Assignment 0: Practicing R

Exercise 1: Normal Dist. in R

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

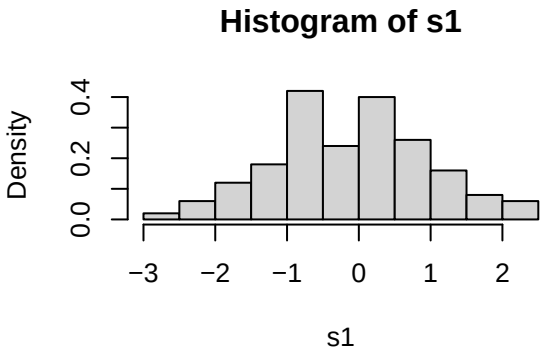
Part (b) and (c):

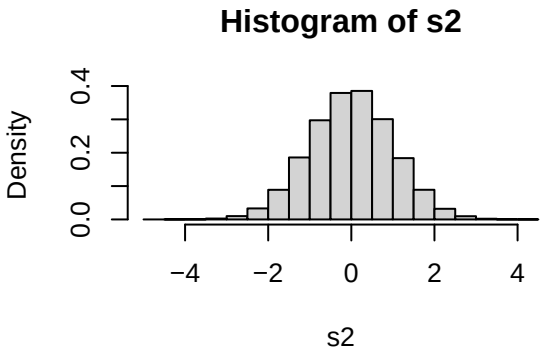
For a standard normal distribution, compute the following 3 probabilities: that an arbitrary outcome
is smaller than 2, that it is bigger then -0.5 and that it is between -1 and 2.

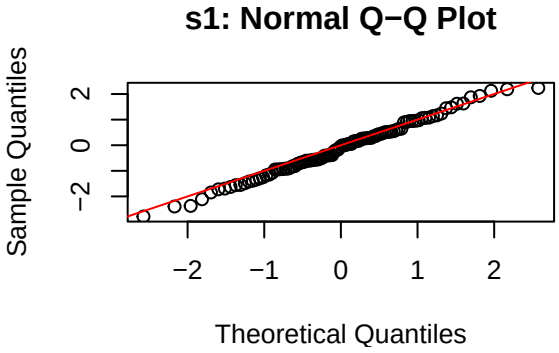
Can you verify the outcomes of b) using only the data from a)?

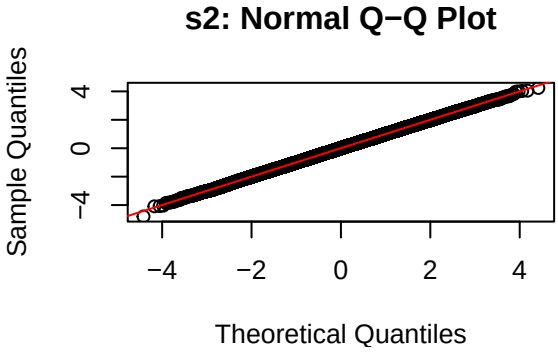
```
part1 = function(mean=0, sd=1) {
  par(mfrow=c(2,2)) # tile graphs
  s1 = rnorm(100, mean=mean, sd=sd)
  s2 = rnorm(100000, mean=mean, sd=sd)

  # 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: 0.4000, estimate1b: 0.3090"
## [1] "estimate 2a: 0.7800, estimate2b: 0.8176"
```

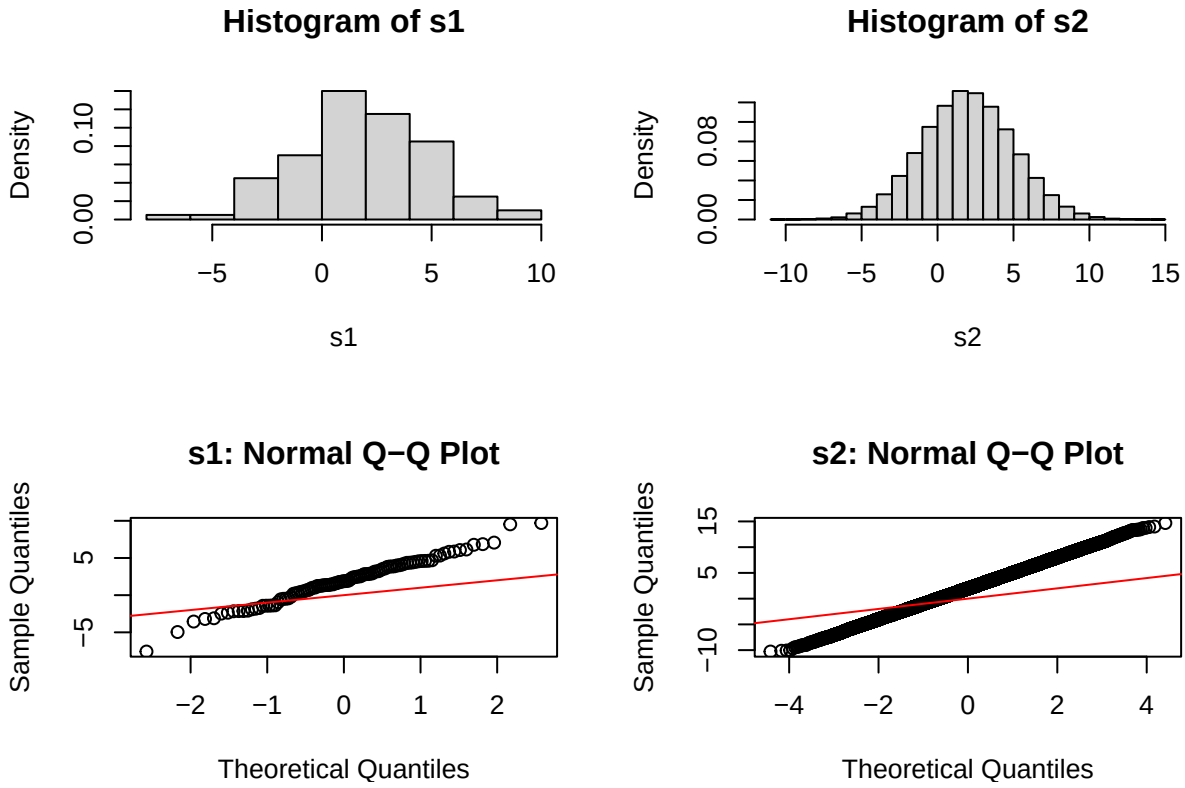
```
# generate two samples from std. norm. dist.
```

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 μ and σ 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: 0.2100, estimate1b: 0.2058"
## [1] "estimate 2a: 0.3400, estimate2b: 0.3432"
```

```
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 = -10.0392, stdev = 4.9806"
```

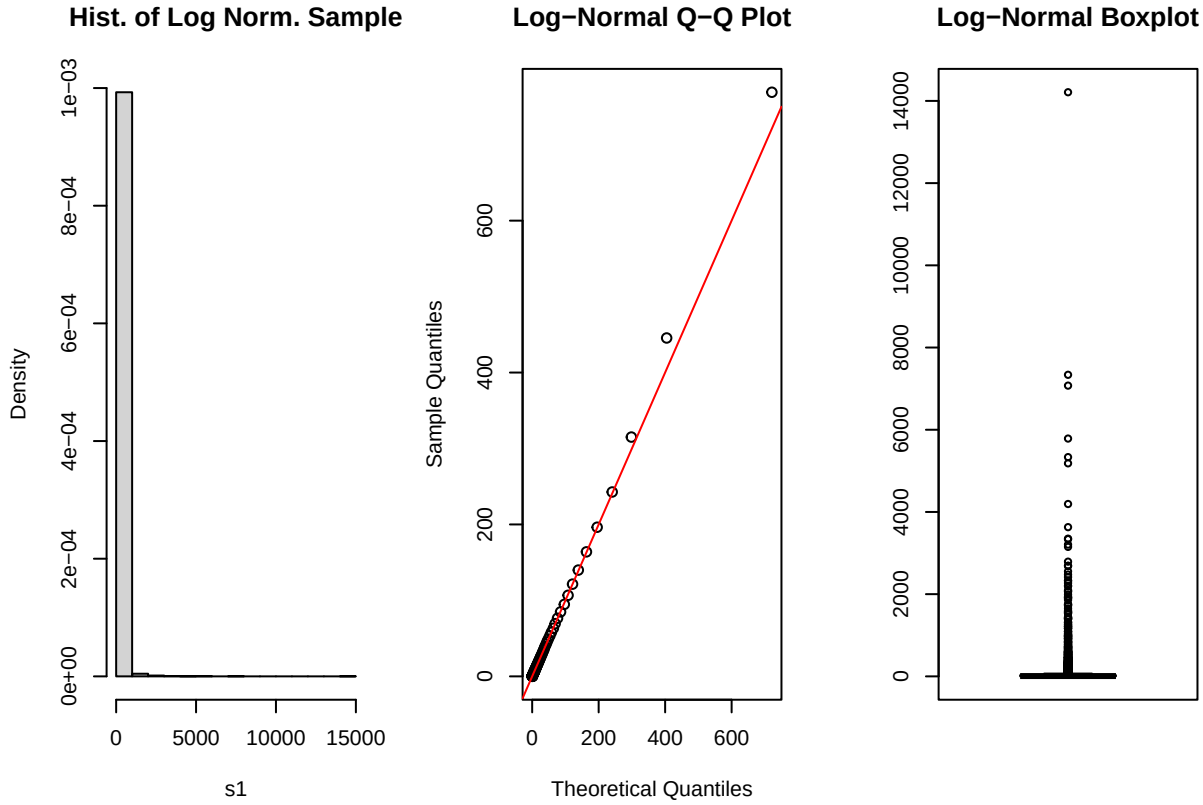
In this case, the qqplots suggest that the samples have the shape of a normal distribution, but aren't from the standard normal distribution. Otherwise the scatter plot would align nicely with $y=x$, (shown as a red line).

Exercise 2: Generating Samples/Plots

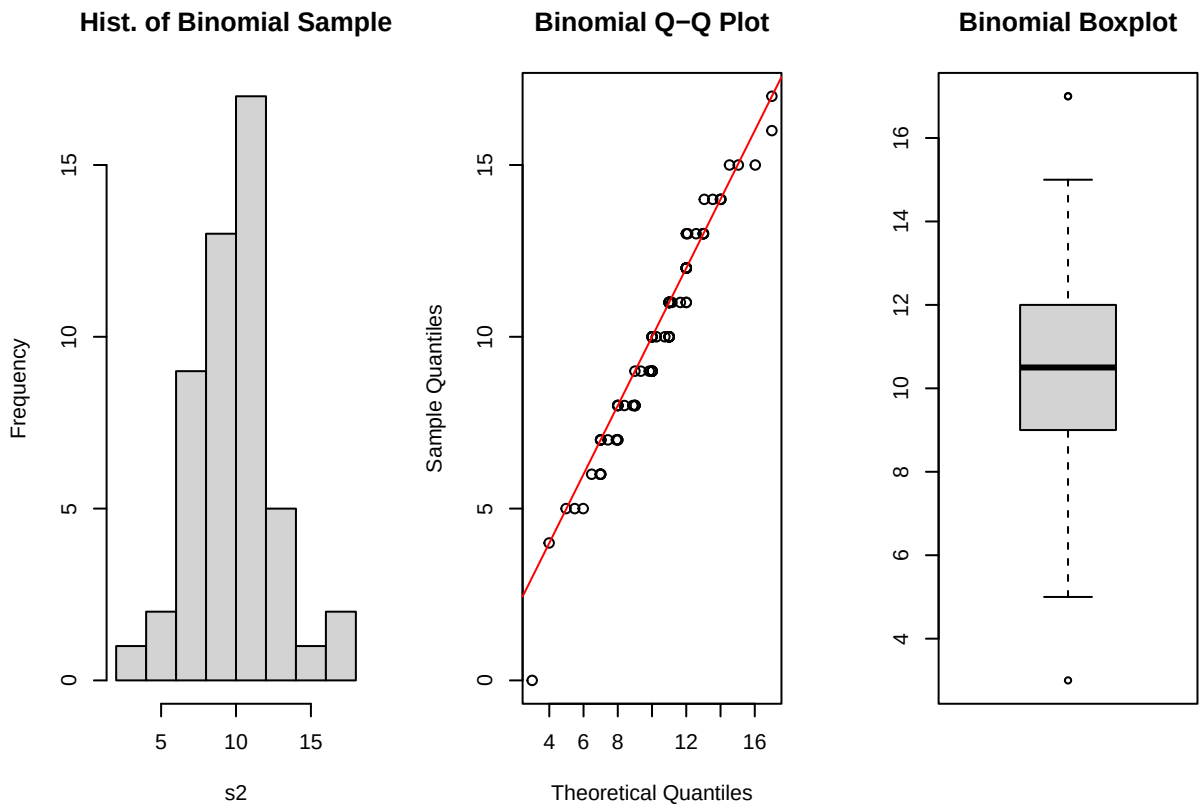
reference: custom qqplots

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

######## log norm sample:
```

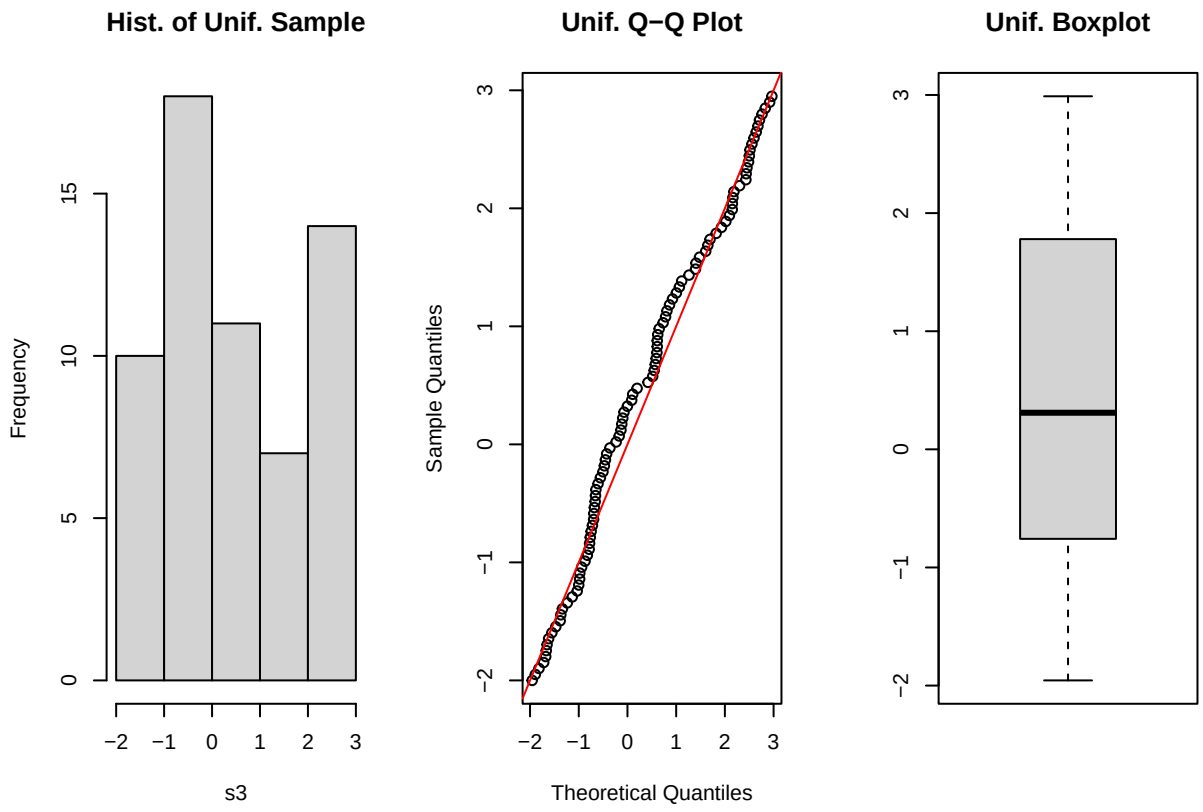
```r
s1 = rlnorm(10000, meanlog=2, sdlog=2)
#hist(s1, freq=F, breaks=seq(0, max(s1), length=10), main="Hist. of Log Norm. Sample")
hist(s1, freq=F, main="Hist. of Log Norm. Sample")
## custom qplot:
# select quantiles
quants = seq(0,1,length=100)
quants = quants[1:length(quants)-1] # omit 1.0 as that gets a value of inf in dist.
ref_quants = qlnorm(quants, meanlog=2, sdlog=2) # 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")
```



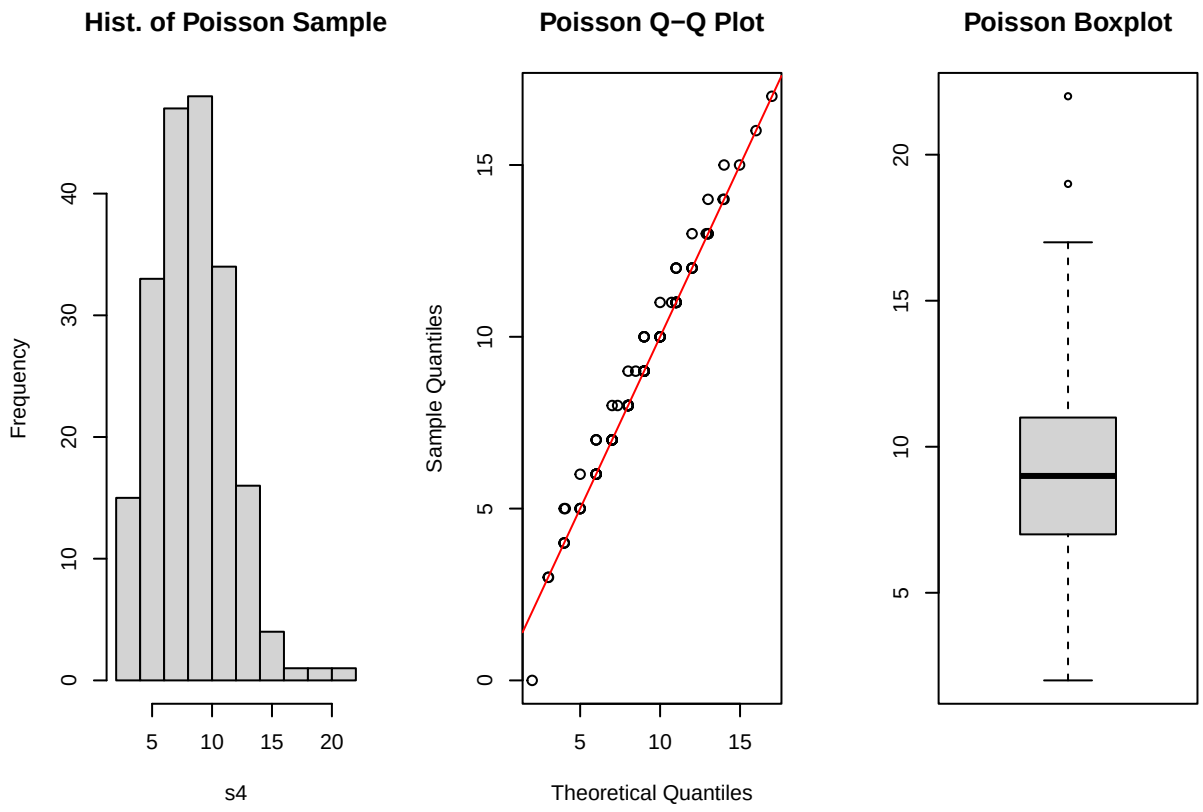

```r
######## binomial dist:
# do 50 instances of sets of n=40 binomial trials
#   (getting the number of successes for each instance)
s2 = rbinom(50, 40, 0.25)
hist(s2, main="Hist. of Binomial Sample")

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")
```




```r
######## 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")
```

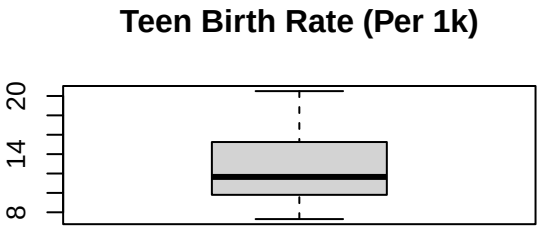


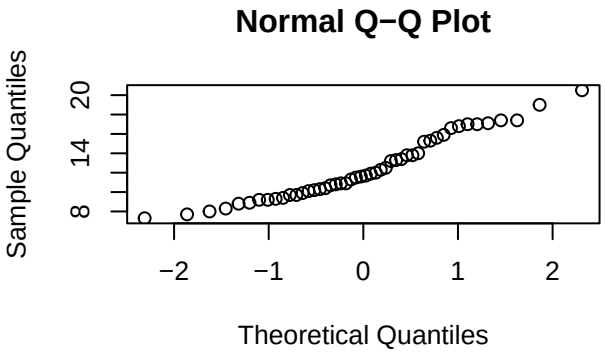
Exercise 3: Summarizing Data

```
data = read.table("mortality.txt", header=T)
#data

describe = function(d, title) {
  writeLines(sprintf("\n%s:", title))
  par(mfrow=c(2,2)) # tile graphs
  boxplot(d); title(main=title)
  hist(d, main=title)
  qqnorm(d)
  print(summary(d))
  print(sprintf("sd=%.4f", sd(d)))
}

teenTitle = "Teen Birth Rate (Per 1k)"
describe(data$teen, teenTitle)
```

```
##
## Teen Birth Rate (Per 1k):
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.30    9.85   11.65   12.43   15.22   20.50
## [1] "sd=3.2930"
```

```
mortTitle = "Infant Mort Rate (Per 1k)"
describe(data$mort, mortTitle)
```

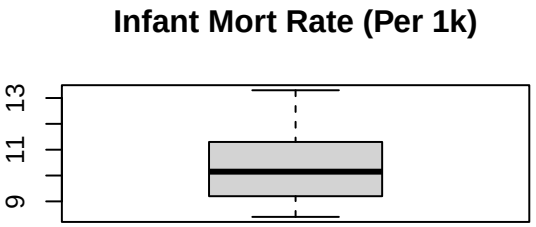


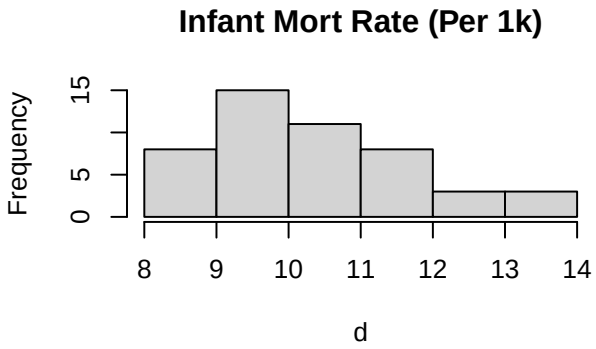


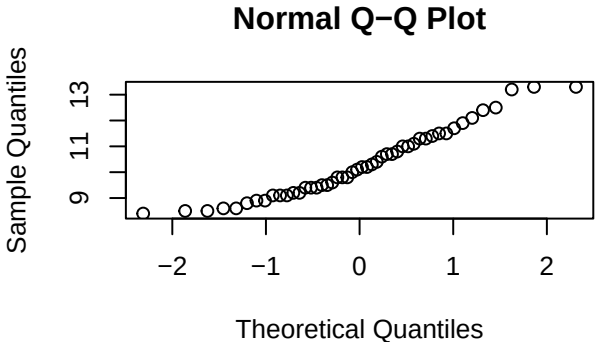

```
##
## Infant Mort Rate (Per 1k):
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    8.40    9.20   10.15   10.32   11.30   13.30
## [1] "sd=1.3499"
```

```
par(mfrow=c(2,2))
```

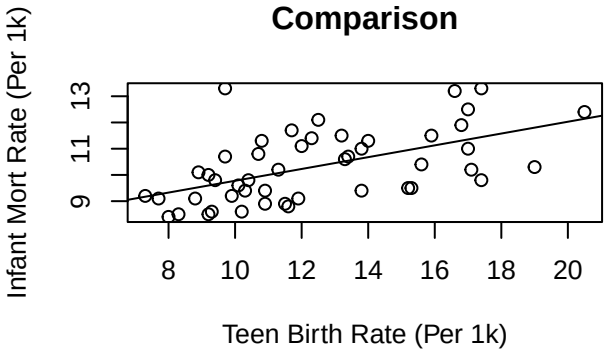







```
plot(data$teen, data$mort, xlab=teenTitle, ylab=mortTitle, main="Comparison")

# plot line of best fit (linear model)
abline(lm(data$mort ~ data$teen))
corCoef = cor(data$mort, data$teen)
print(sprintf("correlation coefiicient =%.4f", corCoef))
```

```
## [1] "correlation coefiicient =0.5491"
```



Teen birth rate and infant mortality rates both appear to be approximately normally distributed.

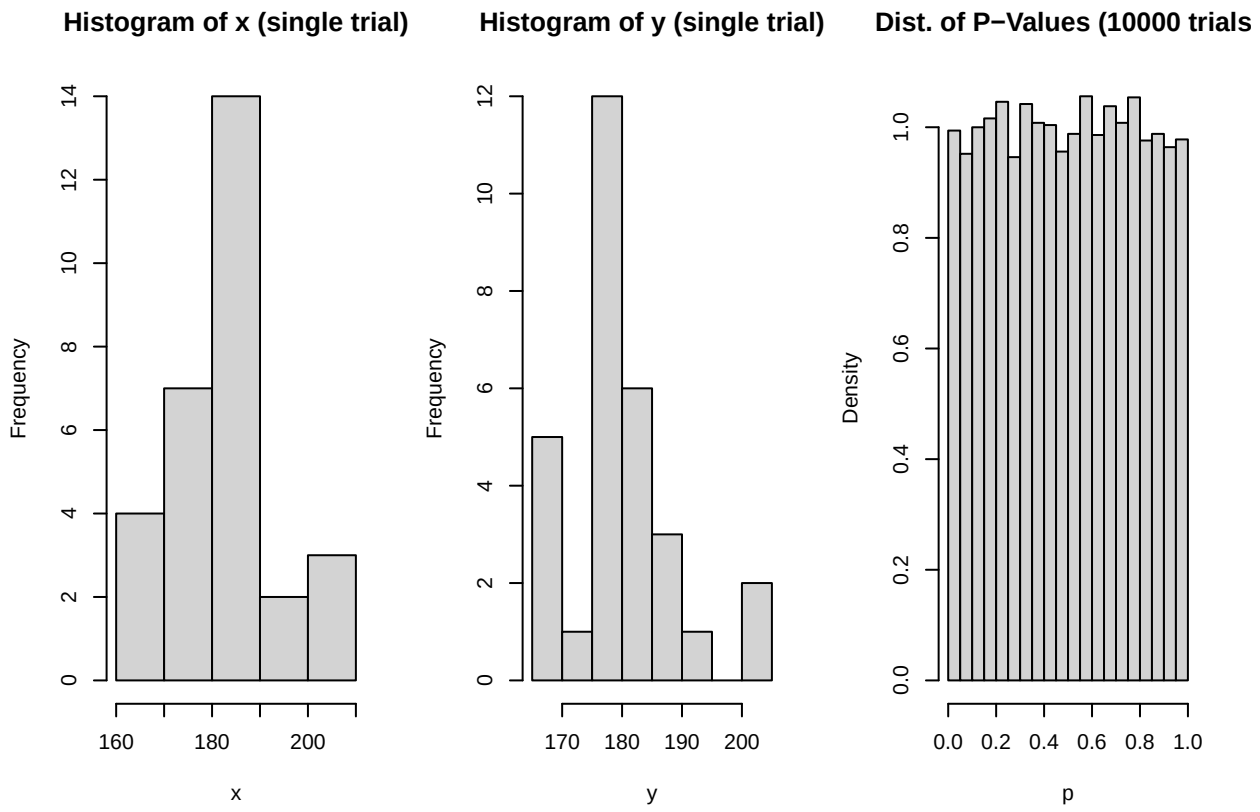
Teen birth rate and infant mortality appear to (roughly) have a linear relationship, although the correlation coefficient is only 0.549.

Exercise 4: P-values of the (2 sample) t-test

```
part4 = function(m, n, mu, nu, sd) {
  par(mfrow=c(1, 3))
  x = rnorm(n, mu, sd)
```

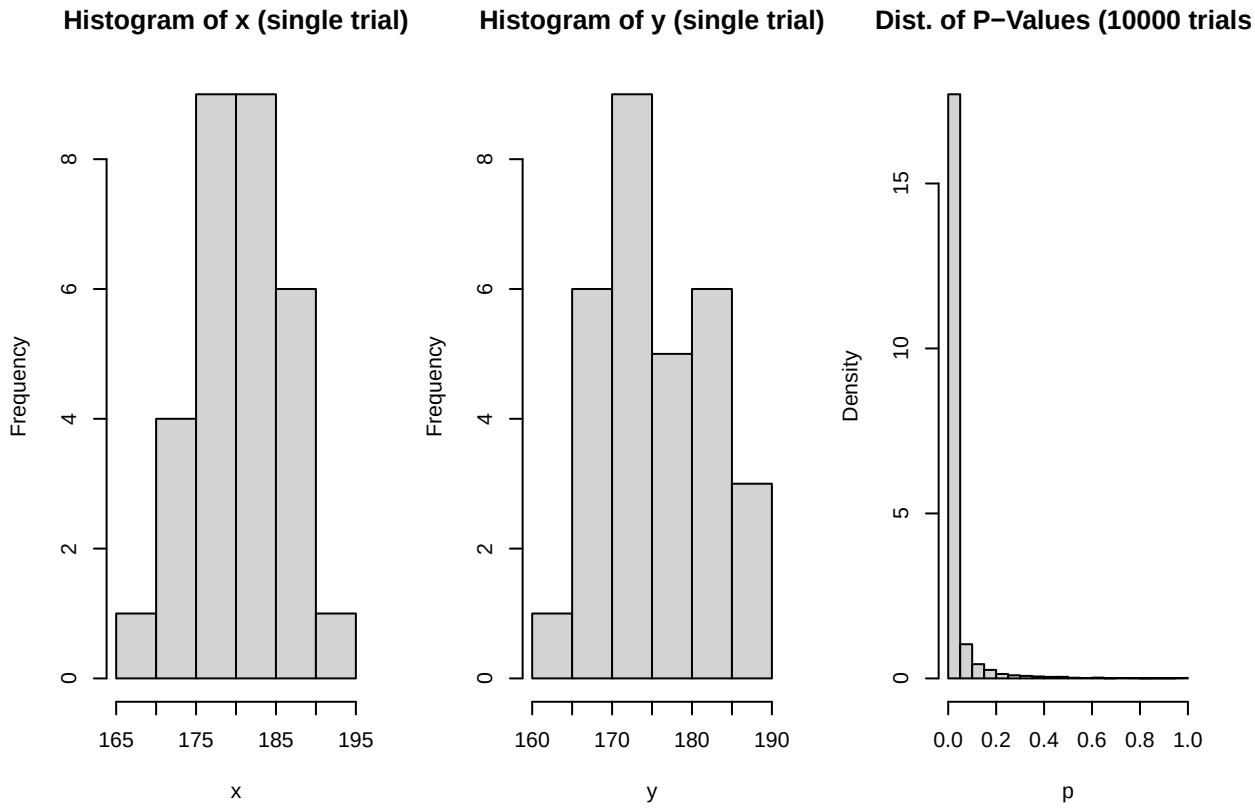
```
  y = rnorm(n, nu, sd)
  hist(x, main="Histogram of x (single trial)")
  hist(y, main="Histogram of y (single trial)")
  #t.test()
  # do 2 sample t test (and treat the variances as equal)
  #res = t.test(x, y, var.equal=T)
  #print(sprintf("single t-test result has p-value %.4f", res[[3]]))


  # estimate power of t-test:
  B = 10000
  p = numeric(B)
  for (b in 1:B) {
    x = rnorm(n, mu, sd)
    y = rnorm(n, nu, sd)
    res = t.test(x, y, var.equal=T, alternative="two.sided")
    p[b] = res[[3]]
  }
  power = mean(p < 0.05) # ratio of p-values < 0.05
  # calculate true power:
  truePower = power.t.test(n=n, sd=sd, delta=abs(mu-nu), type="two.sample", alternative="two.sided")
  # TODO: why is the "true" power off by a factor of 2?
  print(sprintf("estimated power = %.4f (using %d trials) , true power = %.4f", power, B, truePower$pow
  print(truePower)
 #browser() # breakpoint

  #par(mfrow=c(1, 1))
  hist(p, freq=F, main=sprintf("Dist. of P-Values (%d trials)", B))
}

part4(30, 30, 180, 180, 10)
```

```
## [1] "estimated power = 0.0497 (using 10000 trials) , true power = 0.0250"
##
##      Two-sample t test power calculation
##
##               n = 30
##           delta = 0
##              sd = 10
##       sig.level = 0.05
##           power = 0.025
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```




```
part4(30, 30, 180, 175, 6)
```

```
## [1] "estimated power = 0.8852 (using 10000 trials) , true power = 0.8876"
##
##      Two-sample t test power calculation
##
##               n = 30
##           delta = 5
##              sd = 6
##       sig.level = 0.05
##           power = 0.8875572
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```



The findings for part (b) suggest that the statistical power is only about 0.05. Meaning there's a 5% chance of rejecting the H_0. This is to be expected, because in this case the H_0 is correct (x and y come from the same distribution), and with a significance-level of 0.05, the statistical power when H_0 is correct should be 0.05 as well.

The findings for part (c) indicate the statistical power for this scenario is around 0.90, which means there's a 90% chance of (correctly) rejecting H_0 (as in this case the null hypothesis that x and y come from the same distributions is incorrect). This is a good sign, as we expect it to be easy to identify the distributions are different (given the large sample size of 1000).