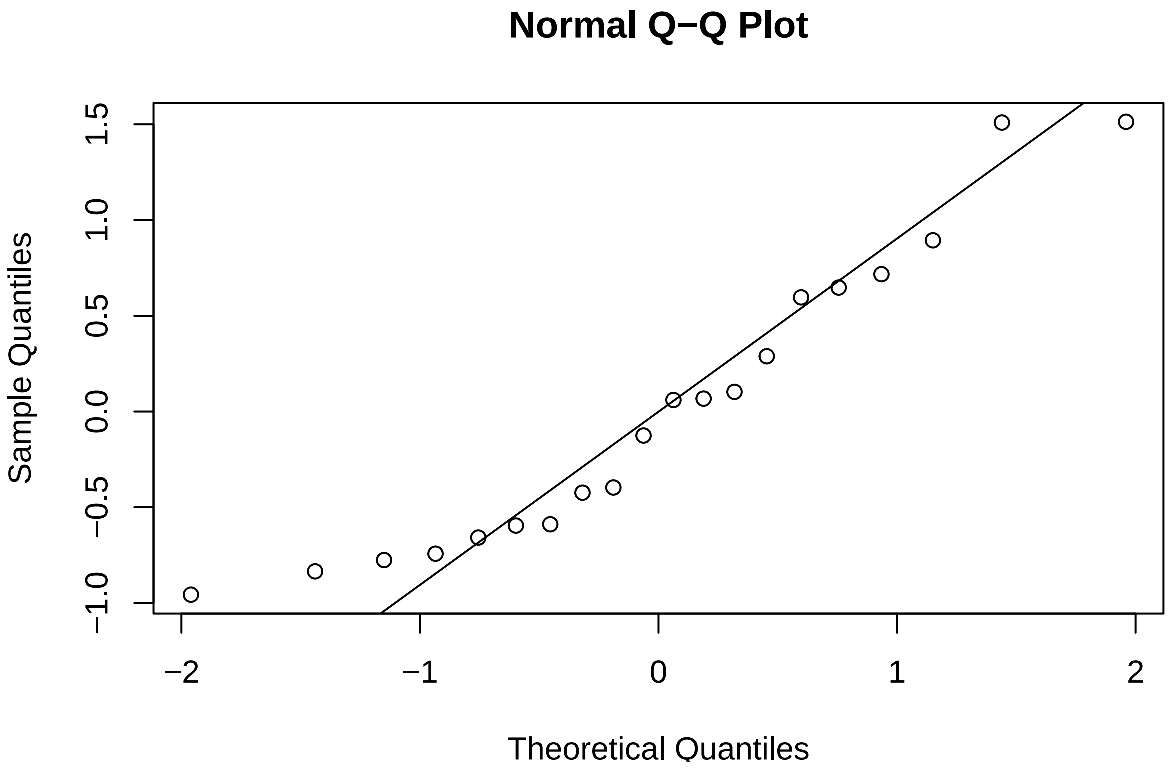
08112023_Statistical_Learning

Mattia G.

2023-11-08

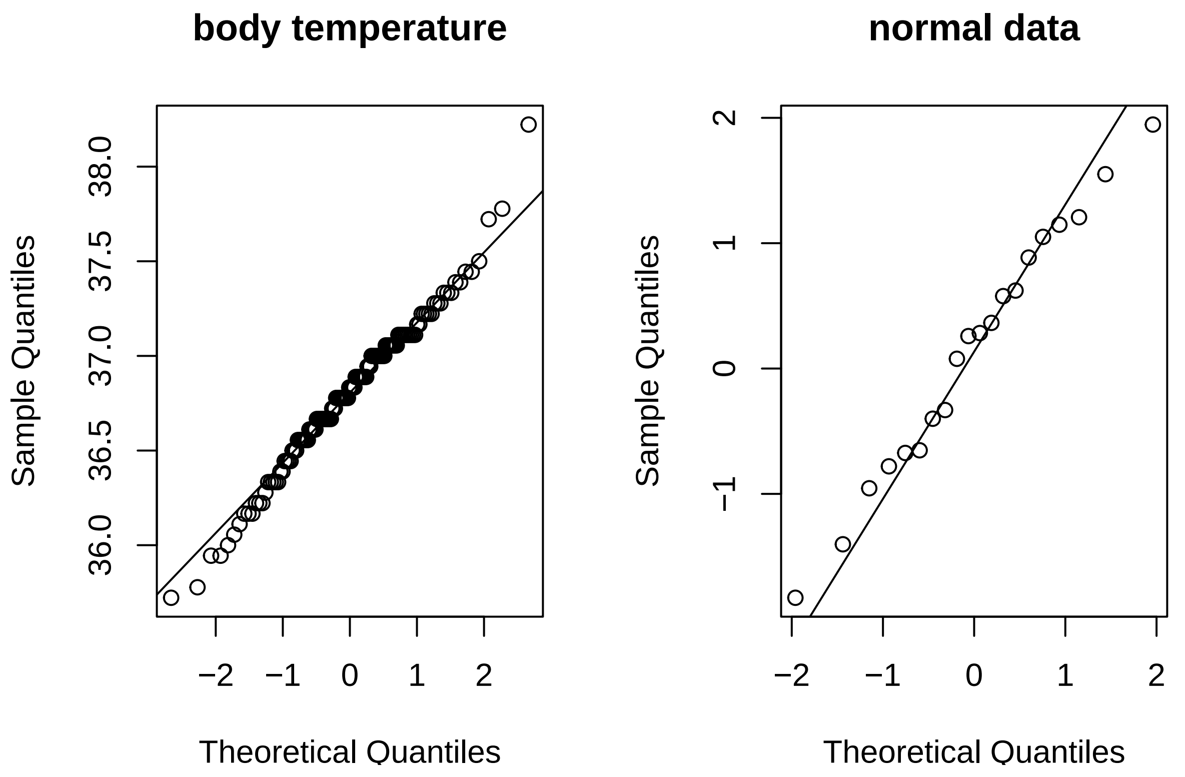
```
##################################
# Loading data into R
##################################

setwd("/Users/mattiagugole/Downloads/normtemp-data")

temp.data <- read.table("normtemp.txt", head=TRUE)
attach(temp.data)
temp.C <- (temperature-32)*5/9




# body temperatures and comparison with normal distribution

x <- rnorm(20)
qqnorm(x)
qqline(x)
```



```
normtemp <- read.table("normtemp.txt", head=TRUE)
attach(normtemp)
```

```
## The following objects are masked from temp.data:
##
##     gender, hr, temperature
```

```
temp.C <- (temperature-32)*5/9


par(mfrow=c(1, 2))
qqnorm(temp.C, main="body temperature")
qqline(temp.C)

x <- rnorm(20)
qqnorm(x, main="normal data")
qqline(x)
```



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

# missing values
#
# default na.strings = "NA"
mydata <- read.table("normtemp-with-NA.txt", head=TRUE)

mydata$temperature[1:10]
```

```
##  [1] "96.3" "*"    "96.9" "97.0" "97.1" "97.1" "97.1" "97.2" "97.3" "97.4"
```

```
is.vector(mydata$temperature)
```

```
## [1] TRUE
```

```
is.character(mydata$temperature)
```

```
## [1] TRUE
```

```
is.numeric(mydata$temperature)
```

```
## [1] FALSE
```

```
# set na.strings="*"
mydata <- read.table("normtemp-with-NA.txt", head=TRUE, na.strings = "*")

mydata$temperature[1:10]
```

```
##  [1] 96.3   NA 96.9 97.0 97.1 97.1 97.1 97.2 97.3 97.4
```

```
is.vector(mydata$temperature)
```

```
## [1] TRUE
```

```
is.character(mydata$temperature)
```

```
## [1] FALSE
```

```
is.numeric(mydata$temperature)
```

```
## [1] TRUE
```

```
# factors
#
mydata <- read.table("normtemp-with-ordinal-var.txt", head=TRUE, comment.char = "#")

is.vector(mydata$age)
```

```
## [1] TRUE
```

```
is.character(mydata$age)
```

```
## [1] TRUE
```

```
is.factor(mydata$age)
```

```
## [1] FALSE
```

```
age <- mydata$age
is.vector(mydata$age)
```

```
## [1] TRUE
```

```
is.vector(age)
```

```
## [1] TRUE
```

```
is.factor(age)
```

```
## [1] FALSE
```

```
# convert into a factor (categorical variable)
age.f <- factor(age)
is.vector(age.f)
```

```
## [1] FALSE
```

```
is.factor(age.f)
```

```
## [1] TRUE
```

```
# different behaviour of the print() function and of summary()
age[1:10]
```

```
##  [1] "<30"      "<30"      "[50, 70)" "<30"      ">=70"     ">=70"
##  [7] "[50, 70)" "[50, 70)" "<30"      "<30"
```

```
age.f[1:10]
```

```
##  [1] <30      <30      [50, 70) <30      >=70     >=70     [50, 70) [50, 70)
##  [9] <30      <30
## Levels: [30, 50) [50, 70) <30 >=70
```

```
summary(age)
```

```
##    Length     Class      Mode
##       130 character character
```

```
summary(age.f)
```

```
## [30, 50) [50, 70)      <30     >=70
##       18       39       56       17
```

```
is.ordered(age.f)
```

```
## [1] FALSE
```

```
# convert into an ordered factor

age.fo <- factor(age, ordered=TRUE, levels=  c("<30", "[30, 50)", "[50, 70)", ">=70" ))

# change directly the data.frame
mydata$age <- factor(mydata$age, ordered=TRUE, levels=  c("<30", "[30, 50)", "[50, 70)", ">=70" ))



is.ordered(age.fo)
```

```
## [1] TRUE
```

```
summary(age.f)
```

```
## [30, 50) [50, 70)      <30     >=70
##       18       39       56       17
```

```
summary(age.fo)
```

```
##      <30 [30, 50) [50, 70)     >=70
##       56       18       39       17
```

```
# change directly the data.frame
mydata$age <- factor(mydata$age, ordered=TRUE, levels=  c("<30", "[30, 50)", "[50, 70)", ">=70" ))


######################################################
# Probability distributions related to the normal one
######################################################
```

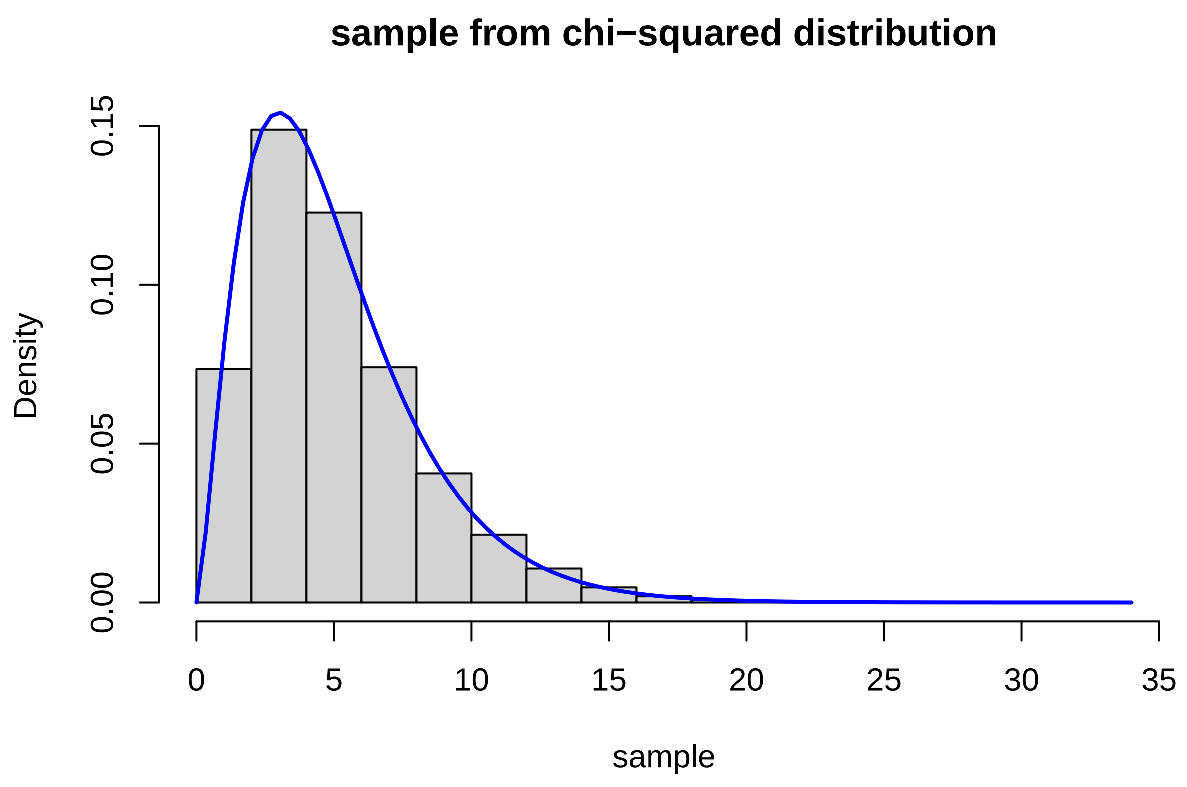
```r
# chi-squared distribution generate simulated values

# generate a value from a chi-square distribution
# with 5 degrees of freedom

# applying the definition

df <- 5
x <- rnorm(df)
sum(x^2)
```

```
## [1] 7.390835
```

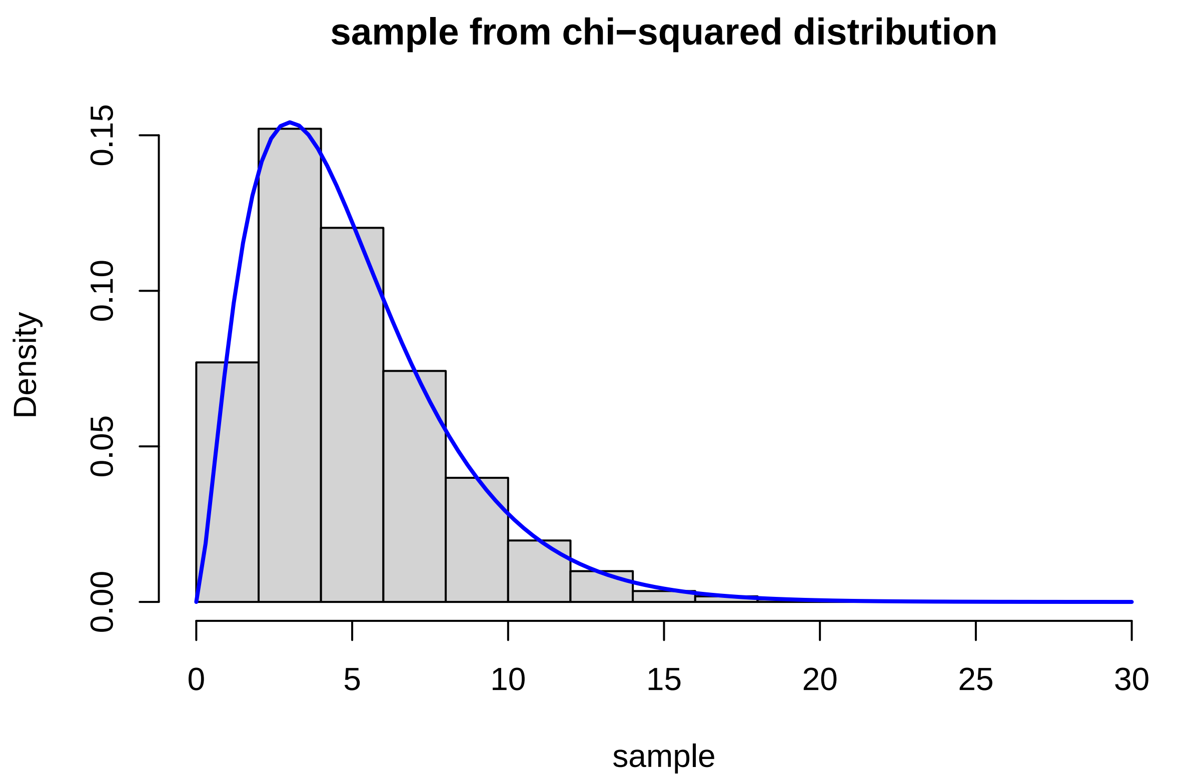
```r
# built-in R function

rchisq(1, df=5)
```

```
## [1] 2.813109
```

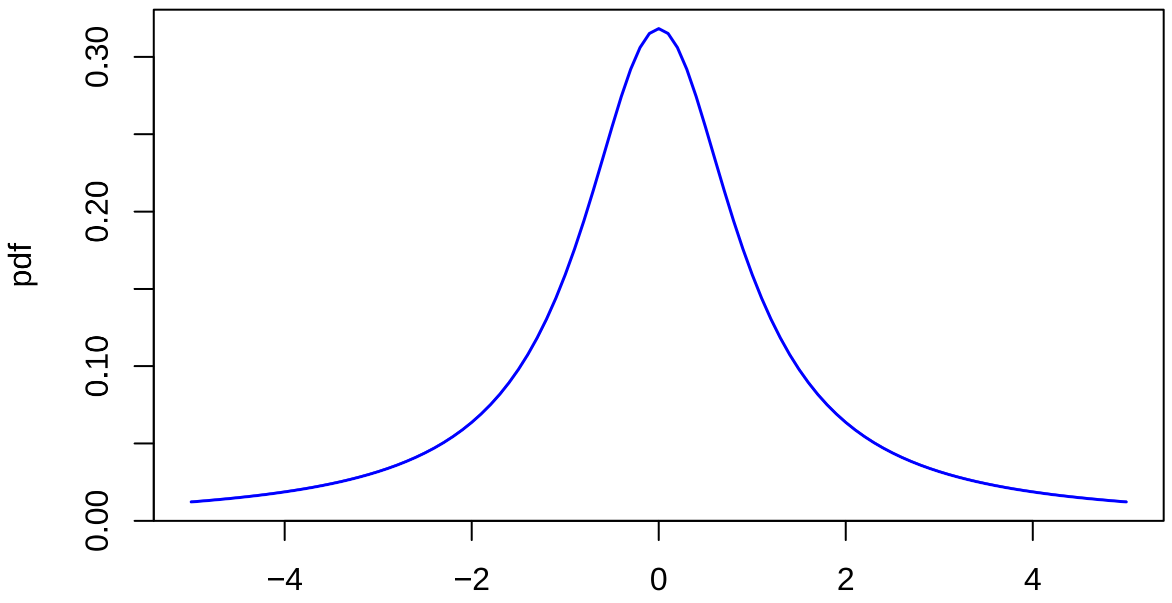
```r
# generate n=10000 independent observations
# from a chi-square distribution with
# 5 degrees of freedom

df <- 5 # degrees of freedom
n <- 10000 # sample size
sample <- c()
for (i in 1:n){
  x <- rnorm(df)
  y <- sum(x^2)
  sample <- c(sample, y)
}

# compare the histogram of the generated observations
# with the density function of the chi-square distribution

hist(sample, col="lightgray", freq=FALSE, main = "sample from chi-squared distribution")
curve(dchisq(x, df), add=TRUE, lwd=2, col="blue")
```



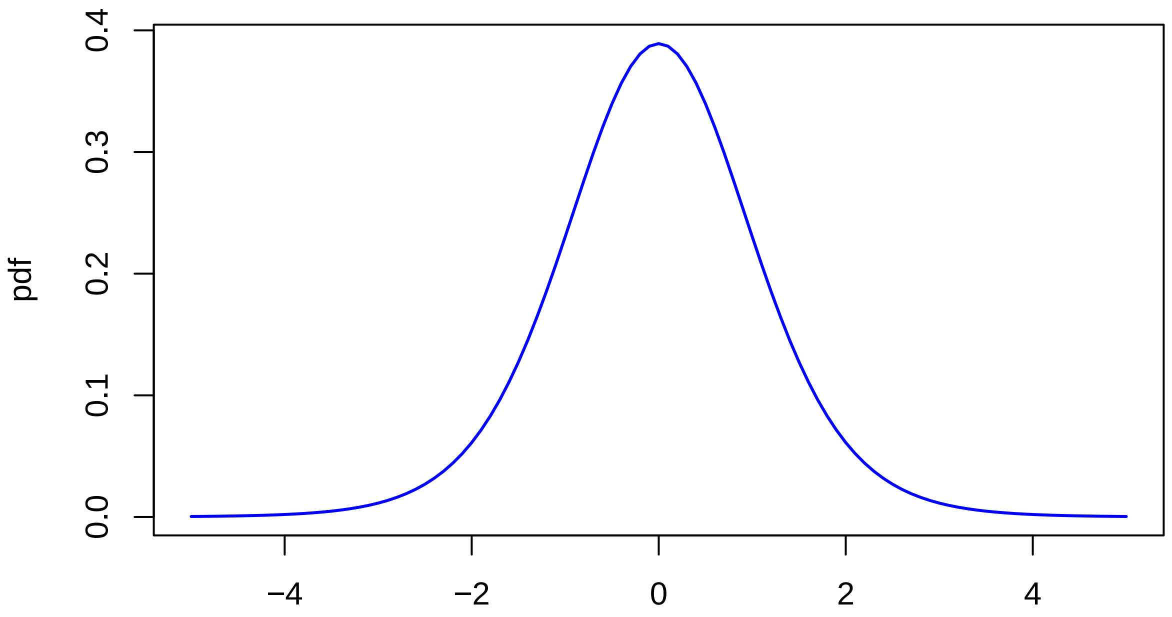
```r
# equivalently using the rchisq() function
sample <- rchisq(n, df)
hist(sample, col="lightgray", freq=FALSE, main = "sample from chi-squared distribution")
curve(dchisq(x, df), add=TRUE, lwd=2, col="blue")
```



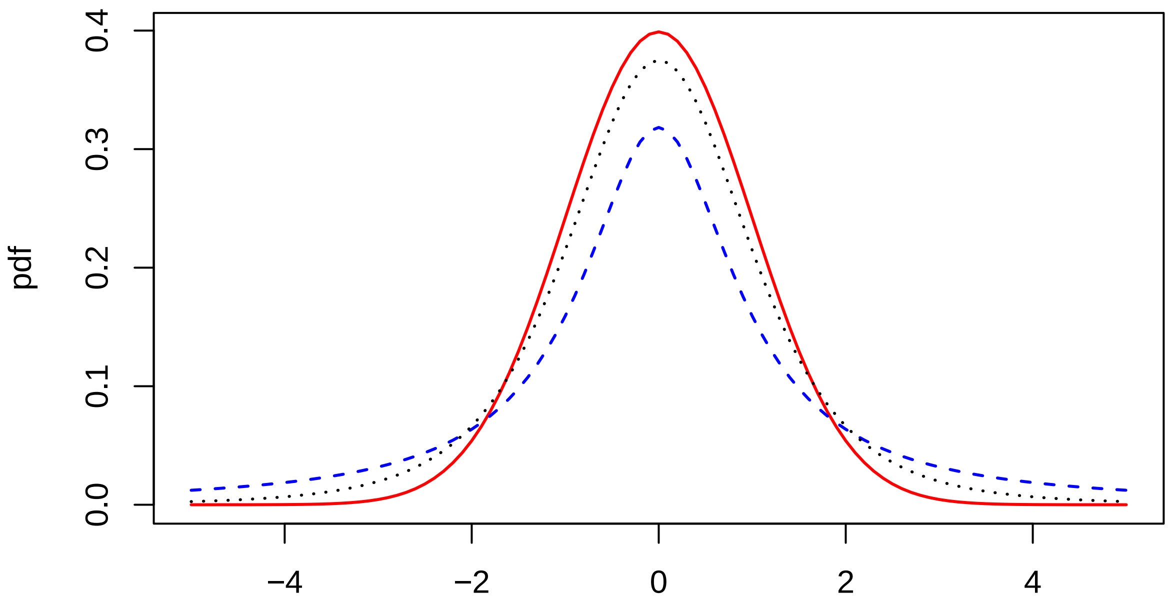
```
# student's t distribution

# density function

# 1 degree of freedom
curve(dt(x, 1),  xlim=c(-5, 5), ylab="pdf", xlab="", col="blue", lwd=1.5)
```



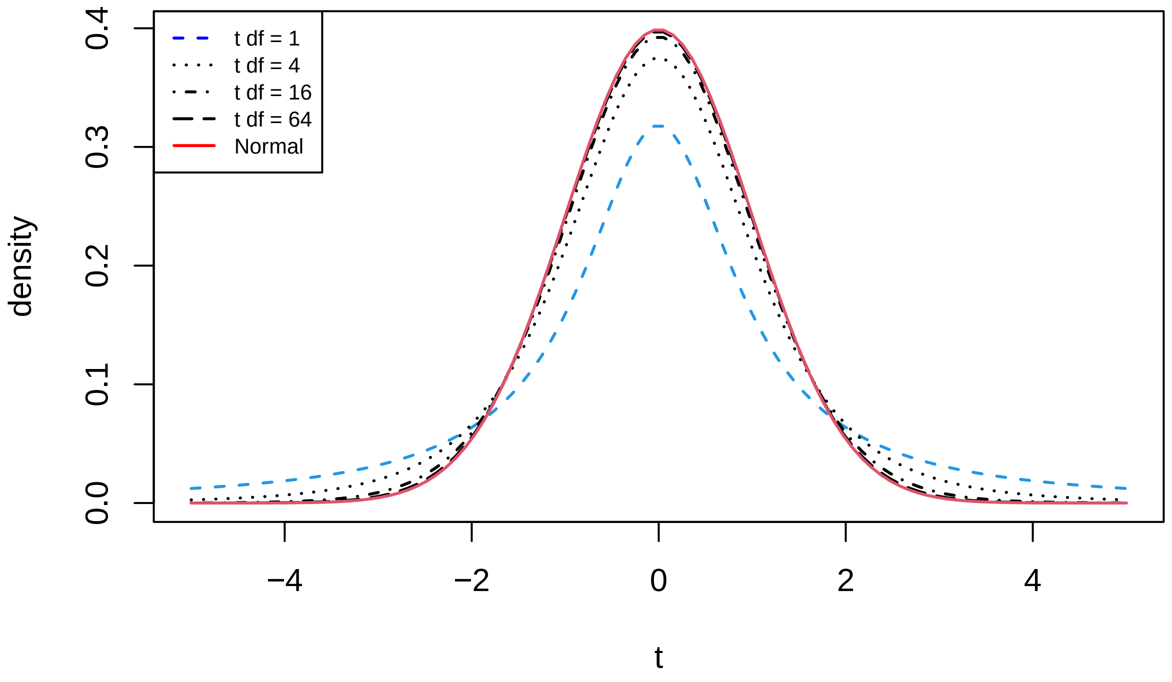
```
# 10 degrees of freedom
curve(dt(x, 10),  xlim=c(-5, 5), ylab="pdf", xlab="", col="blue", lwd=1.5)
```



```
# comparison with the normal distribution

curve(dnorm, xlim=c(-5, 5), ylab="pdf", xlab="", col="red", lwd=1.5)
curve(dt(x, 1), add=TRUE, lty=2, col="blue", lwd=1.5)
curve(dt(x, 4), add=TRUE, lty=3, col="black", lwd=1.5)
```

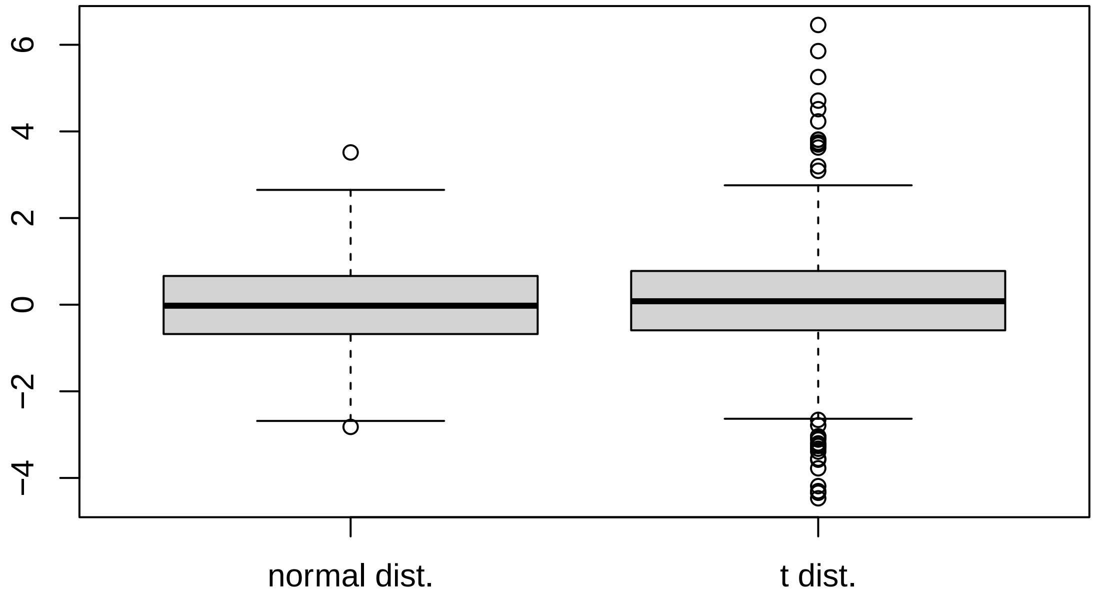


```
# compare densities
x <- seq(-5, 5, length=100)
Y <- cbind(dt(x,1), dt(x,4), dt(x,16), dt(x,64), dnorm(x))
matplot (x, Y, type="l", ylab="density", xlab="t", lty=c(2:5,1), col=c(4,1,1,1,2), lwd=1.5)

# add a legend
leg.label <- c("t df = 1", "t df = 4", "t df = 16", "t df = 64", "Normal" )
leg.lty   <- c(2, 3, 4, 5, 1)
leg.col <- c("blue", "black", "black", "black", "red")
# or equivalently: leg.col <- c(4,1,1,1,2)
legend (x="topleft", y=NULL, legend=leg.label, lty=leg.lty, col=leg.col, lwd=1.5, cex=.7)
```



```
# compare boxplots
x <- rnorm(1000)
```

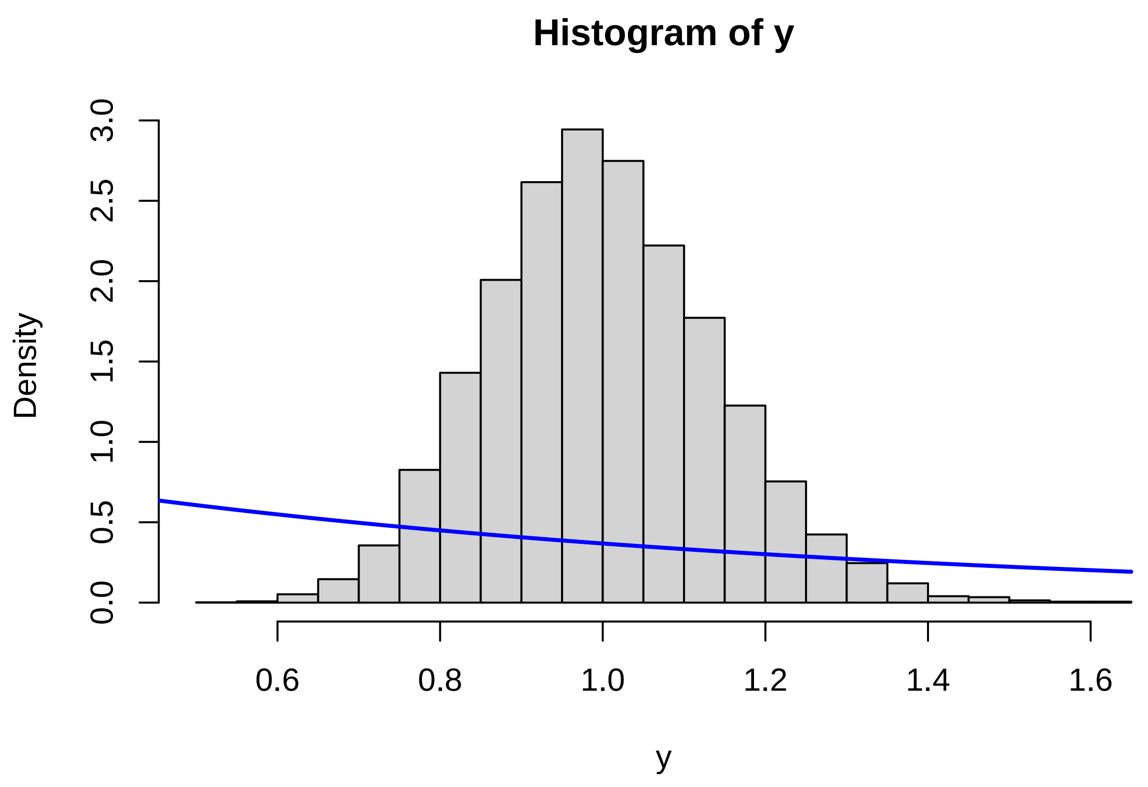
```
y <- rt(1000, 5)

boxplot(x, y, names=c("normal dist.", "t dist."), col="lightgray")
```



```
##########################################################
# Linear conbination of rv's and the central limit theorem
##########################################################


# linear combination of iid rv's with exponential distributions

# X is exp(1)
# Y is the mean of n exp(1) iid rv's

n <- 3
y <- mean(rexp(n, rate=1)) # rate is lambda
y
```

```
## [1] 2.256943
```

```
# generate 10000 values from Y for different values of n
# and compare the distribution of Y with that of X
n <- 3
#n <- 10
n <- 50

y <- c()
for(i in 1:10000) y <- c(y, mean(rexp(n, 1)))

hist(y, breaks=20, freq=FALSE)
curve(dexp(x, 1), from=0.001, add=TRUE, col="blue", lwd=2)
```



```
# check empirically that E(Y)=E(X) where E(X)=1

mean(y)
```

```
## [1] 1.00019
```

```
# check empirically that Var(Y)=Var(X)/n where Var(X)=1

1/n
```

```
## [1] 0.02
```

```
var(y)
```

```
## [1] 0.01991046
```

```
# linear combination of iid normally distributed rv's


# X is N(1, 1)
# Y is the mean of n N(1,1) iid rv's


# generate 10000 values from Y for n=4
# and compare the distribution of Y with that of X

n <- 4

y <- c()
for(i in 1:10000) y <- c(y, mean(rnorm(n, mean=1, sd=1)))

hist(y, breaks=20, freq=FALSE, main="", xlim=c(-1.5, 3.5))
```

```
# normal density with mean=1 and variance=1
curve(dnorm(x, mean=1, sd=1), add=TRUE, col="red", lwd=1.5)


# check empirically that E(Y)=E(X) where E(X)=1

mean(y)
```

```
## [1] 0.9977283
```

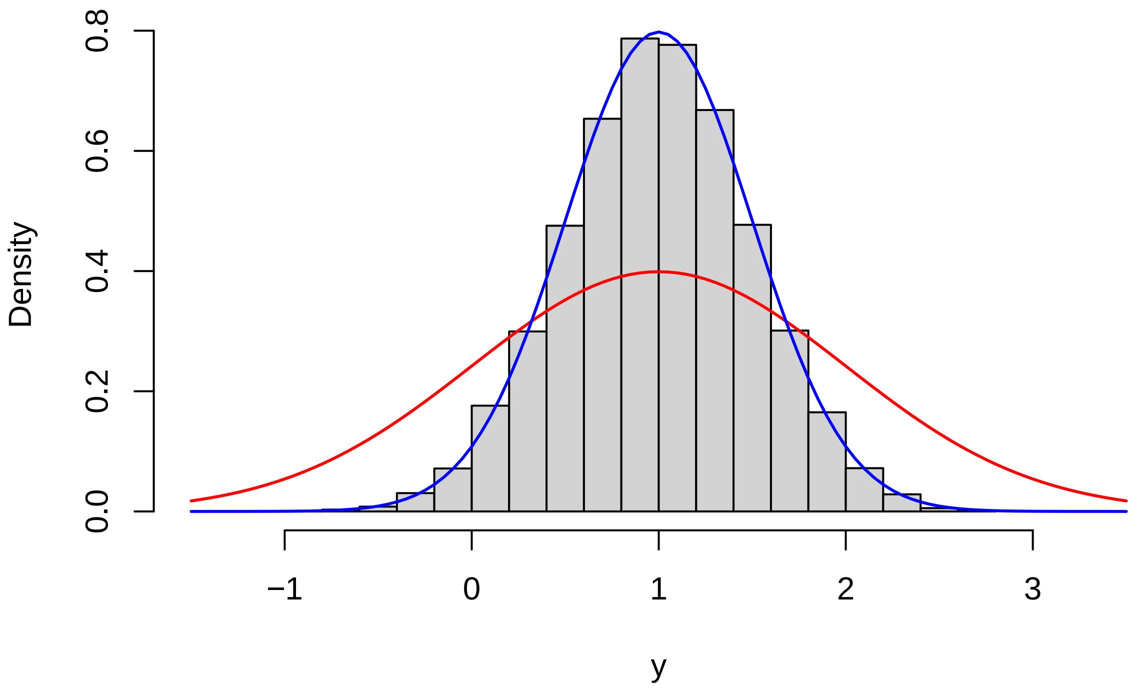
```
# check empirically that Var(Y)=Var(X)/n where Var(X)=1

1/n
```

```
## [1] 0.25
```

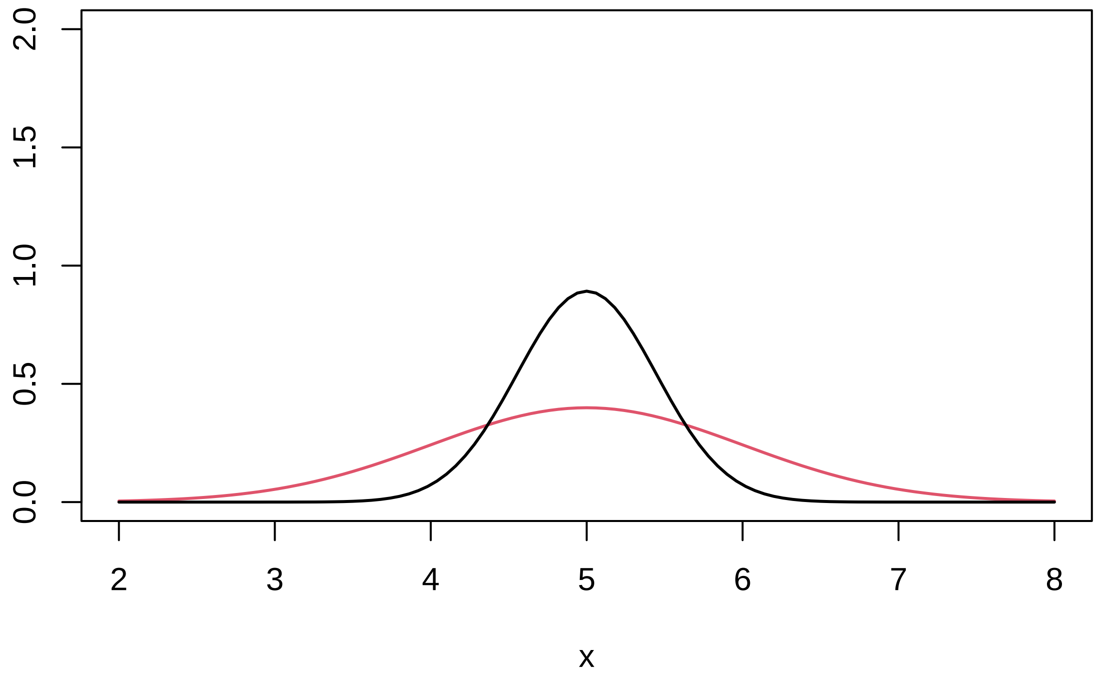
```
var(y)
```

```
## [1] 0.2496358
```

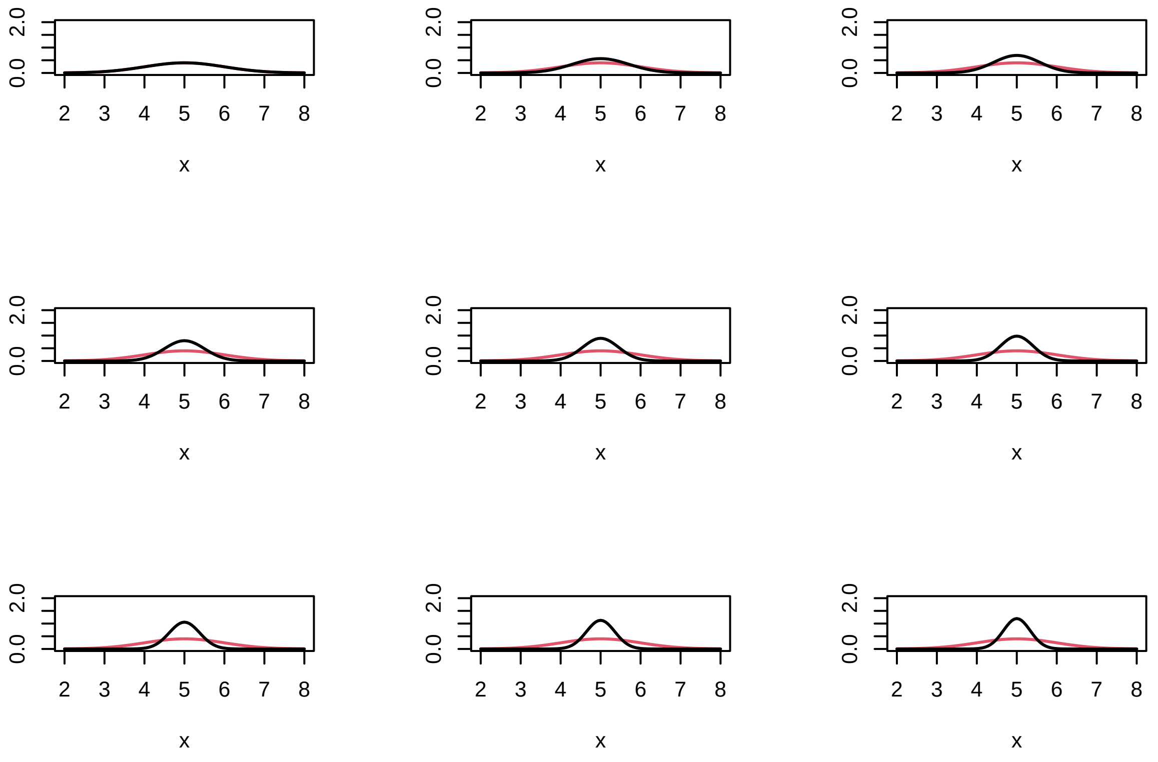
```
# normal density with mean=1 and variance=1/n so that sd=1/sqrt(n)
curve(dnorm(x, mean=1, sd=1/sqrt(n)), add=TRUE, col="blue", lwd=1.5)
```



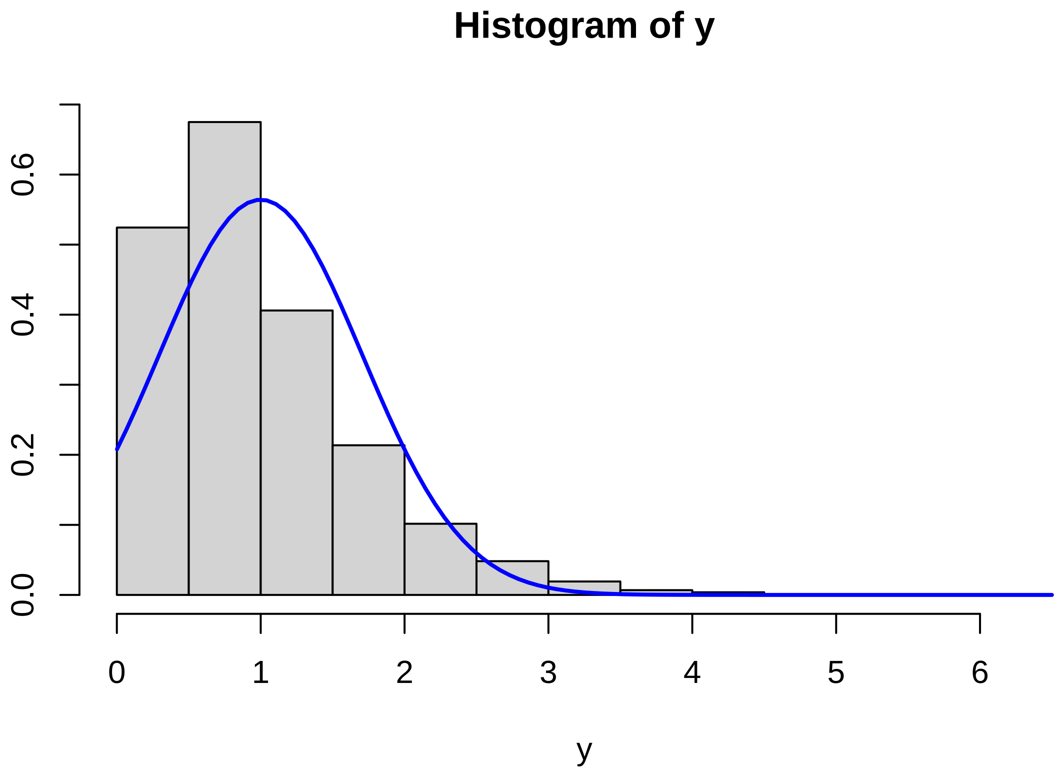
```
# distribution of the mean for different sample sizes

samplingdist <- function(n) {
  curve(dnorm(x,5,1), xlim=c(2,8), ylim=c(0,2), col=2, lwd=1.5, ylab="")
  curve(dnorm(x,5,1/sqrt(n)), xlim=c(2,8), ylim=c(0,2), lwd=1.5,add=TRUE)
  Sys.sleep(0.2)}

# apply for a single value of n
samplingdist(5)
```



```r
# apply for 9 values of n
par(mfrow=c(3,3))
ignore <- sapply(1:9, samplingdist)
```



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

# Central Limit Theorem

# empirical check with exponential distribution
# X is exp(1)
# Y is the mean of n exp(1) iid rv's

# generate 10000 values from Y for different values of n
# and compare the distribution of Y with that
# of the asymptotic normal distribution
n <- 2
#n <- 6
#n <- 10
#n <- 50
#n <- 100

y <- c()
for(i in 1:10000) y <- c(y, mean(rexp(n, 1)))

hist(y, breaks=20, freq=FALSE, ylab="")

mu.y <- 1
sigma.y <- 1/sqrt(n)
curve(dnorm(x, mean=mu.y, sd=sigma.y), from=0.001, add=TRUE, col="blue", lwd=2)
```


Histogram of y
y