Assignment 1

Tim But, Mehmet Bedirhan Gursoy, Vincentas Ryliskis, group 035

6 November 2023

Task 1

a)

Suppose the following:
- P(D) is the probability of a positive cancer diagnosis
- P(C) is the probability of an isndividual having cancer

Then, given the information given in exercise 1.3:

$$P(D|C) = 0.95$$

$$P(D|C') = 0.05$$

$$P(C) = 0.004$$

To calculate the probability of a random person being given a positive cancer diagnosis, the following must be calculated:

$$P(D) = P(D|C) * P(C) + P(D|C') * P(C')$$

$$= 0.95 * 0.04 + 0.05 * 0.996$$

$$= 0.0536$$

This differs from probability we are asked to calculate in the Exercise 1.3, as there we are asked to calculate the probability of someone having cancer given a cancer diagnosis, or: P(C|D). These probabilities refer to different events, as the latter already assumes P(D) as being true.

b)

As mentioned above, we need to calculate P(C|D). We can calculate this value using Bayes' Theorem

$$P(C|D) = \frac{P(D|C) * P(C)}{P(D)} = \frac{0.95 * 0.004}{0.0536} \approx 0.071$$

c)

The two events that a person has cancer and that the test is positive are dependent. This is shown by the fact that probabilities P(C|D) and P(C) are different, as shown below (based on calculations done in parts a) and b)):
- P(C|D) = 0.071
- p(C) = 0.004
The fact that P(C|D) is larger than P(C) shows that a cancer diagnosis leads to an increased risk in cancer, when compared to the population average.

Task 2

a)

b)

c)

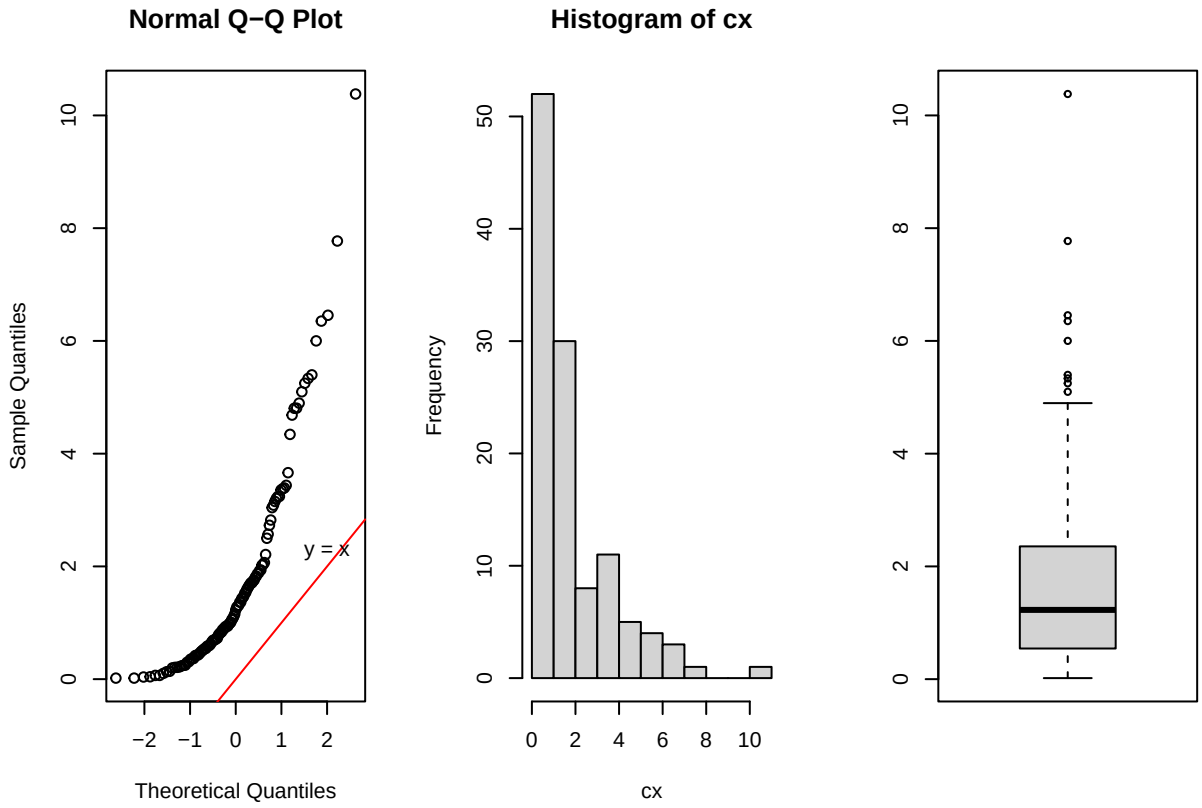
d)

e)

Task 3

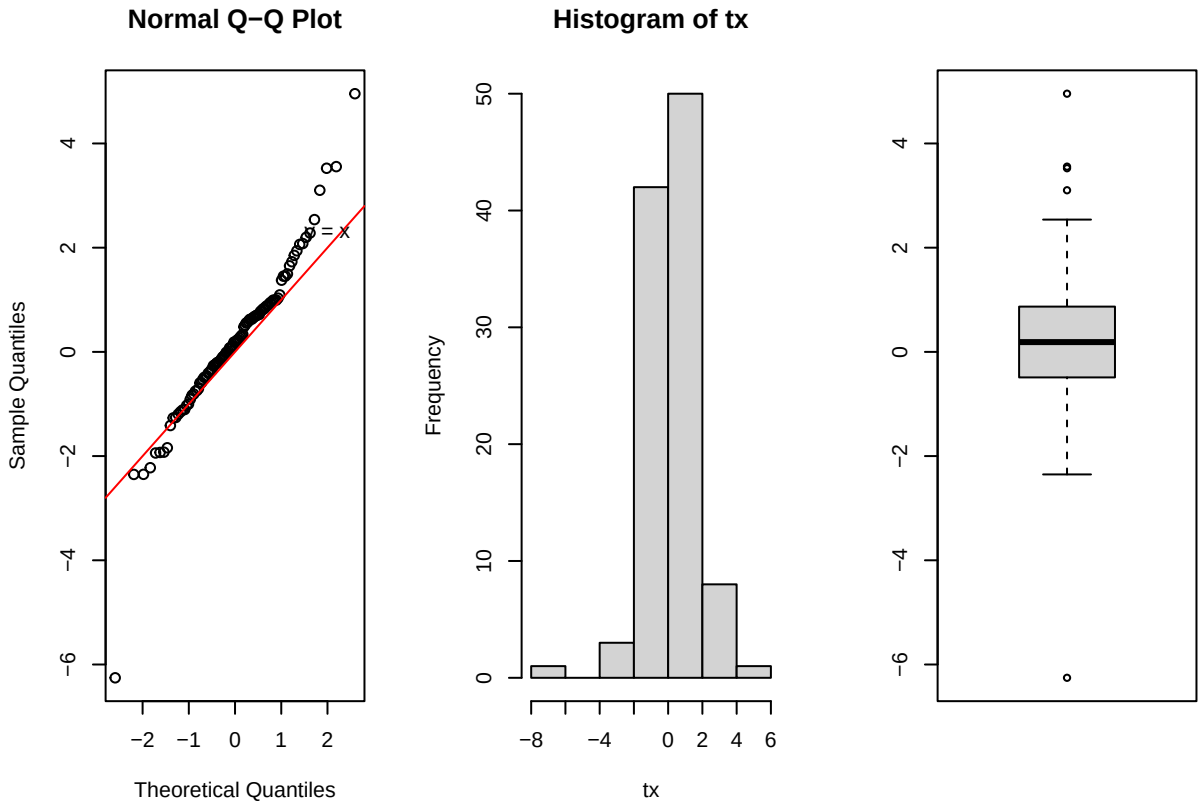
a)

```
# Setting a seed
set.seed(123)
par(mfrow = c(1,3))


cn = 115
cdf=2
ci = sequence(cn)
cx = rchisq(ci, df=cdf)

qqnorm(cx)
abline(0, 1, col = 'red')
text(2, 2.25, "y = x")

hist(cx)
boxplot(cx)
```



(i)

```r
par(mfrow = c(1,3))

tn = 105
tdf=4
ti = sequence(tn)
tx = rt(ti, df=tdf)

qqnorm(tx)
abline(0, 1, col = 'red')
text(2, 2.25, "y = x")

hist(tx)
boxplot(tx)
```



(ii)

b) #### (i)

```r
smallerThan3 = pnorm(3)
print(smallerThan3)
```

```
## [1] 0.9986501
```

```r
biggerthanM0.5 = 1-pnorm(-0.5)
print(biggerthanM0.5)
```

```
## [1] 0.6914625
```

```r
betweenM1and2 = pnorm(2) - pnorm(-1)
print(betweenM1and2)
```

```
## [1] 0.8185946
```

```r
variance = 4
sd = sqrt(variance)

smallerThan3 = pnorm(3, mean = 3, sd = sd)
print(smallerThan3)
```

(ii)

```
## [1] 0.5
```

```r
biggerthanM0.5 = 1-pnorm(-0.5, mean = 3, sd = sd)
print(biggerthanM0.5)
```

```
## [1] 0.9599408
```

```r
betweenM1and2 = pnorm(2, mean = 3, sd = sd) - pnorm(-1, mean = 3, sd = sd)
print(betweenM1and2)
```

```
## [1] 0.2857874
```

```r
NFpercSmaller = qnorm(0.95, mean = 3, sd = sd)
print(NFpercSmaller)
```

```
## [1] 6.289707
```

```r
mean = -1
sd = 5

samples = rnorm(1000)

corrected_samples = (samples * sd) +mean

sample_mean = mean(corrected_samples)
print(sample_mean)
```

(iii)

```
## [1] -0.9403835
```

```r
sample_sd = sd(corrected_samples)
print(sample_sd)
```

```
## [1] 5.005643
```

```r
verification_function <- function(min = -Inf, max = Inf){

  sample_100 = rnorm(100)
  sample_100k = rnorm(100000)

  prob_sample_100 = mean(sample_100 > min & sample_100 < max)
  prob_sample_100k = mean(sample_100k > min & sample_100k < max)

  cat("Probability (Sample 100):", prob_sample_100, "\n")
  cat("Probability (Sample 100k):", prob_sample_100k, "\n")
}

verification_function(max = 3)
```

(iv)

```
## Probability (Sample 100): 1
## Probability (Sample 100k): 0.99868
```

```r
verification_function(min = -0.5)
```

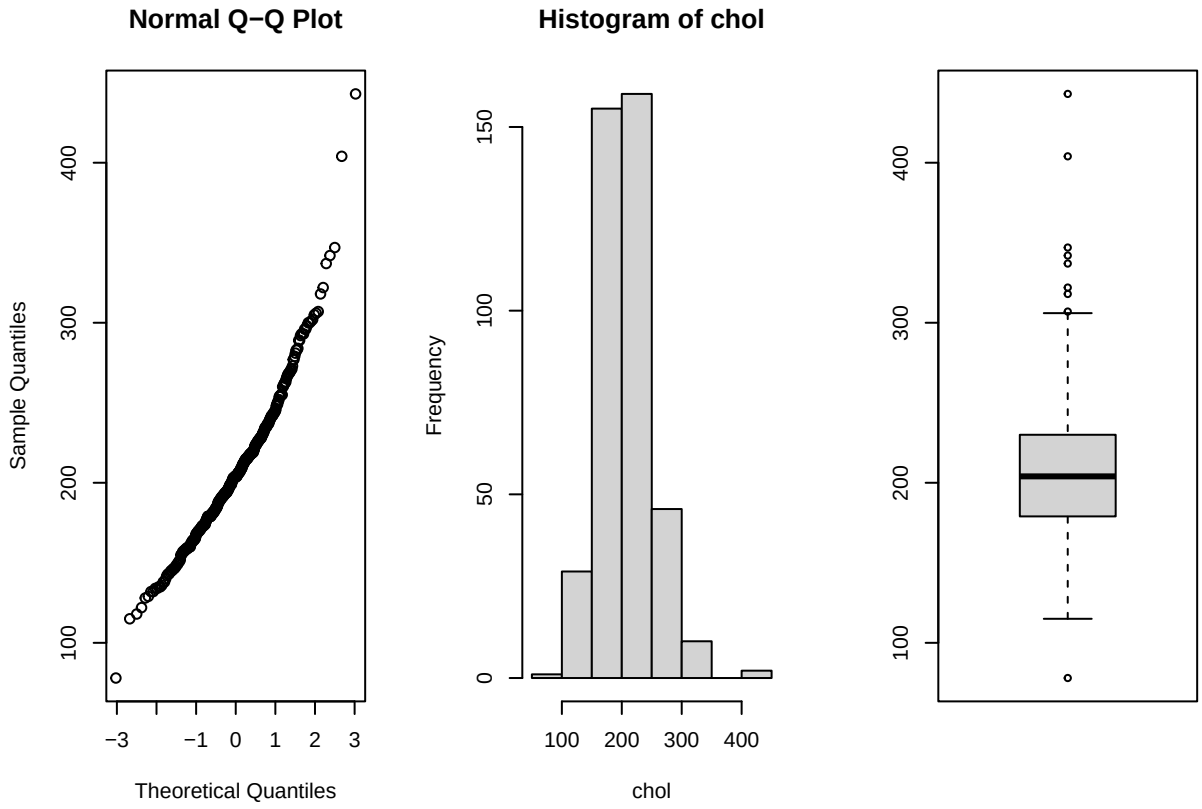
```
## Probability (Sample 100): 0.66
## Probability (Sample 100k): 0.69379
```

```r
verification_function(min = -1, max = 2)
```

```
## Probability (Sample 100): 0.8
## Probability (Sample 100k): 0.81713
```

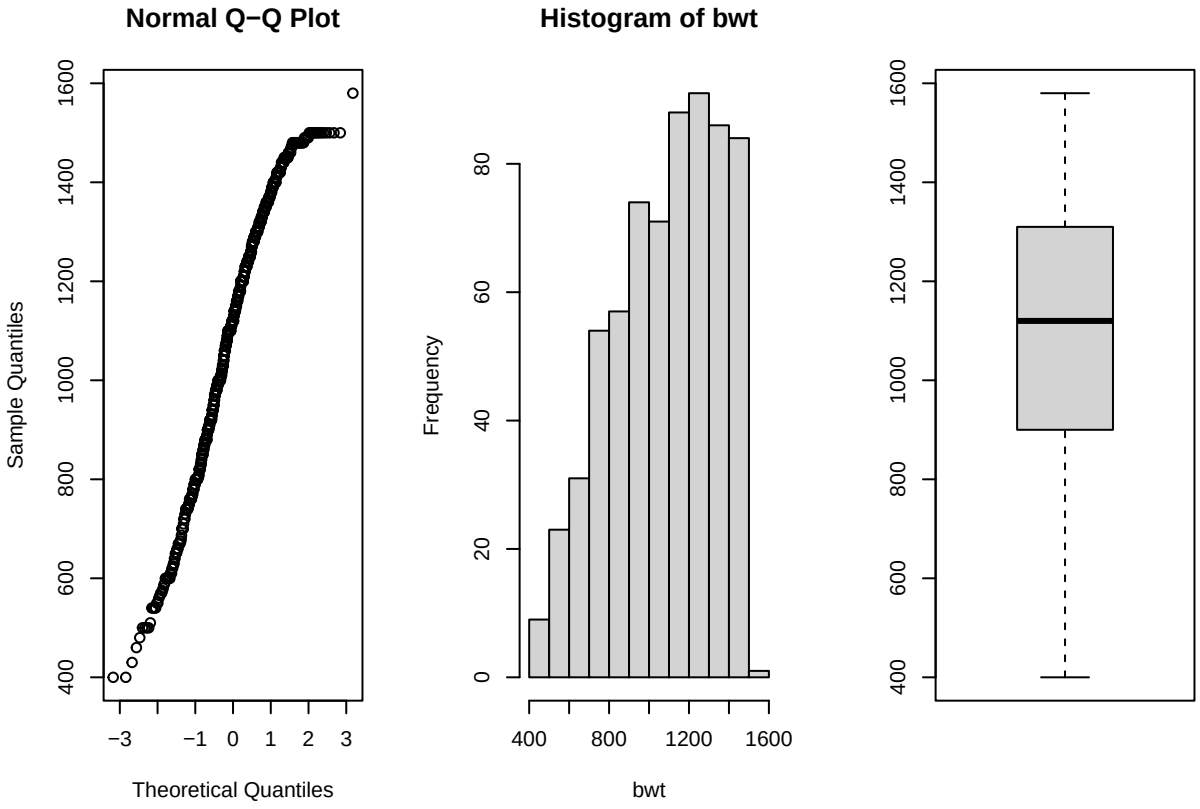
c)

```r
par(mfrow = c(1,3))

diabetes <- read.csv("diabetes.csv")
chol = diabetes$chol

qqnorm(chol)

hist(chol)
boxplot(chol)
```



(i)

```r
par(mfrow = c(1,3))

vlbw <- read.csv("vlbw.csv")
bwt = vlbw$bwt

qqnorm(bwt)
hist(bwt)
boxplot(bwt)
```

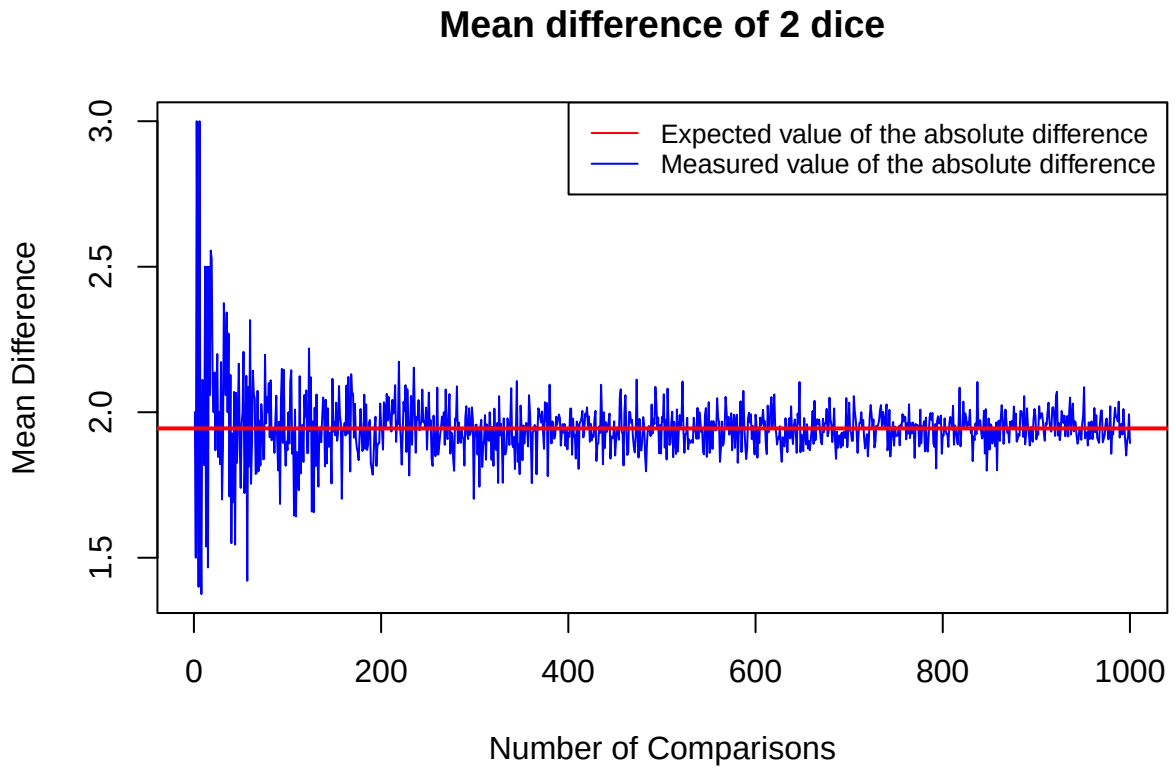


(ii)

Task 4

a)

```
source("function02.txt")
n = 1000
meandiff = numeric(n)


for (i in 1:1000){
  meandiff[i] = mean(diffdice(i))

}

plot(meandiff, type = 'l', col = "blue", xlab = "Number of Comparisons", ylab = "Mean Differenc
   main = "Mean difference of 2 dice")
abline(h = 1.9444, col = "red", lwd = 2)

legend("topright", legend=c("Expected value of the absolute difference", "Measured value of the
       col=c("red", "blue"), lty=1, cex=0.8)
```



b)

```r
source("function02.txt")

n = 100000
Diffs100k = diffdice(n)
expectation = mean(Diffs100k)
print(expectation)
```

```
## [1] 1.93781
```

```r
expectation = 0

for (i in 0 : 6){
  probability_of_i = length(which(Diffs100k == i))/n

  expectation = expectation + (probability_of_i*i)
}
expectation = expectation

print(expectation)
```

```
## [1] 1.93781
```

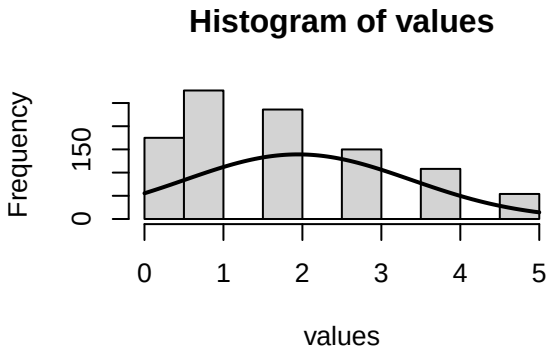
```r
probabiliy_of_3 = length(which(Diffs100k == 3))/n
print(probabiliy_of_3)
```

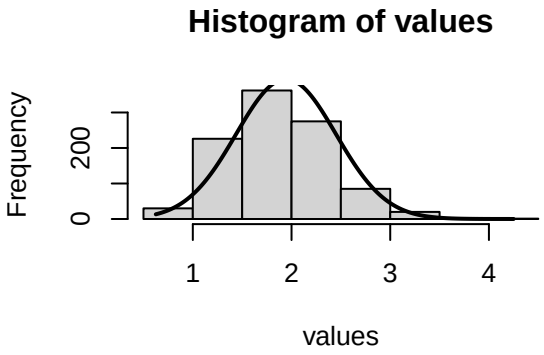
```
## [1] 0.16797
```

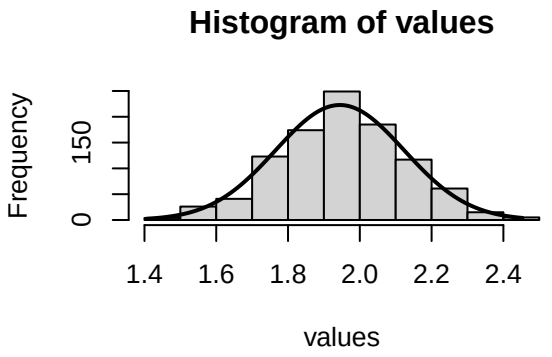
c)

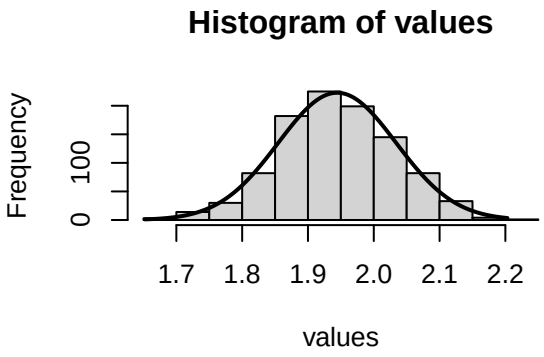
```r
source("function02.txt")
par(mfrow = c(2,2))
CLTGenerator = function(n = 1, sample_means = 1000){
  values = numeric(sample_means)
  for (i in 1:sample_means){
    values[i] = mean(diffdice(n))
  }
  h <- hist(values)
  xfit <- seq(min(values), max(values), length = 1000)
  yfit <- dnorm(xfit, mean = 1.9444, sd = sqrt(((1.4326)^2)/n))
  yfit <- yfit * diff(h$mids[1:2]) * length(values)
  lines(xfit, yfit, col = "black", lwd = 2)

}
```

```
CLTGenerator(n = 1)
CLTGenerator(n = 8)
CLTGenerator(n = 64)
CLTGenerator(n = 256)
```









d)