

# Assignment 0: Practicing R

## Exercise 1

Practicing working with normal distributions.

### Part (a):

Generate two samples of sizes 100 and 100000 from a standard normal distribution. Make histograms and QQ-plots, compute the means and standard deviations of the both samples. Explain your findings

### Part (b) and (c):

For a standard normal distribution, compute the following 3 probabilities: that an arbitrary outcome is smaller than 2, that it is bigger than -0.5 and that it is between -1 and 2.

Can you verify the outcomes of b) using only the data from a)?

```
part1 = function(mean=0, sd=1) {  
  par(mfrow=c(2,2))  
  s1 = rnorm(100, mean=mean, sd=sd)  
  s2 = rnorm(100000, mean=mean, sd=sd)  
  
  # make histograms  
  hist(s1, freq=F)  
  hist(s2, freq=F)  
  
  # make qqplots to verify samples appear to come from norm. dist.  
  #dist = function(x, mean, sd) {  
  #  return rnorm(x, mean=mean, sd=sd)  
  #}  
  
  # making qplot against std. norm. dist.  
  qqnorm(s1, main="s1: Normal Q-Q Plot")  
  abline(0, 1, col = 'red')  
  qqnorm(s2, main="s2: Normal Q-Q Plot")  
  abline(0, 1, col = 'red')  
  
  prob1 = pnorm(-0.5)  
  print(sprintf("prob (x < -0.5) = %.4f", prob1))  
  
  prob2 = pnorm(2) - pnorm(-1)  
  print(sprintf("prob (-1 < x < 2) = %.4f", prob2))  
  
  # note that qnorm does the opposite:  
  # qnorm(pnorm(-0.5)) = -0.5  
  
  # estimate these probabilities using data from samples:
```

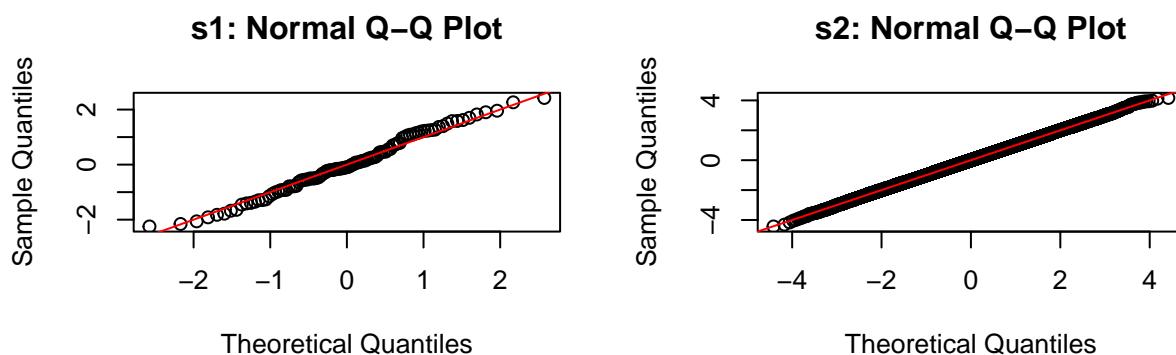
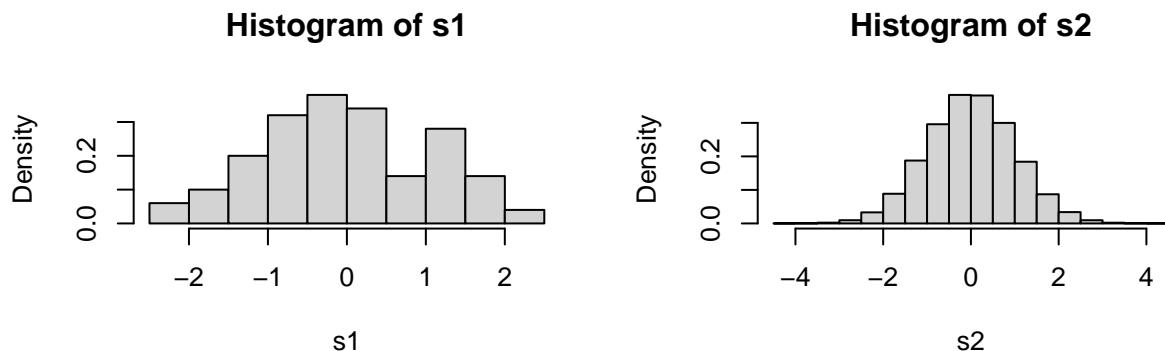
```

est1a = sum(s1 < -0.5) / length(s1)
est1b = sum(s2 < -0.5) / length(s2)
print(sprintf("estimate 1a: %.4f, estimate1b: %.4f", est1a, est1b))

est2a = (sum(s1 < 2) - sum(s1 < -1)) / length(s1)
est2b = (sum(s2 < 2) - sum(s2 < -1)) / length(s2)
print(sprintf("estimate 2a: %.4f, estimate2b: %.4f", est2a, est2b))
}

part1()

```



```

## [1] "prob (x < -0.5) = 0.3085"
## [1] "prob (-1 < x < 2) = 0.8186"
## [1] "estimate 1a: 0.3400, estimate1b: 0.3088"
## [1] "estimate 2a: 0.8000, estimate2b: 0.8162"
# generate two samples from std. norm. dist.

```

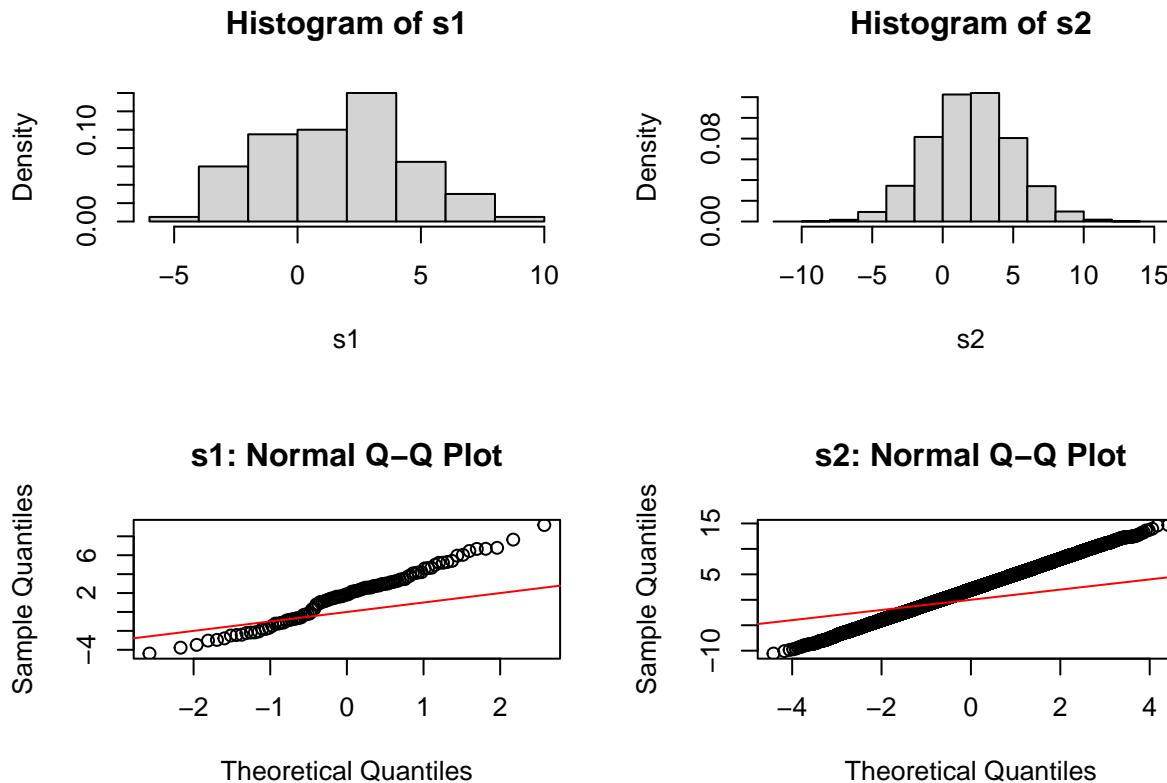
The findings suggest that as expected, both samples appear to come from the standard normal distribution. The bell shaped curves of the histograms, as well as the straight line in the QQ plots provide evidence for this.

#### Part (d) and (e):

Repeat a) and b) for a normal distribution with mean=3 and sd=2. Find also the value such that 95% of the outcomes are smaller than that value.

For part e) we create a sample of a distribution with a custom  $\mu$  and  $\sigma$  from a sample of the normal distribution.

```
part1(mean=2, sd=3)
```



```
## [1] "prob (x < -0.5) = 0.3085"  
## [1] "prob (-1 < x < 2) = 0.8186"  
## [1] "estimate 1a: 0.2800, estimate1b: 0.2028"  
## [1] "estimate 2a: 0.3100, estimate2b: 0.3407"  
qnorm(0.95, mean=2, sd=3)
```

```
## [1] 6.934561  
# part e:  
s1 = rnorm(1000)  
s1 = -10 + 5 * s1  
print(sprintf("mean = %.4f, stdev = %.4f", mean(s1),  
sqrt(var(s1))))
```

```
## [1] "mean = -9.9962, stdev = 4.8032"
```

In this case, the qqplots suggest that the samples have the shape of a normal distribution, but aren't from the standard normal distribution. Otherwise the scatter plot would align nicely with  $y=x$ , (shown as a red line).

## Exercise 2

reference: custom qqplots

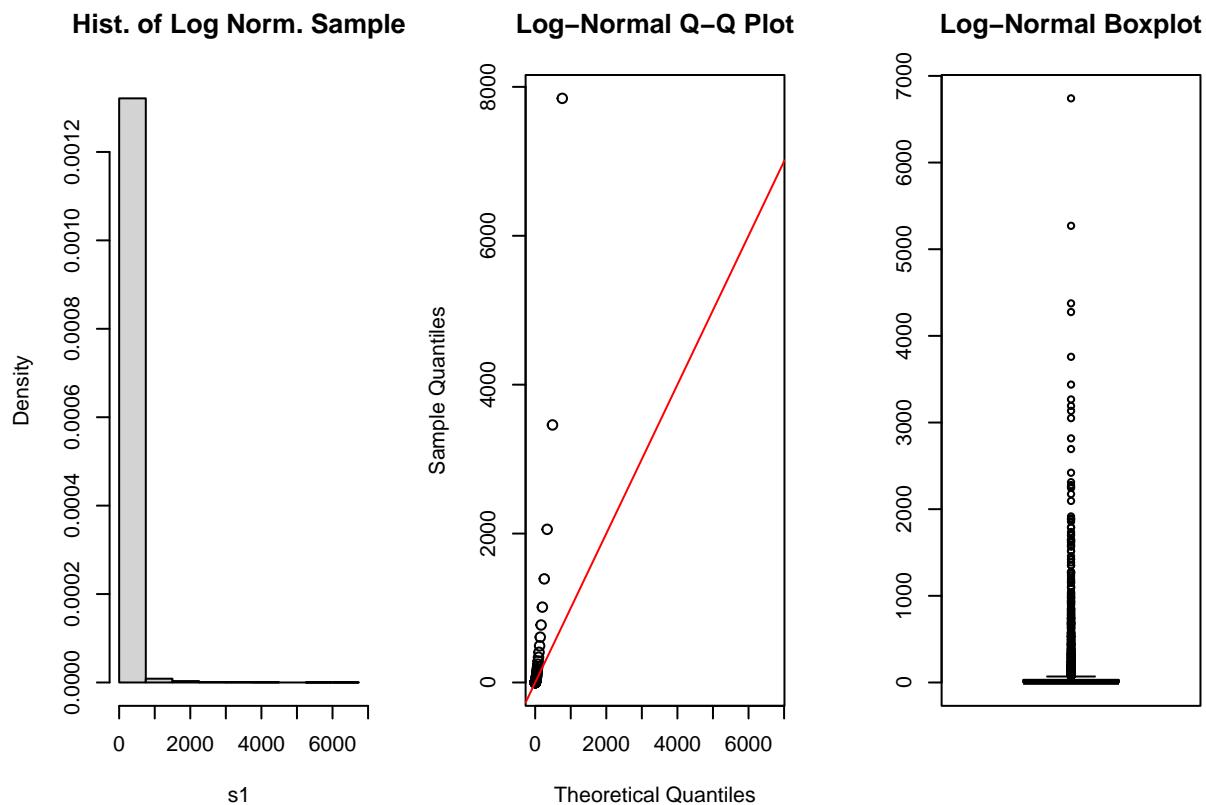
```
# configure plot tiling:  
par(mfrow=c(1,3))  
  
##### log norm sample:
```

```

s1 = rlnorm(10000, meanlog=2, sdlog=2)
hist(s1, freq=F, breaks=seq(0, max(s1), length=10), main="Hist. of Log Norm. Sample")

### custom qplot:
# select quantiles
quants = seq(0,1,length=100)
ref_quants = qlnorm(quants, meanlog=2, sdlog=3) # theoretical values for each quantile
act_quants = quantile(s1, quants) # actual values in sample
plot(act_quants, ref_quants, xlab="Theoretical Quantiles", ylab="Sample Quantiles")
abline(0, 1, col = 'red')
title(main="Log-Normal Q-Q Plot")
boxplot(s1)
title(main="Log-Normal Boxplot")

```

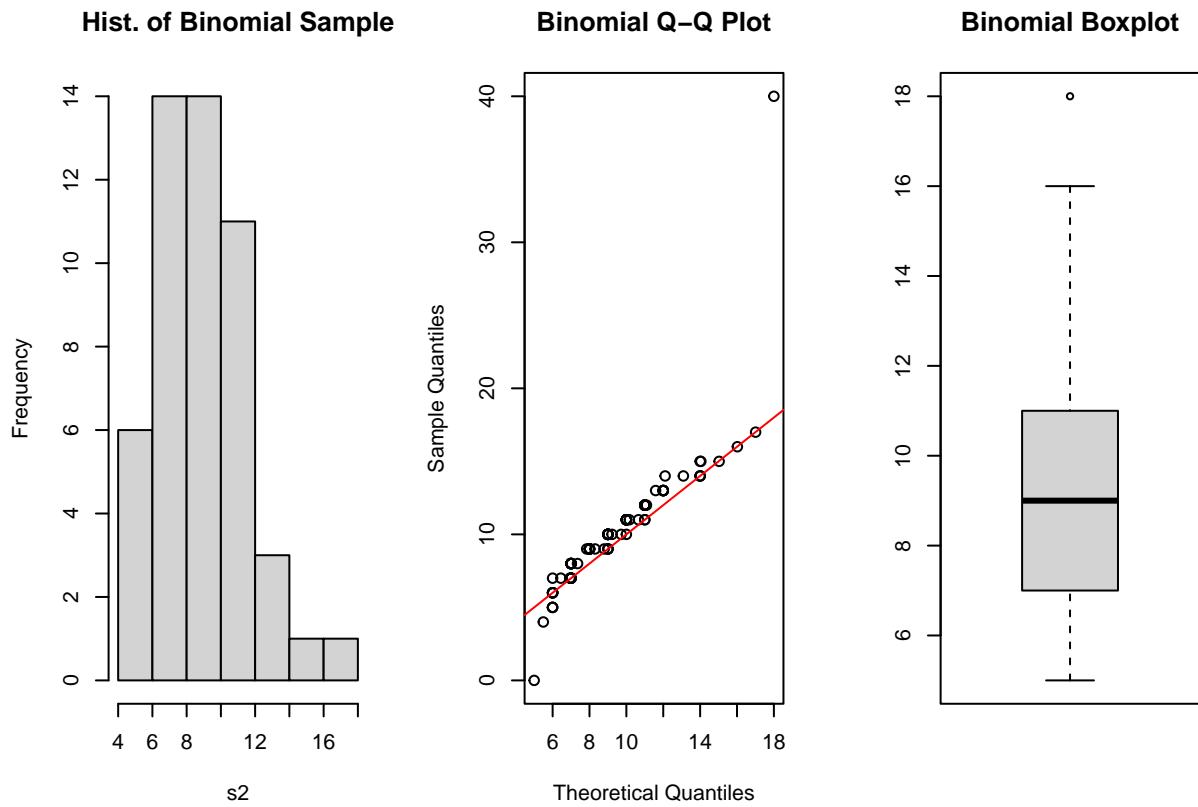


```

##### binomial dist:
# do 50 instances of sets of n=40 binomial trials
# (getting the number of successes for each instance)
s2 = rbinom(50, 40, 0.25)
hist(s2, main="Hist. of Binomial Sample")

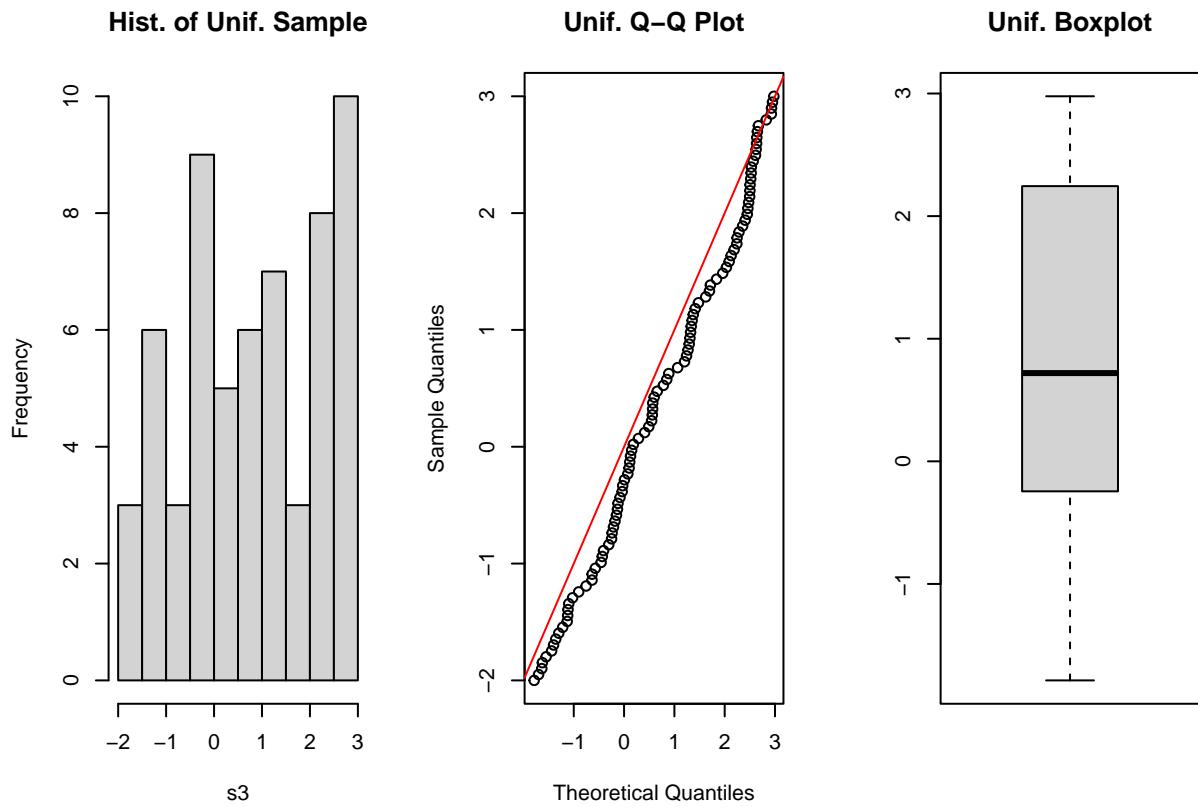
ref_quants = qbinom(quants, 40, 0.25)
act_quants = quantile(s2, quants)
plot(act_quants, ref_quants, xlab="Theoretical Quantiles", ylab="Sample Quantiles")
abline(0, 1, col = 'red')
title(main="Binomial Q-Q Plot")
boxplot(s2); title(main="Binomial Boxplot")

```



```
#####
# uniform dist:
s3 = runif(60, min=-2, max=3)
hist(s3, main="Hist. of Unif. Sample")

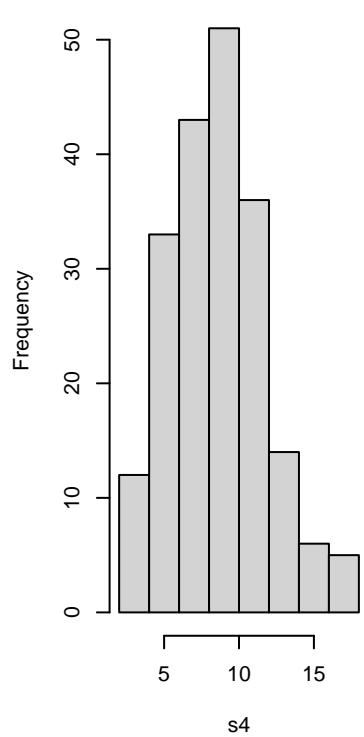
ref_quants = qunif(quants, min=-2, max=3)
act_quants = quantile(s3, quants)
plot(act_quants, ref_quants, xlab="Theoretical Quantiles", ylab="Sample Quantiles")
abline(0, 1, col = 'red')
title(main="Unif. Q-Q Plot")
boxplot(s3); title(main="Unif. Boxplot")
```



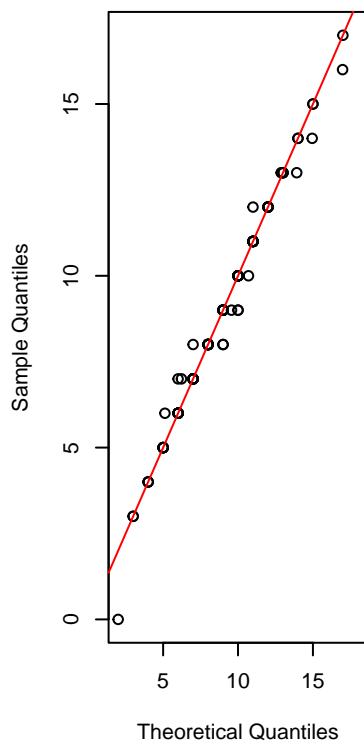
```
##### poisson dist:
s4 = rpois(200, 9)
hist(s4, main="Hist. of Poisson Sample")

ref_quants = qpois(quants, 9)
act_quants = quantile(s4, quants)
plot(act_quants, ref_quants, xlab="Theoretical Quantiles", ylab="Sample Quantiles")
abline(0, 1, col = 'red')
title(main="Poisson Q-Q Plot")
boxplot(s4); title(main="Poisson Boxplot")
```

**Hist. of Poisson Sample**



**Poisson Q-Q Plot**



**Poisson Boxplot**

