000 50.2857444  0.2419586
```

Example II: Student status (from Campelo's Analysis and Design of Experiments Course)

As a second example, let's consider a survey of students on their height and weight. We can easily load data that is stored as a **csv file** into an R data fram (data frames and lists are R's main data structures).

```r
students <- read.csv("student_data.csv")
students
```

```
##    ID Course Gender Height.m Weight.kg
## 1   1  PPGEE      F     1.57      45.5
## 2   2  PPGEE      F     1.62      53.0
## 3   3  PPGEE      F     1.70      57.0
## 4   4  PPGEE      F     1.62      59.0
## 5   5  PPGEE      F     1.67      63.0
## 6   6  PPGEE      F     1.76      78.0
## 7   7  PPGEE      F     1.64      51.0
## 8   8  PPGEE      M     1.79      80.0
## 9   9  PPGEE      M     1.58      58.0
## 10 10  PPGEE      M     1.74      85.0
## 11 11  PPGEE      M     1.75     115.0
## 12 12  PPGEE      M     1.78      71.0
## 13 13  PPGEE      M     1.71      71.0
## 14 14  PPGEE      M     1.78      86.0
## 15 15  PPGEE      M     1.81      80.0
## 16 16  PPGEE      M     1.79      72.0
## 17 17  PPGEE      M     1.70      62.5
## 18 18  PPGEE      M     1.82     100.0
## 19 19  PPGEE      M     1.72      52.0
## 20 20  PPGEE      M     1.83      84.0
## 21 21  PPGEE      M     1.70      74.0
## 22 22  PPGEE      M     1.83      92.0
## 23 23  PPGEE      M     1.73      60.0
## 24 24  PPGEE      M     1.69      57.5
## 25 25  PPGEE      M     1.89      87.0
## 26 26  PPGEE      M     1.63      81.0
## 27 27  PPGEE      M     1.81      78.0
## 28 28  PPGEE      M     1.73      68.0
## 29 29 ENGSIS      F     1.67      53.0
## 30 30 ENGSIS      F     1.61      55.0
## 31 31 ENGSIS      F     1.56      43.0
## 32 32 ENGSIS      M     1.89      80.0
## 33 33 ENGSIS      M     1.80      97.0
## 34 34 ENGSIS      M     1.77      78.0
## 35 35 ENGSIS      M     1.67      65.0
## 36 36 ENGSIS      M     1.81     110.0
## 37 37 ENGSIS      M     1.86     110.0
## 38 38 ENGSIS      M     1.70      63.0
## 39 39 ENGSIS      M     1.79      64.0
## 40 40 ENGSIS      M     1.78      84.0
## 41 41 ENGSIS      M     1.75      57.5
## 42 42 ENGSIS      M     1.70      64.0
## 43 43 ENGSIS      M     1.82      80.0
## 44 44 ENGSIS      M     1.83      70.0
## 45 45 ENGSIS      M     1.84      73.0
```

```
## 46 46 ENGSIS      M     1.69      58.0
## 47 47 ENGSIS      M     1.75      70.0
## 48 48 ENGSIS      M     1.76      74.8
## 49 49 ENGSIS      M     1.83      85.0
```

```r
mean(students$Height.m)    # Student mean height
```

```
## [1] 1.740204
```

```r
mean(students$Weight.kg)    # Student mean weight
```

```
## [1] 72.54694
```

We might be interested in calculating the BMI of the students who answered the survey:

```r
bmi = students$Weight.kg / (students$Height.m ** 2)
students["BMI"] <- bmi
students
```

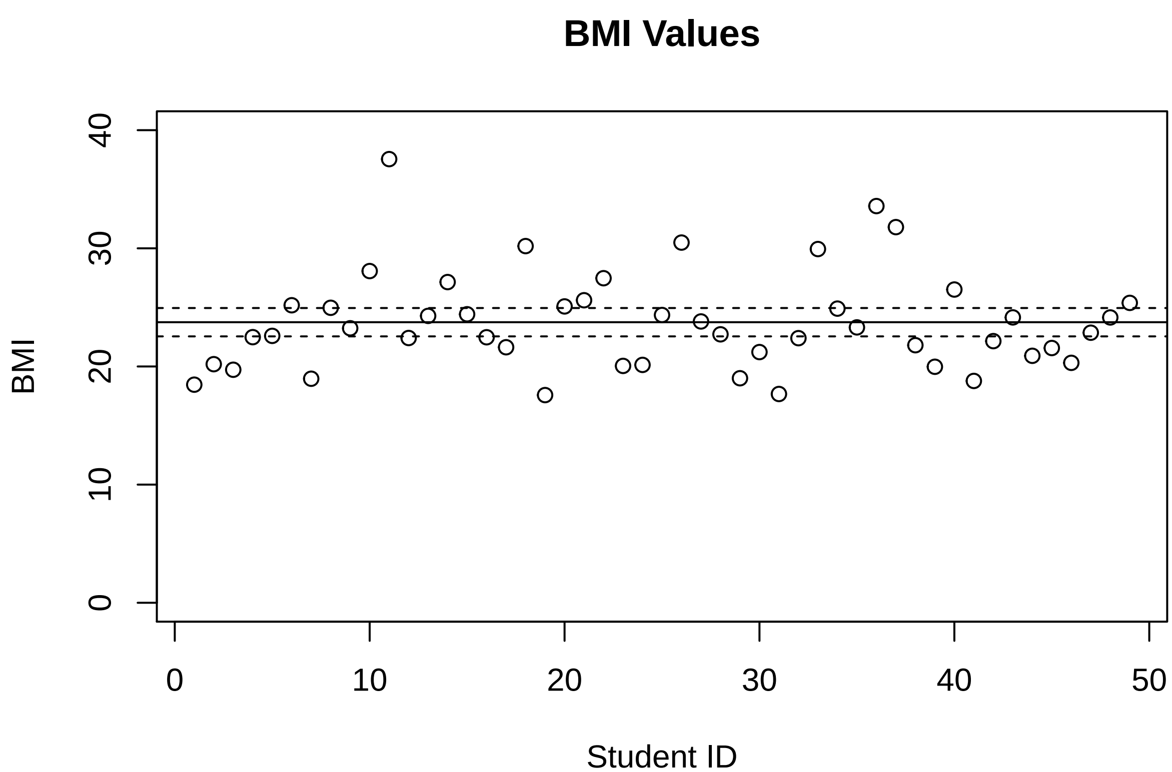
```
##    ID Course Gender Height.m Weight.kg      BMI
## 1   1  PPGEE      F     1.57      45.5 18.45917
## 2   2  PPGEE      F     1.62      53.0 20.19509
## 3   3  PPGEE      F     1.70      57.0 19.72318
## 4   4  PPGEE      F     1.62      59.0 22.48133
## 5   5  PPGEE      F     1.67      63.0 22.58955
## 6   6  PPGEE      F     1.76      78.0 25.18079
## 7   7  PPGEE      F     1.64      51.0 18.96193
## 8   8  PPGEE      M     1.79      80.0 24.96801
## 9   9  PPGEE      M     1.58      58.0 23.23346
## 10 10  PPGEE      M     1.74      85.0 28.07504
## 11 11  PPGEE      M     1.75     115.0 37.55102
## 12 12  PPGEE      M     1.78      71.0 22.40879
## 13 13  PPGEE      M     1.71      71.0 24.28098
## 14 14  PPGEE      M     1.78      86.0 27.14304
## 15 15  PPGEE      M     1.81      80.0 24.41928
## 16 16  PPGEE      M     1.79      72.0 22.47121
## 17 17  PPGEE      M     1.70      62.5 21.62630
## 18 18  PPGEE      M     1.82     100.0 30.18959
## 19 19  PPGEE      M     1.72      52.0 17.57707
## 20 20  PPGEE      M     1.83      84.0 25.08286
## 21 21  PPGEE      M     1.70      74.0 25.60554
## 22 22  PPGEE      M     1.83      92.0 27.47171
## 23 23  PPGEE      M     1.73      60.0 20.04745
## 24 24  PPGEE      M     1.69      57.5 20.13235
## 25 25  PPGEE      M     1.89      87.0 24.35542
## 26 26  PPGEE      M     1.63      81.0 30.48666
## 27 27  PPGEE      M     1.81      78.0 23.80880
## 28 28  PPGEE      M     1.73      68.0 22.72044
## 29 29 ENGSIS      F     1.67      53.0 19.00391
## 30 30 ENGSIS      F     1.61      55.0 21.21832
## 31 31 ENGSIS      F     1.56      43.0 17.66930
## 32 32 ENGSIS      M     1.89      80.0 22.39579
## 33 33 ENGSIS      M     1.80      97.0 29.93827
## 34 34 ENGSIS      M     1.77      78.0 24.89706
## 35 35 ENGSIS      M     1.67      65.0 23.30668
## 36 36 ENGSIS      M     1.81     110.0 33.57651
## 37 37 ENGSIS      M     1.86     110.0 31.79558
```

```
## 38 38 ENGSIS      M      1.70      63.0 21.79931
## 39 39 ENGSIS      M      1.79      64.0 19.97441
## 40 40 ENGSIS      M      1.78      84.0 26.51180
## 41 41 ENGSIS      M      1.75      57.5 18.77551
## 42 42 ENGSIS      M      1.70      64.0 22.14533
## 43 43 ENGSIS      M      1.82      80.0 24.15167
## 44 44 ENGSIS      M      1.83      70.0 20.90239
## 45 45 ENGSIS      M      1.84      73.0 21.56191
## 46 46 ENGSIS      M      1.69      58.0 20.30741
## 47 47 ENGSIS      M      1.75      70.0 22.85714
## 48 48 ENGSIS      M      1.76      74.8 24.14773
## 49 49 ENGSIS      M      1.83      85.0 25.38147
```

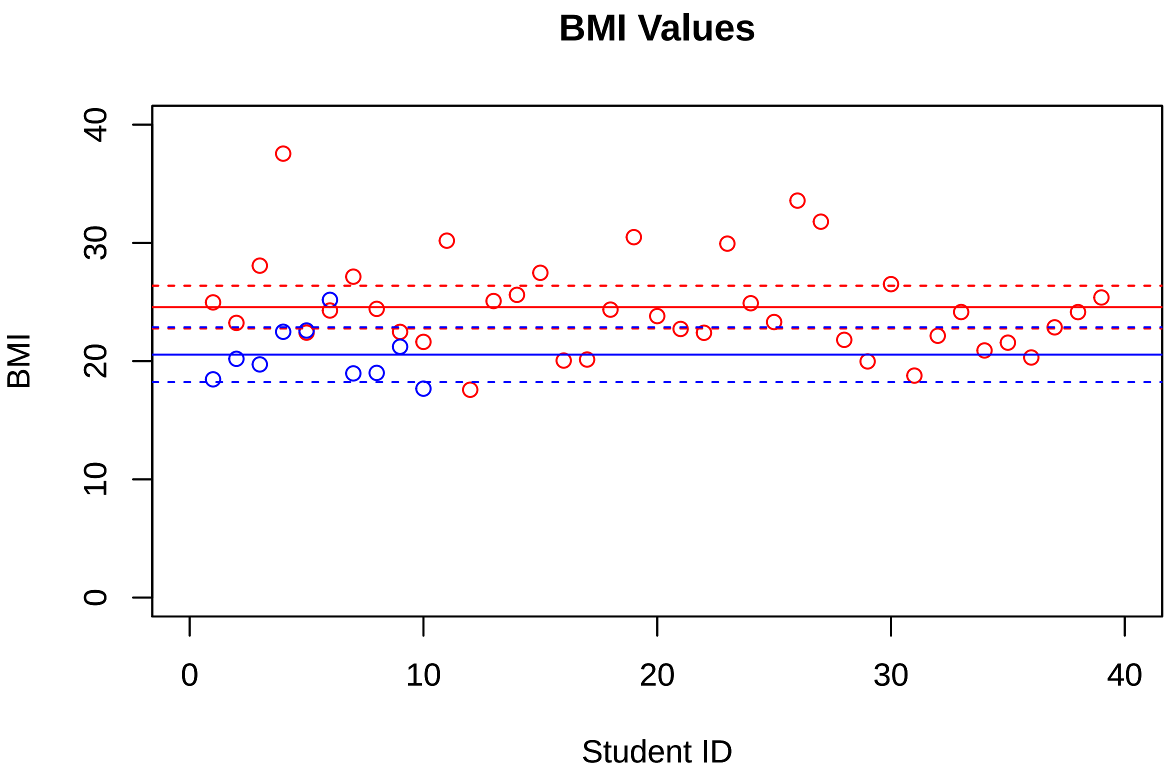
The mean BMI of the students is 23.7461942 with an estimated error of 0.5970859.

However, point indicators alone give us a limited vision of the data that we want to study. When analysing data, always accompany point indicators with statistical intervals and **figures**. Let's plot the mean BMI of the students, as well as the 95% confidence interval for this estimator:

```
values <- students$BMI  # shorter name for variable
ci_alpha <- 0.05        # Alpha value for confidence interval. Confidence = 1 - alpha.
# calculating low and high bounds for ci
ci_low <- mean(values) + qt(ci_alpha/2, length(values))*(sd(values)/sqrt(length(values)))
ci_hi <- mean(values) + qt(1 - ci_alpha/2, length(values))*(sd(values)/sqrt(length(values)))

# plot the values and the confidence interval of the mean
plot(students$BMI, ylim = c(0,40), xlab = "Student ID", ylab = "BMI")
title("BMI Values")
abline(h = mean(students$BMI))
abline(h = ci_low, lty=2)
abline(h = ci_hi, lty=2)
```

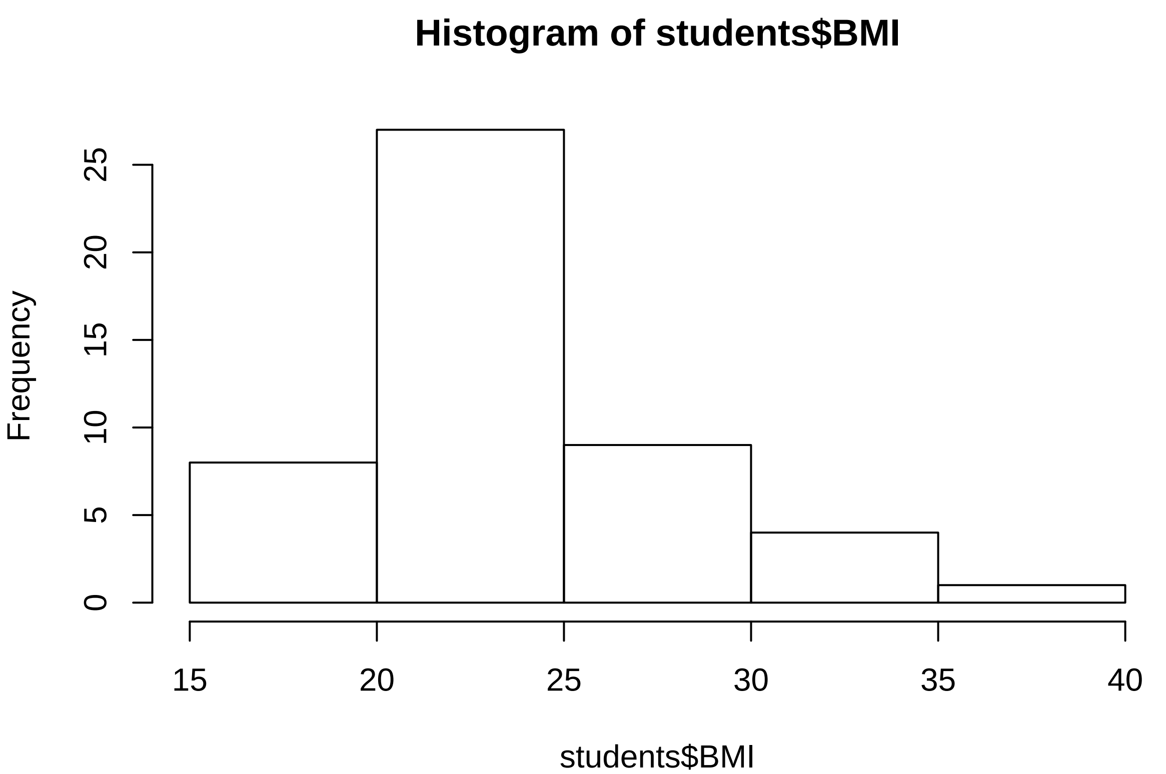


Let's investigate the difference in BMI between male students and female students:

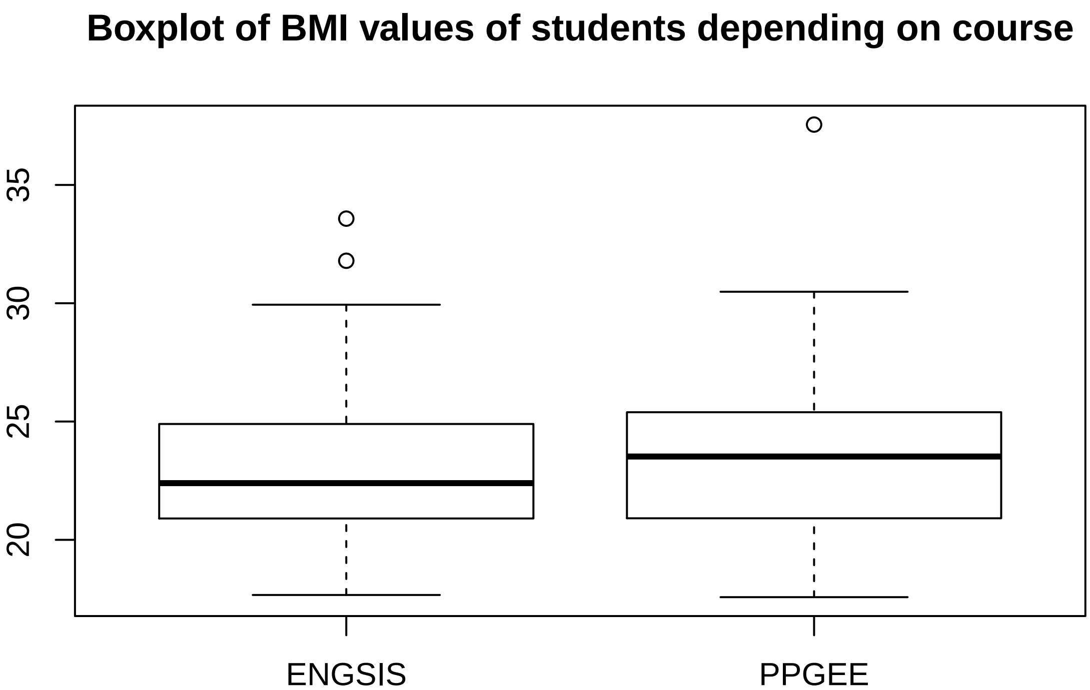


Let's see two other ways to observe the values of a sample.

The histogram is also a good way to visualise the distribution of a random variable within a sample:



Boxplots give a good idea of upper and lower limits of a sample's value:

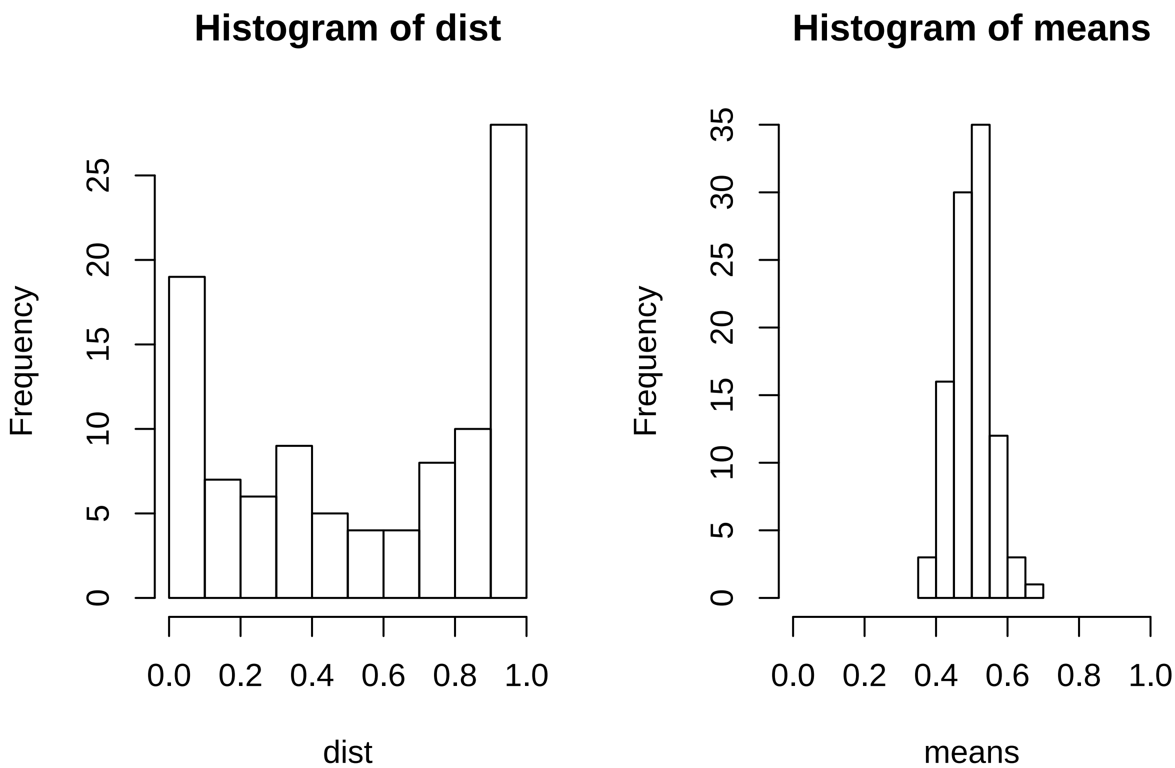


Topic II: Central Limit Theorem

The Central Limit Theorem (CLT) states that for most distributions, the distribution of sample means tends to follow a normal distributio under certain conditions. Let's observe this effect. Modify the code below to test different base distributions:

```
sample_number <-100  # Leave this fixed
sample_size <- 50    # Change this

### distributions: uncomment one of these:
means = replicate(sample_number,mean(rbeta(sample_size, 0.5, 0.5)))
dist = rbeta(sample_number, 0.5, 0.5)
name = "Beta Distribution"

# means = replicate(sample_number,mean(rchisq(sample_size, df=2)))
# dist = rchisq(sample_number, df=2)
# name = "Chi Squared Distribution"

# means = replicate(sample_number,mean(rf(sample_size, df1=5, df2=10)))
# dist = rf(sample_number, df1=5, df2=10)
# name = "F distribution"

# means = replicate(sample_number,mean(rnorm(sample_size)))
# dist = rnorm(sample_number)
# name = "Normal Distribution"

lims = c(min(c(means,dist)),max(c(means,dist)))

par(mfcol=(c(1,2)))
hist(dist, xlim = lims)
```

```r
hist(means, xlim = lims)
```




```r
# TODO: make this prettier
```