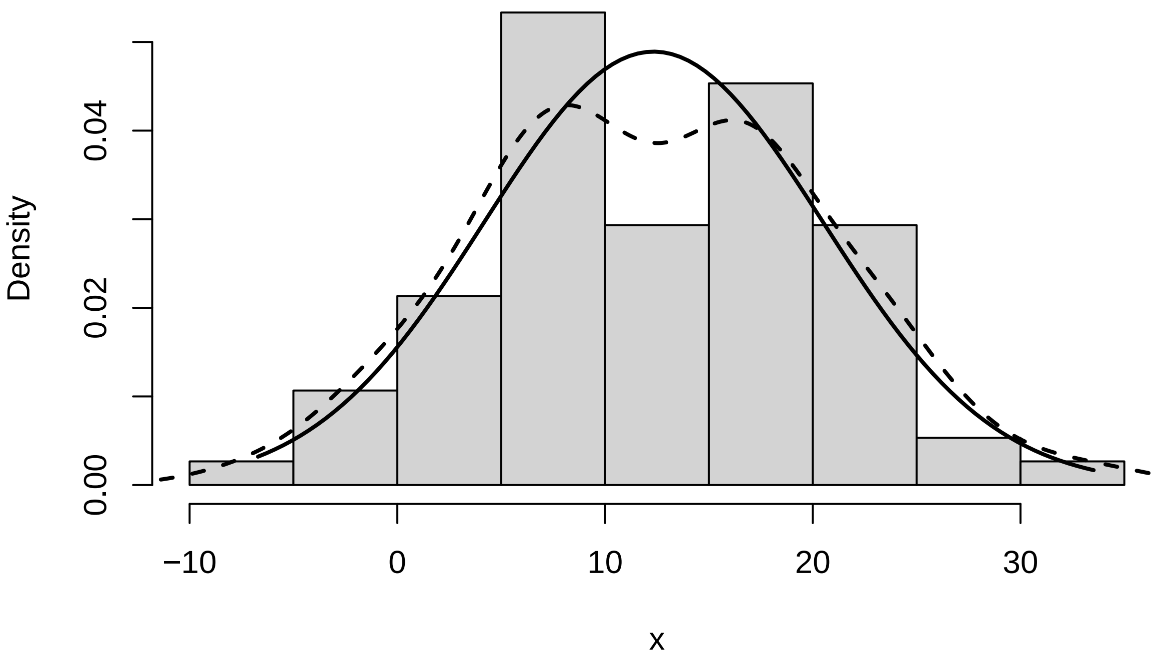
Red-Team-GOF_R

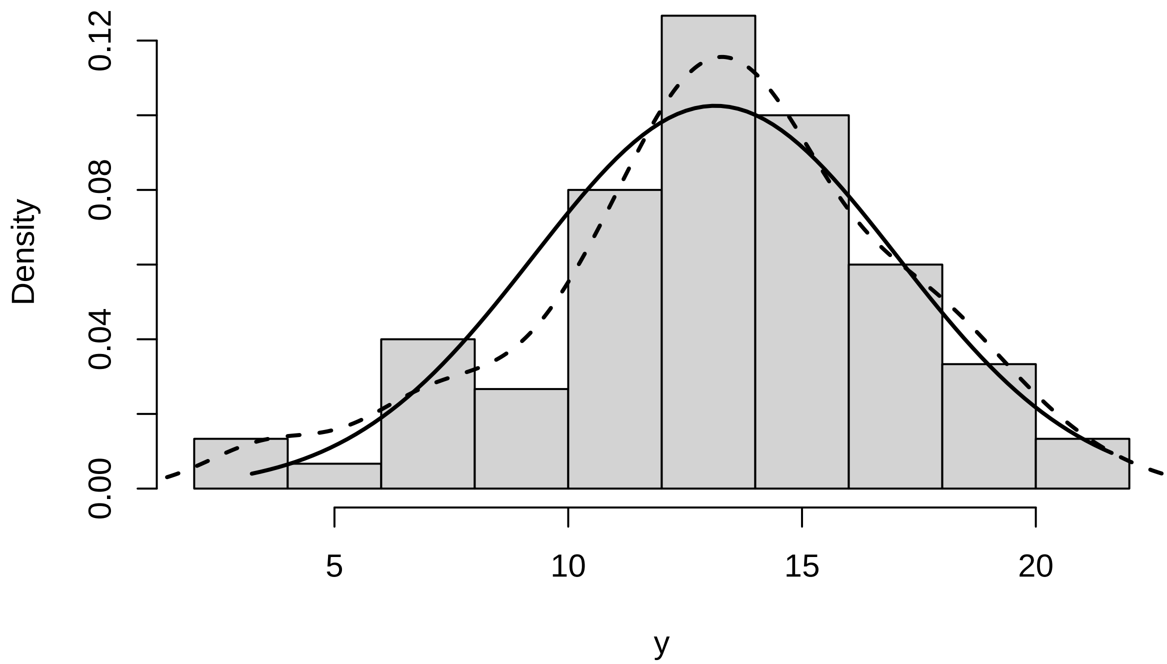
Red Team

```r
x = rnorm(75, mean=14, sd = 8) #x ~ N(10,4)
y = rnorm(75, mean=14, sd = 4) #y ~ N(14,4)
cert = data.frame(x,y)
#hist(x)
#hist(y)
with(cert, hist(x, main="", freq=FALSE))
with(cert, lines(density(x), main="X", lty=2, lwd=2))
xvals = with(cert, seq(from=min(x), to=max(x), length=100))
with(cert, lines(xvals, dnorm(xvals, mean(x), sd(x)), lwd=2))
```




```r
# Changing the values of the mean and standard deviation will shift the center
# of the distribution. Furthermore, specifically changing the standard
# deviation will affect the  variability of the data. Changing the values will
# affect the goodness-of-fit to a normal distribution, and the p-values will inform
# you how well the data set apply to the different types of distributions. To have
# a higher p-value suggests a better fit, and a lower p-value suggests a poorer fit.
```

```r
with(cert, hist(y, main="", freq=FALSE))
with(cert, lines(density(y), main="", lty=2, lwd=2))
xvals = with(cert, seq(from=min(y), to=max(y), length=100))
with(cert, lines(xvals, dnorm(xvals, mean(y), sd(y)), lwd=2))
```




```
if (!require("vcd")) install.packages("vcd", dep=TRUE)
```

```
## Loading required package: vcd

## Loading required package: grid
```

```
library("vcd")
#favstats(x)
mean(x, trim=.05)
```

```
## [1] 12.27626
```

```
quantile(x, seq(from=.025, to= .975, by=.1))
```

```
##      2.5%     12.5%     22.5%     32.5%     42.5%     52.5%     62.5%     72.5%
## -1.540010  3.408854  5.810524  7.897689  9.505448 13.124891 15.987547 17.060646
##     82.5%     92.5%
## 20.788193 23.601457
```

```
t.test(x, mu=12, conf.level=.9) #test for mu=12 and 90 percent ci
```

```
##
##  One Sample t-test
##
## data:  x
## t = 0.36003, df = 74, p-value = 0.7198
## alternative hypothesis: true mean is not equal to 12
## 90 percent confidence interval:
##  10.77035 13.90779
## sample estimates:
## mean of x
##  12.33907
```

```r
#favstats(y)
mean(y, trim=.05)
```

```
## [1] 13.24062
```

```r
quantile(y, seq(from=.025, to= .975, by=.1))
```

```
##      2.5%     12.5%     22.5%     32.5%     42.5%     52.5%     62.5%     72.5%
##  3.978578  8.422427 10.808819 11.966969 12.756522 13.505295 14.327741 15.097827
##     82.5%     92.5%
## 16.578981 18.289454
```

```r
t.test(y, mu=12, conf.level=.9) #test for mu=12 and 90 percent ci
```

```
##
##  One Sample t-test
##
## data:  y
## t = 2.5494, df = 74, p-value = 0.01286
## alternative hypothesis: true mean is not equal to 12
## 90 percent confidence interval:
##  12.39703 13.89375
## sample estimates:
## mean of x
##  13.14539
```

```r
#if (!require("coin")) install.packages("coin", dep=TRUE)
```

```r
#library("coin")
wilcox.test(x,y)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  x and y
## W = 2605, p-value = 0.4365
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(x,y)
```

```
##
##  Welch Two Sample t-test
##
## data:  x and y
## t = -0.77275, df = 106.02, p-value = 0.4414
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.875048  1.262411
## sample estimates:
## mean of x mean of y
##  12.33907  13.14539
```

```r
ks.test(x,y)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  x and y
## D = 0.28, p-value = 0.005377
## alternative hypothesis: two-sided
```

```r
if (!require("nortest")) install.packages("nortest", dep=TRUE)
```

```
## Loading required package: nortest
```

```r
library("nortest")
ad.test(x)
```

```
##
##  Anderson-Darling normality test
##
## data:  x
## A = 0.33298, p-value = 0.5055
```

```r
cvm.test(x)
```

```
##
##  Cramer-von Mises normality test
##
## data:  x
## W = 0.066463, p-value = 0.307
```

```r
lillie.test(x)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  x
## D = 0.078637, p-value = 0.3008
```

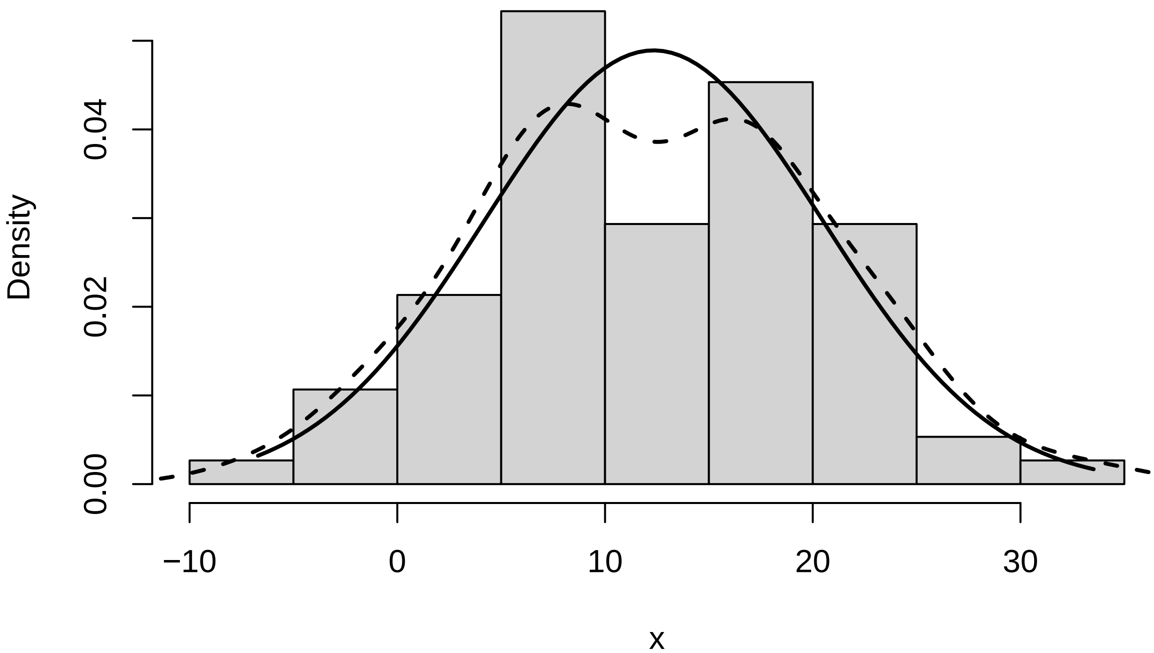
```r
pearson.test(x)
```

```
##
##  Pearson chi-square normality test
##
## data:  x
## P = 16.04, p-value = 0.06605
```

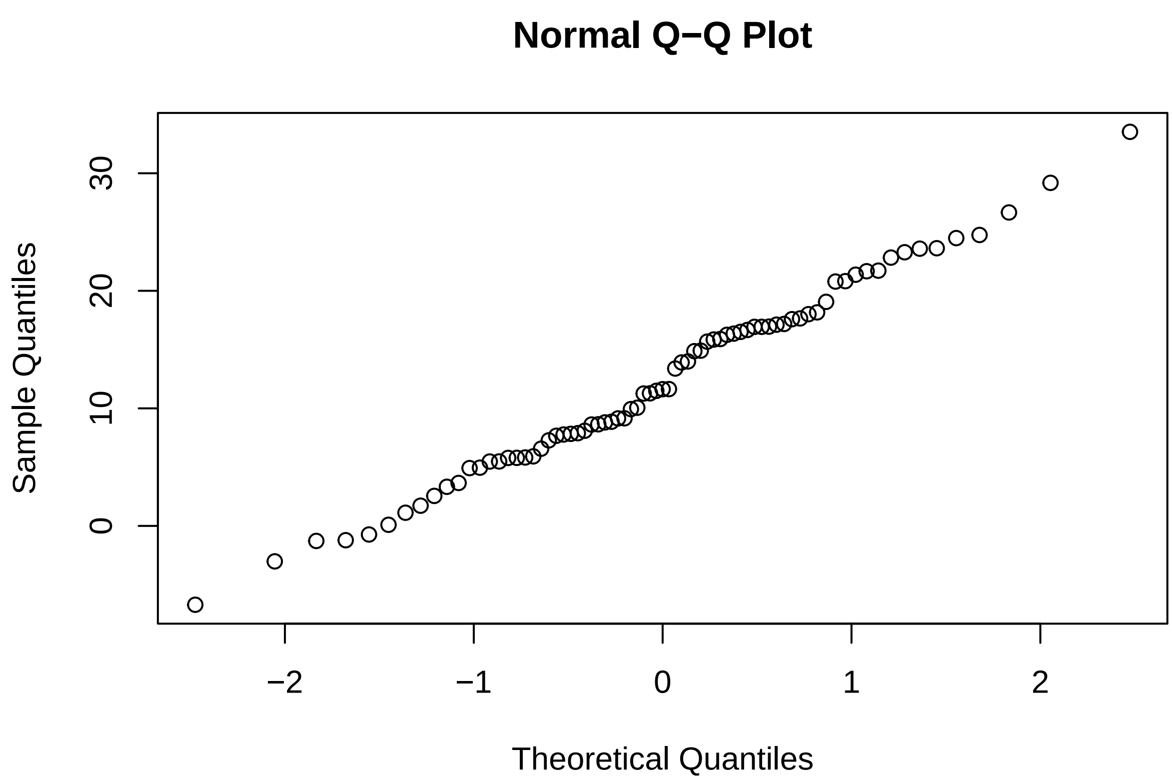
```r
sf.test(x)
```

```
##
##  Shapiro-Francia normality test
##
## data:  x
## W = 0.99142, p-value = 0.8251
```

```
with(cert, hist(x, main="", freq=FALSE))
with(cert, lines(density(x), main="X", lty=2, lwd=2))
xvals = with(cert, seq(from=min(x), to=max(x), length=100))
with(cert, lines(xvals, dnorm(xvals, mean(x), sd(x)), lwd=2))
```




```
qqnorm(x)
```




```
x = 2*rlnorm(75, mean=0, sd = 1) + 10
cert = data.frame(x,y) #redefined data.frame with new x
ad.test(x)
```

```
##
##  Anderson-Darling normality test
##
## data:  x
## A = 6.7348, p-value < 2.2e-16
```

```
cvm.test(x)
```

```
## Warning in cvm.test(x): p-value is smaller than 7.37e-10, cannot be computed
## more accurately
```

```
##
##  Cramer-von Mises normality test
##
## data:  x
## W = 1.2267, p-value = 7.37e-10
```

```
lillie.test(x)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  x
## D = 0.21714, p-value = 2.128e-09
```

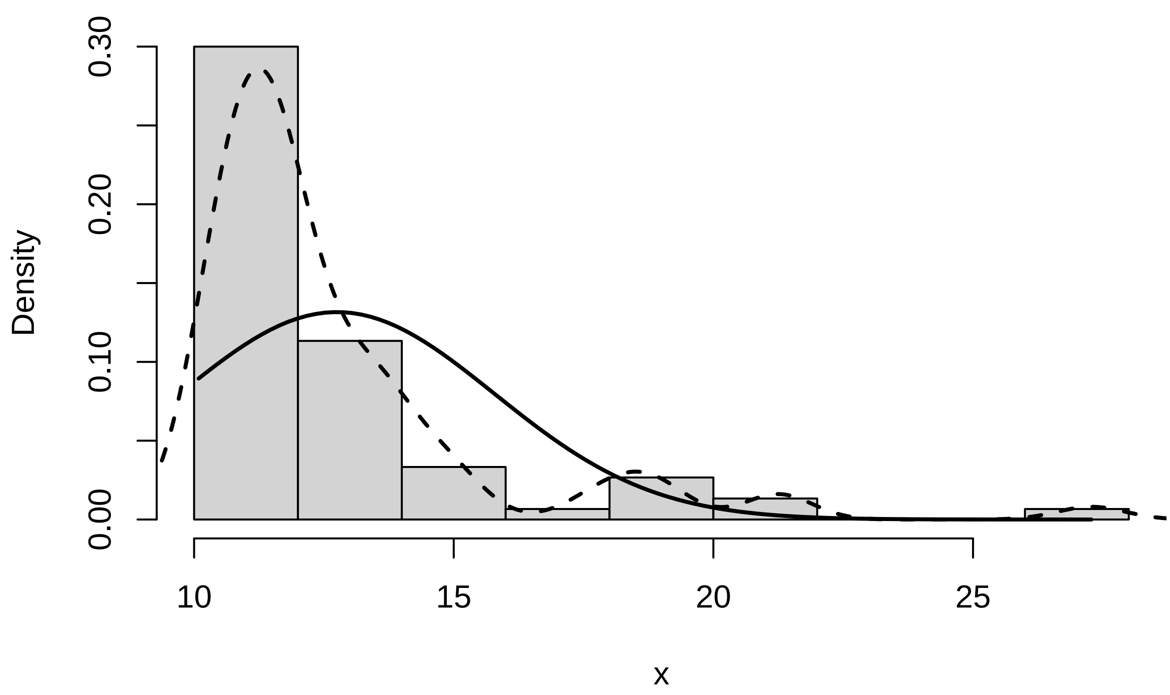
```r
pearson.test(x)
```

```
##
##  Pearson chi-square normality test
##
## data:  x
## P = 62.44, p-value = 4.529e-10
```

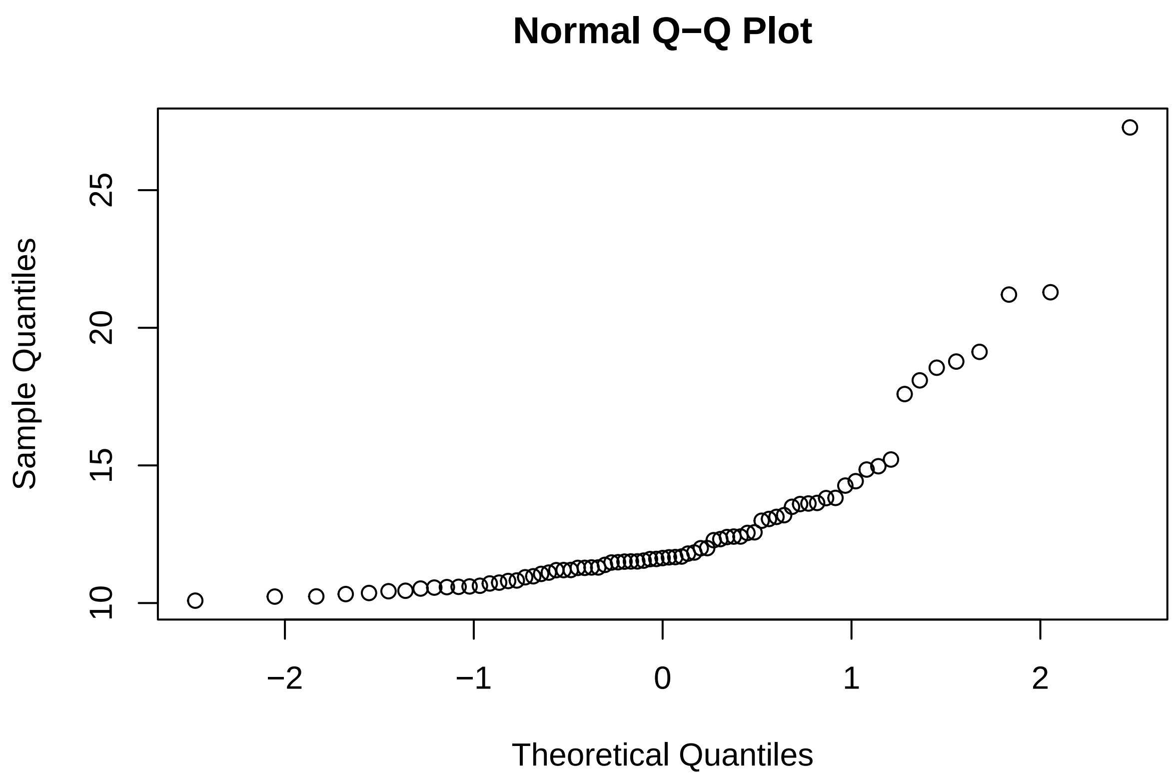
```r
sf.test(x)
```

```
##
##  Shapiro-Francia normality test
##
## data:  x
## W = 0.70721, p-value = 1.684e-09
```

```r
with(cert, hist(x, main="", freq=FALSE))
with(cert, lines(density(x), main="X", lty=2, lwd=2))
xvals = with(cert, seq(from=min(x), to=max(x), length=100))
with(cert, lines(xvals, dnorm(xvals, mean(x), sd(x)), lwd=2))
```




```r
qqnorm(x)
```


Normal Q−Q Plot
Sample Quantiles
10
15
20
25
−2
−1
0
1
2
Theoretical Quantiles