Hypothesis Testing

Gordon Erlebacher

What we have done

- R basics:
  - vectors, data frames,
  - factors, extraction,
  - logical expressions, scripts, read and writing data files
  - histograms, plotting

Functions used

- c(), data.frame(), as.factor()
- seq(), extraction functions ([ ] and [,])
- read.csv, read.table (there is also write.csv and write.table)
- mean(), var(), rnorm()
- hist(), plot()
- source()
- .... and some others ...

What is next?

- Sample versus population through R
- Discussion of distributions, and use of
  rnorm, dnorm, qnorm, pnorm
  and similar functions for other distributions
- Hypothesis testing
  - use of t.test for H0/H1 hypothesis
  - use of shapiro.test for normality test
  - test for normality via plotting

Population

A statistical population is a set of entities concerning which statistical inferences are to be drawn...

WIKIPEDIA
The Free Encyclopedia

Population

Students in ISC4244C in the fall of 2012.

Males residing within the city limits of Tallahassee.

Floridians with an income greater than $100,000.

Whooping cranes (n~437 in North America).

19th Century British petty criminals.

Good Cambridge men.

Population

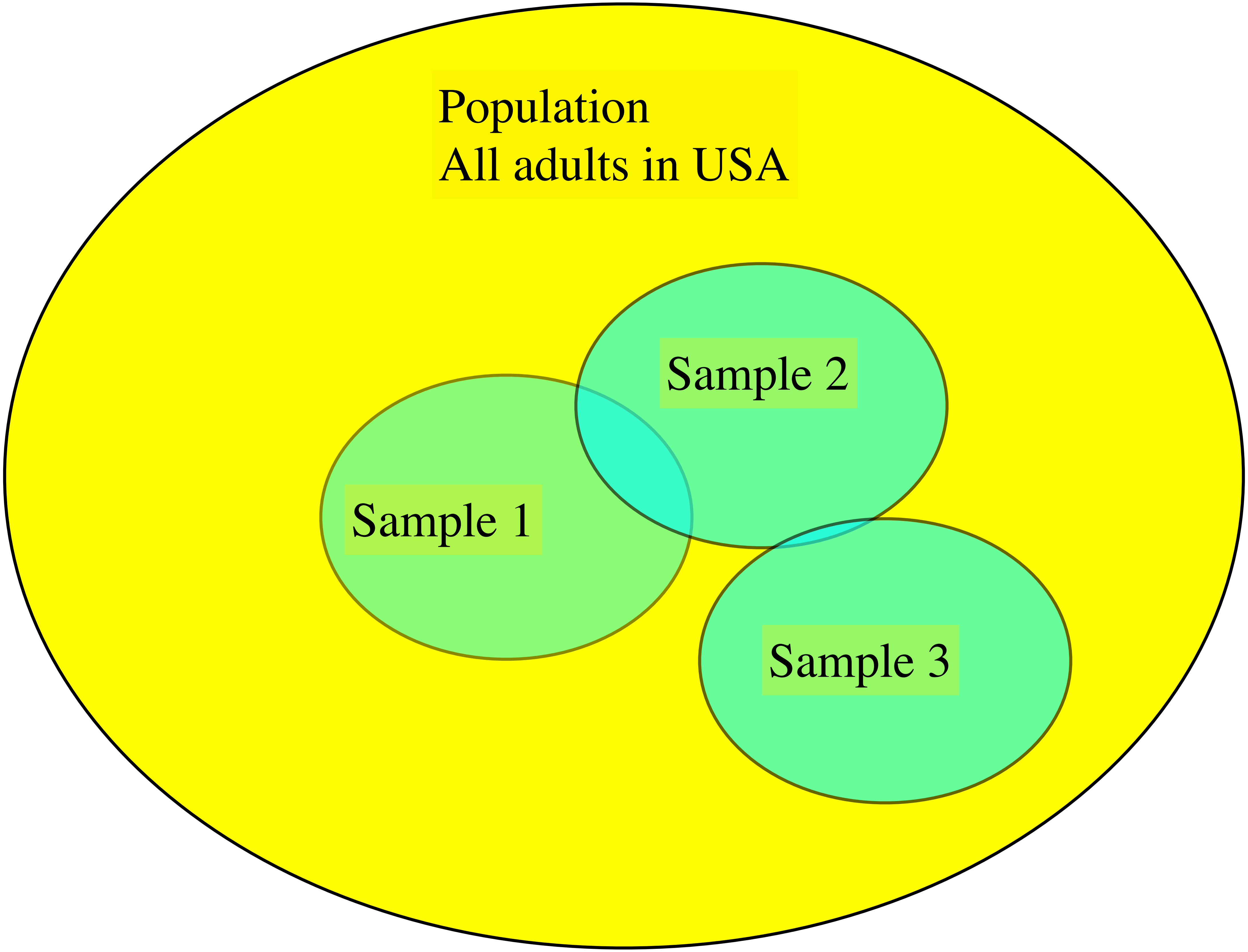
Consider the population of all adults between 30 and 50.

We are interested in the average height of this population.

We cannot practically measure everyone's height. What to do?

Population vs Sample

- Consider the population of all adults between 30 and 50
- We are interested in the average height of this population
- Since we cannot ask everybody their height, we identify a sample from the entire population, and compute the average height of this sample
- The average sample height is clearly an approximation to the average height of the population
    - Take a different sample, and one gets a different sample average height
    - The average sample height changes from sample to sample.
    - The average sample variance is also a function of the particular sample


Population
All adults in USA
Sample 2
Sample 1
Sample 3

Population vs Sample

- Population has **N** individuals
- A sample has **n** individuals
  - **n** << **N**
- 300 million people in the USA
  - perhaps 100 million adults who can vote
- Pick a sample of 5,000 individuals
  - 5,000 << 100 million

Work with Variables

A variable is the quantity or quality of members of the population about which we are interested. e.g., sex, opinion, height, number of employers held in the past ten years.

To be clear, we talk about individual variates, which are the observed values of variables, e.g., female, strongly agree, 1.7m, 3.

Type of Variables

Attributes – male vs. female, criminal vs. Cambridge student (also called a **factor** or **category**)

Ranked variables – strongly disagree, disagree, neutral, agree, strongly agree (**ordered factor**)

Measurement variables (**numerics, with decimal)**)

Discontinuous (meristic) – number of employers within the past ten years (**integers**)

Continuous (metric) – height (can take any value)

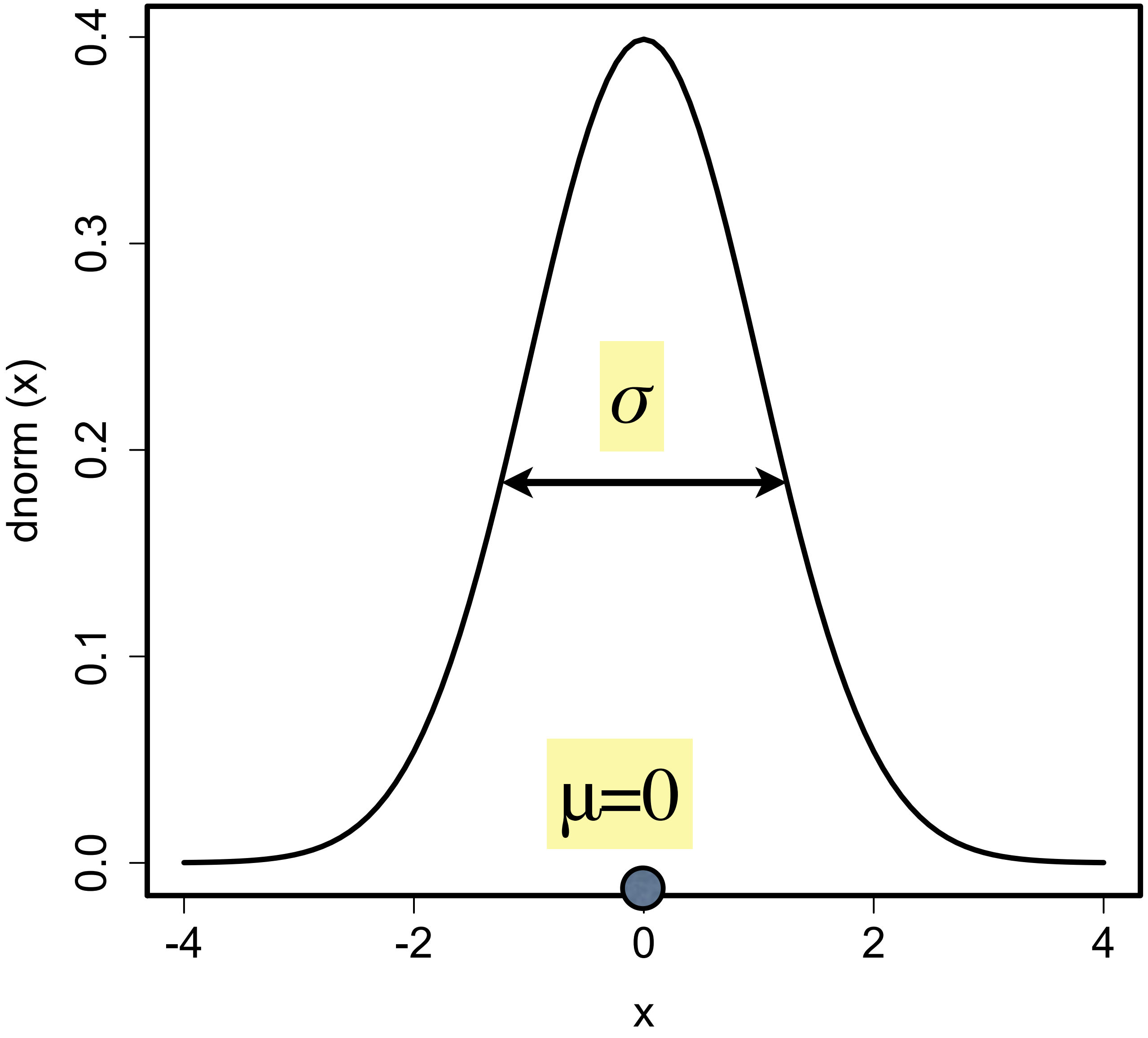
Random Variable

- Let the variable **x** be a random pick from the population
- **x** is called a random variable
  - its value is a random pick from the population of all adults with age between 30 and 50
- A variable either has a definite value or is a random pick from some population
- Let us return to our experiment

Create a population in R

- Assume that the distribution of heights in our population is a normal distribution with mean μ and variance σ^2
- We also write (see your course on statistics)
  - μ = E(*X*)
  - σ^2 = var(*X*)
- We say that $X \in N(\mu, \sigma^2)$

Normal Distribution



ps: point size
lwd: line width

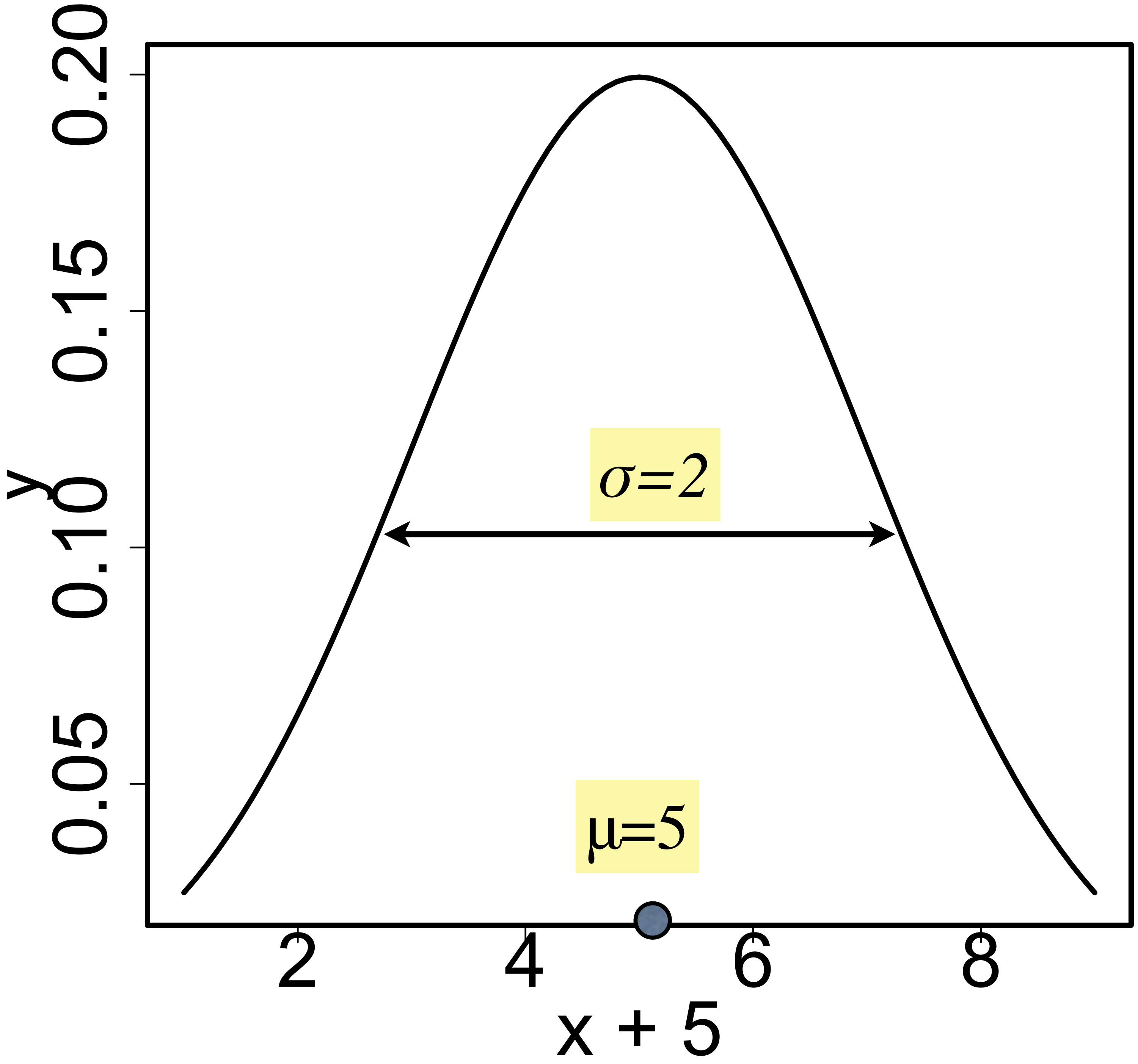
```
> par(ps=18, lwd=3)
> plot(dnorm, from=-4, to= 4)
```

N(0,1)

$\sigma = 1$

μ=0

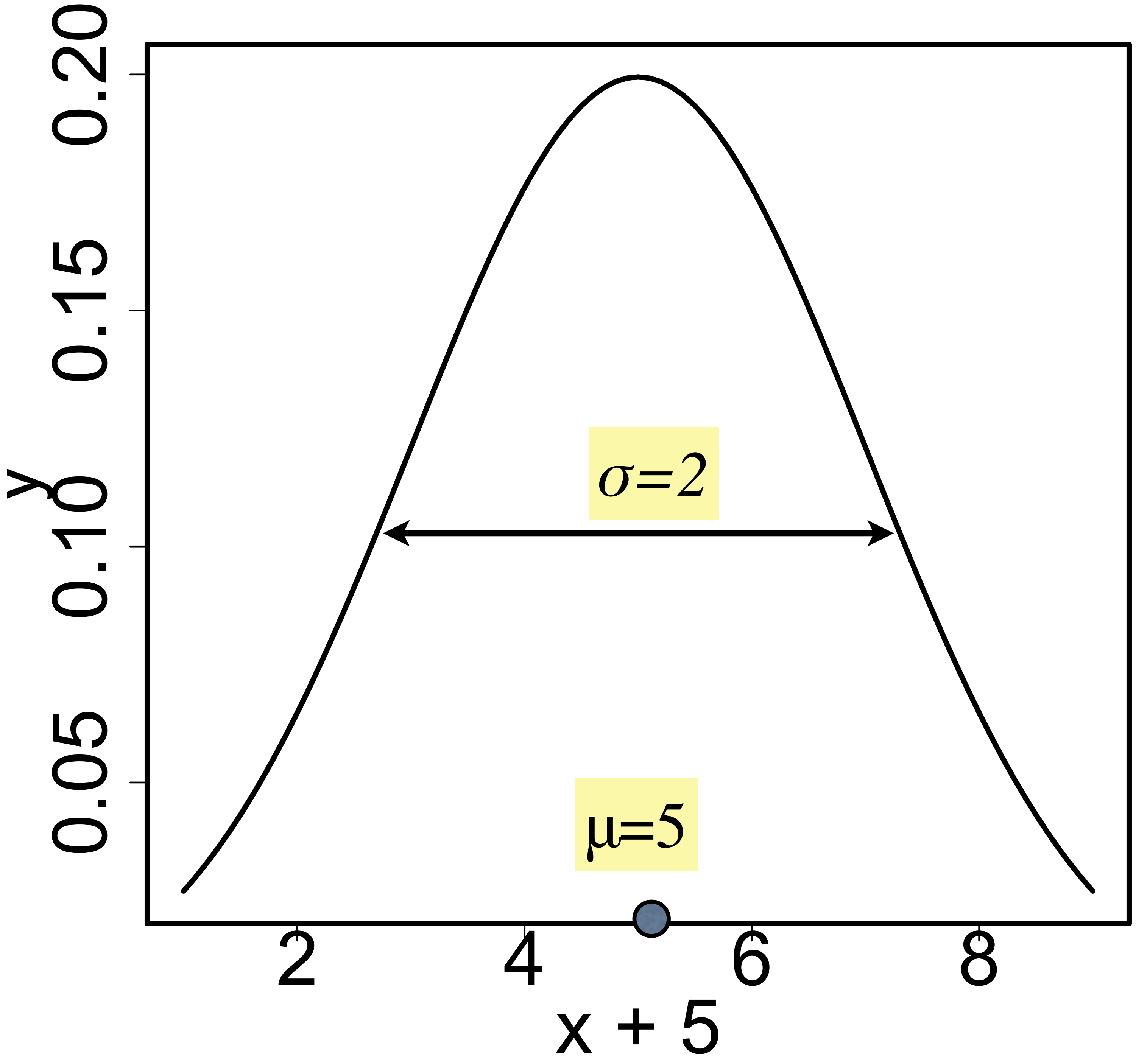
N(5,2)



```
> par(lwd=3)
> x = seq(-4,4,.1) + 5
> y = dnorm(x+5, mean=5, sd=2)
> plot(x,y, type='l')
```

line plot

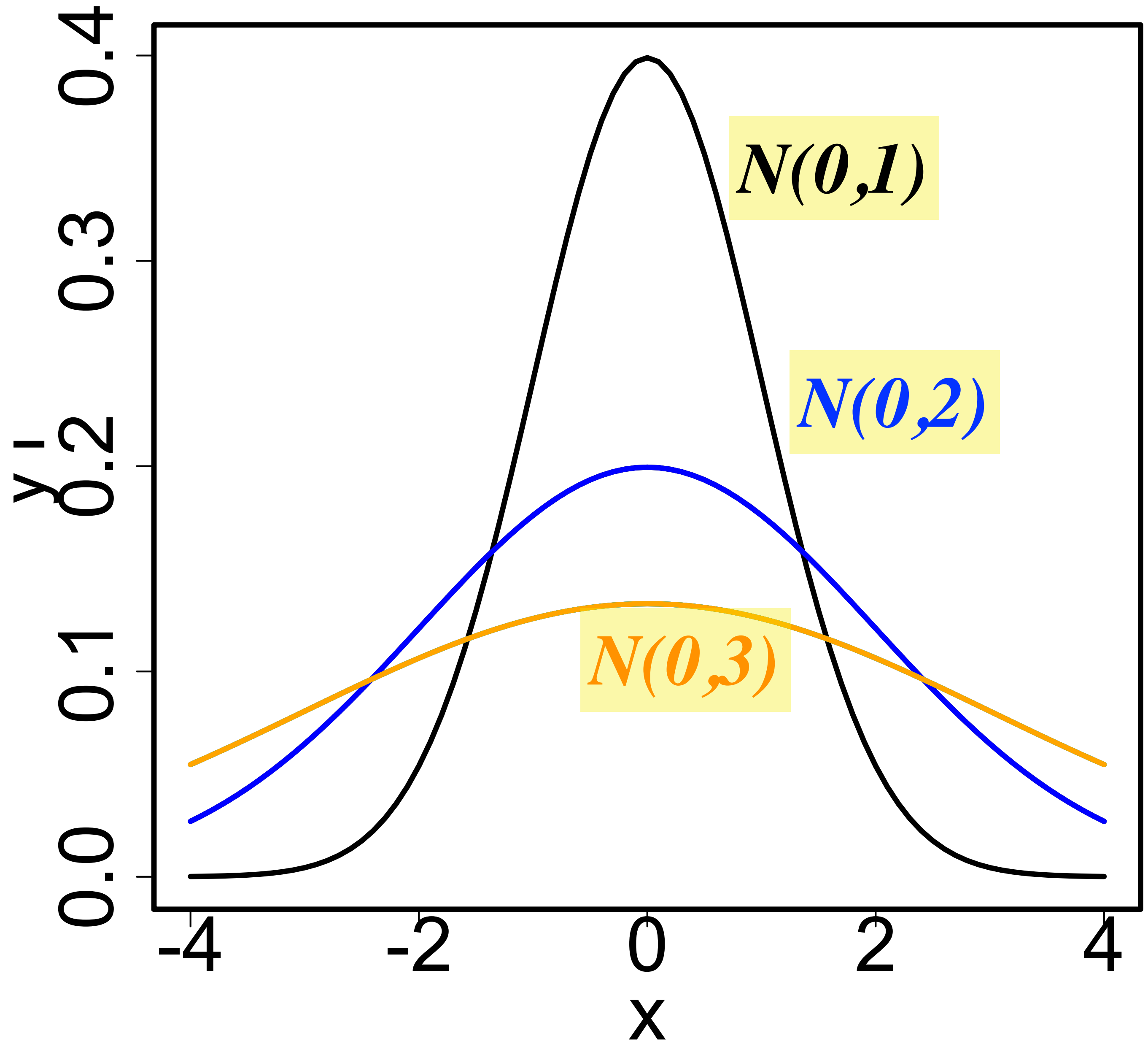
Alternative



dnorm(2.3, mean=5) returns the value of the normal distribution when x = 2.3

```
> par(lwd=3)
> plot(dnorm)
# plots dnorm at a sequence
# of points chosen by R
```

Multiple lines on same plot



```
> x = seq(-4,4,.1)
> y1 = dnorm(x,sd=1)
> y2 = dnorm(x,sd=2)
> y3 = dnorm(x,sd=3)
> plot(x,y1,type='l')
> lines(x,y2,type='l',col='blue')
```

The plotting commands do not include the labels in the plot

Random Variable

- Assume that the height of the adult population (age 30-50) satisfy a normal distribution
  - this is an approximation since a normal distribution is continuous
  - only as the population file becomes infinite can a normal population actually hold
- Assume
  - a mean=5’7” = 67” (inches)
  - a standard deviation=12” (inches)
- We need to choose a random adult from this population
  - **rnorm(1, mean=67, sd=12)**
- Pick 100 elements from this population
  - **rnorm(100, mean=67, sd=12)**

Theoretical Population

An imagined population about which everything is known.

In particular, we know or assert the parameters that describe its statistical distribution.

These parameters are usually designated by greek letters to distinguish them from estimates we will make later on, e.g.,

μ=E(x) σ^2=var(X)

Functions related to the normal distribution

dnorm(x, mean = 0, sd = 1, ...)	Density
pnorm(q, mean = 0, sd = 1, ...)	Cumulative
qnorm(p, mean = 0, sd = 1, ...)	Quantile
rnorm(n, mean = 0, sd = 1)	Random value from a normal distribution

We will discuss these over the next few lessons

Functions related to the uniform distribution

dunif(x, mean = 0, sd = 1, ...)	Density
punif(q, mean = 0, sd = 1, ...)	Cumulative
qunif(p, mean = 0, sd = 1, ...)	Quantile
runif(n, mean = 0, sd = 1)	Random value from a normal distribution

We will discuss these over the next few lessons

A Random Sample

A random sample is a presumed representative subset of a population that will be used to draw conclusions about the parent population.

Sample

- We now have a population
  - we replace millions of adults by a distribution that assumes an infinity of adults (much much greater than the sample size)
- Now consider a sample of size 1000

  ```
  > sampl = rnorm(1000, mean=67, sd=12)
  ```

- Average height in this sample:

  ```
  > avg.height = mean(sampl)
  ```

Two Samples

```
> sampl = rnorm(1000, mean=67, sd=12)
> mean(sampl)
[1] 67.1902
> sampl = rnorm(1000, mean=57, sd=12)
> mean(sampl)
[1] 56.63856
> sampl = rnorm(1000, mean=67, sd=12)
> mean(sampl)
[1] 66.86192
```

Each sample has its own mean

Sample Mean

- Each sample has its own mean
- Assume we take 10,000 samples (very large number) and we generate 10,000 different means. **What is the distribution of these means?**
- The theoretical distribution is close to N(67, 12/sqrt(10000)) = N(67, 0.12)
- We now do this experiment in R and plot the results
  - assume a sample size of 1000, and 100 samples

Programming without conditionals and loops

Most programming languages allow expressions such as

```
while (n < 10) {
    do ... something ...
    n = n + 1
}
```

or

```
if (n < 10) {
   do ... something ...
}
```

I would like to avoid these constructs if possible

Use of apply(...)

Description:

Returns a vector or array or list of values obtained by applying a function to margins of an array or matrix.

Usage: apply(X, MARGIN, FUN, ...)

Arguments:

X: an array, including a matrix.

MARGIN: a vector giving the subscripts which the function will be applied over. E.g., for a matrix '1' indicates rows, '2' indicates columns, 'c(1, 2)' indicates rows and columns. Where 'X' has named dimnames, it can be a character vector selecting dimension names.

FUN: the function to be applied: see 'Details'. In the case of functions like '+', '%*%', etc., the function name must be backquoted or quoted.

Apply

- First argument of **apply(..)** is a matrix
- The **FUN** argument is applied to each row of the matrix if **MARGIN=1**, and to each column of the matrix if **MARGIN=2**

Recall: a matrix is essentially a data frame where all columns are of the same type

Example 1

```
> r = matrix(c(1:12), nrow=3)
> r
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
> apply(r,FUN=mean,MARGIN=1)
[1] 5.5 6.5 7.5
> apply(r,FUN=mean,MARGIN=2)
[1]  2  5  8 11
```

3 rows

The function returns a single for each row if margin=1, or one value for each column if margin=2

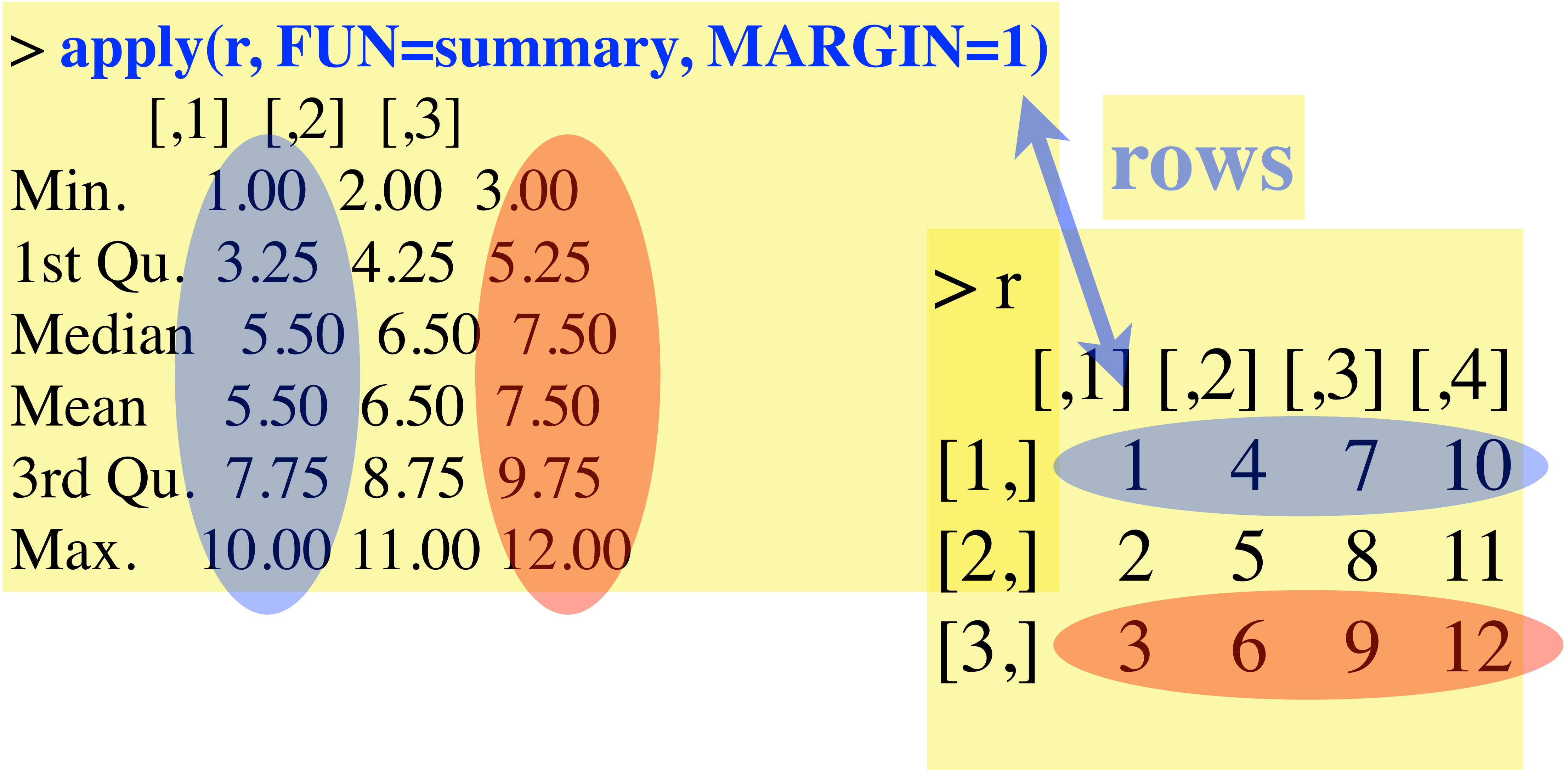
Example 2
(same result as example 1)

```
> r = matrix(c(1:12), ncol=4)
> r
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
> apply(r,FUN=mean,MARGIN=1)
[1] 5.5 6.5 7.5
> apply(r,FUN=mean,MARGIN=2)
[1]  2  5  8 11
```

4 columns

The *apply* function returns a single result for each row if margin=1, or one valuefor each column if margin=2

Example 2

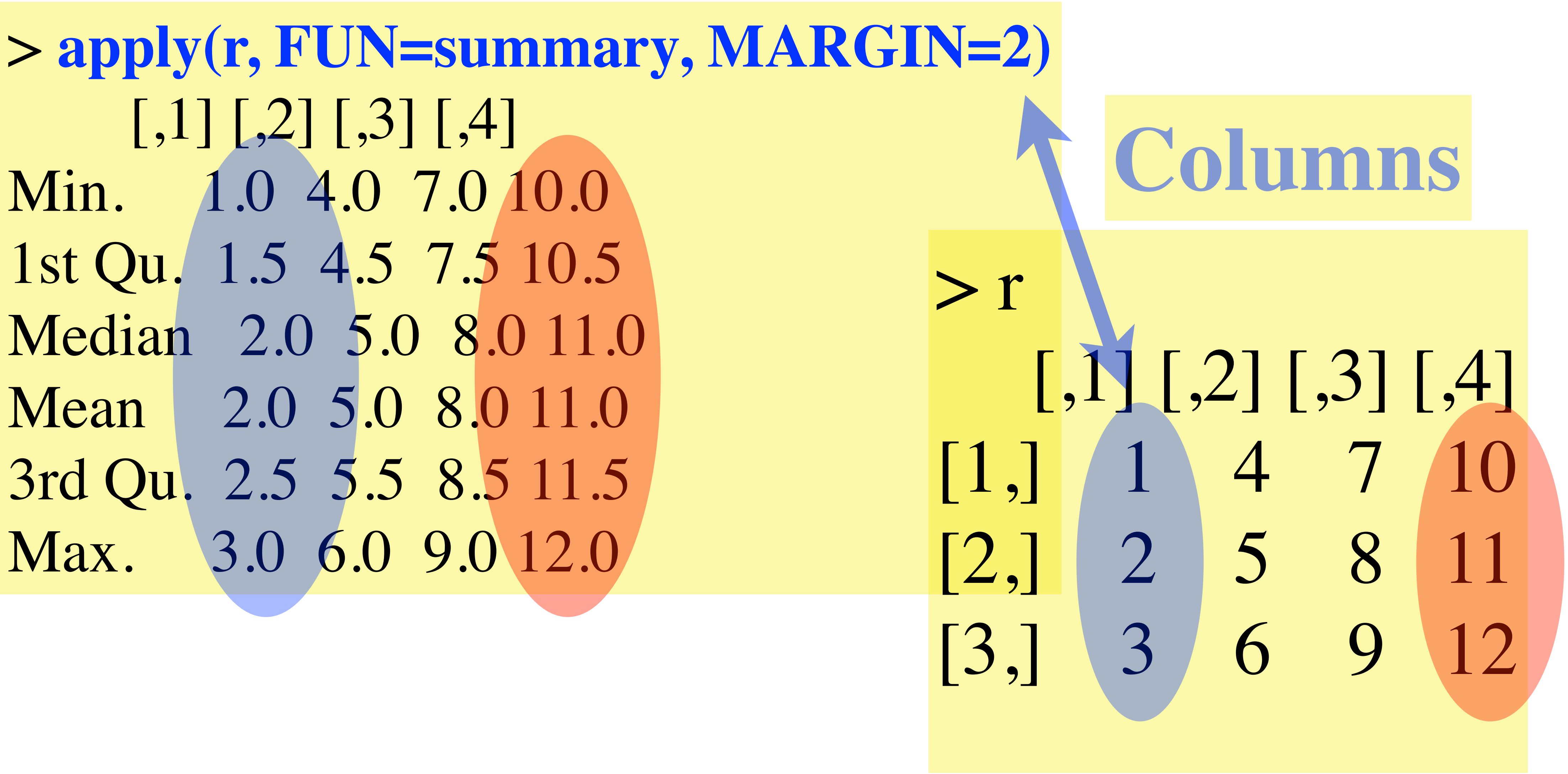


```
> apply(r, FUN=summary, MARGIN=1)
         [,1]  [,2]  [,3]
Min.      1.00  2.00  3.00
1st Qu.  3.25  4.25  5.25
Median    5.50  6.50  7.50
Mean      5.50  6.50  7.50
3rd Qu.  7.75  8.75  9.75
Max.     10.00 11.00 12.00
```

rows

```
> r
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

Example 3



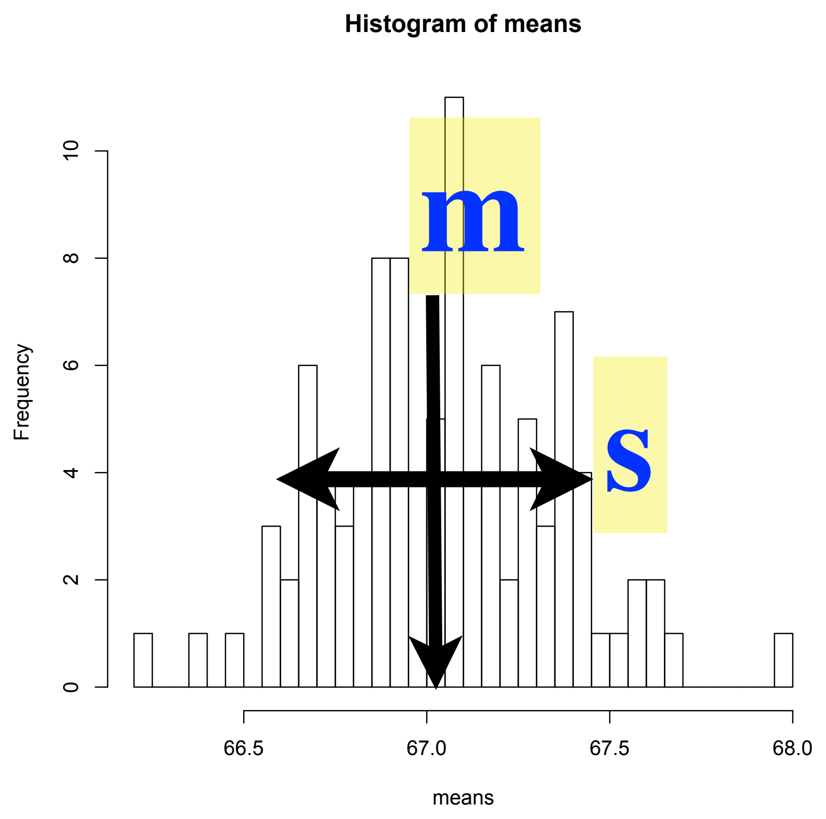
```
> apply(r, FUN=summary, MARGIN=2)
        [,1] [,2] [,3] [,4]
Min.     1.0  4.0  7.0 10.0
1st Qu.  1.5  4.5  7.5 10.5
Median   2.0  5.0  8.0 11.0
Mean     2.0  5.0  8.0 11.0
3rd Qu.  2.5  5.5  8.5 11.5
Max.     3.0  6.0  9.0 12.0
```

Columns

```
> r
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

```
nb.samples = 100
sample.size = 1000
all.samples = rnorm(sample.size*nb.samples,mean=67,sd=12)
mat.samples = matrix(all.samples, ncol=nb.samples)
means = apply(mat.samples, MARGIN=2, FUN=mean)
hist(means,breaks=30)
```

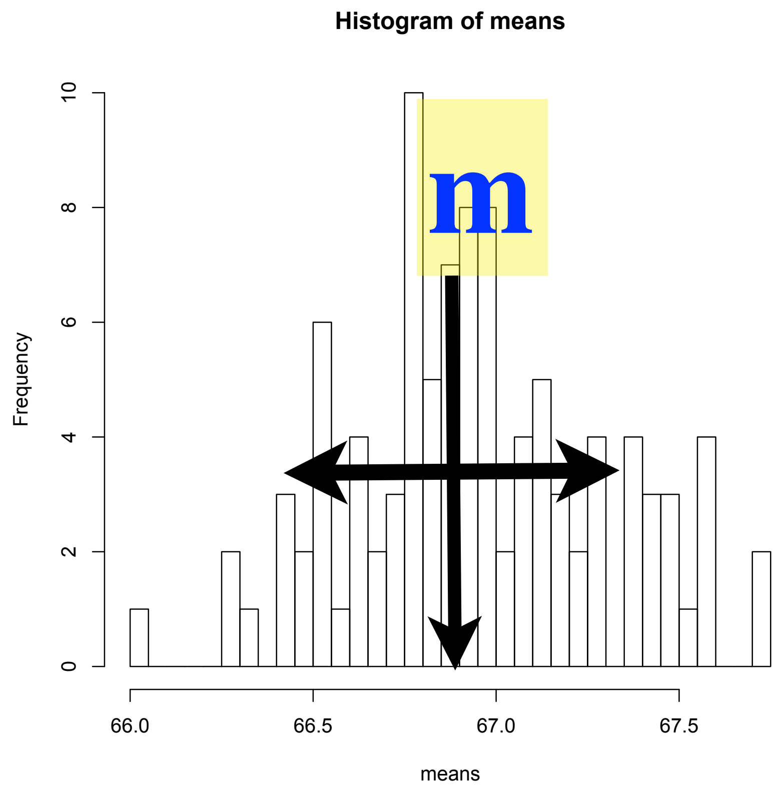
- I ran the above series of commands four times.
- The samples are different each time.
- Their distributions are plotted on the next slide
  - using the **hist(vector)**

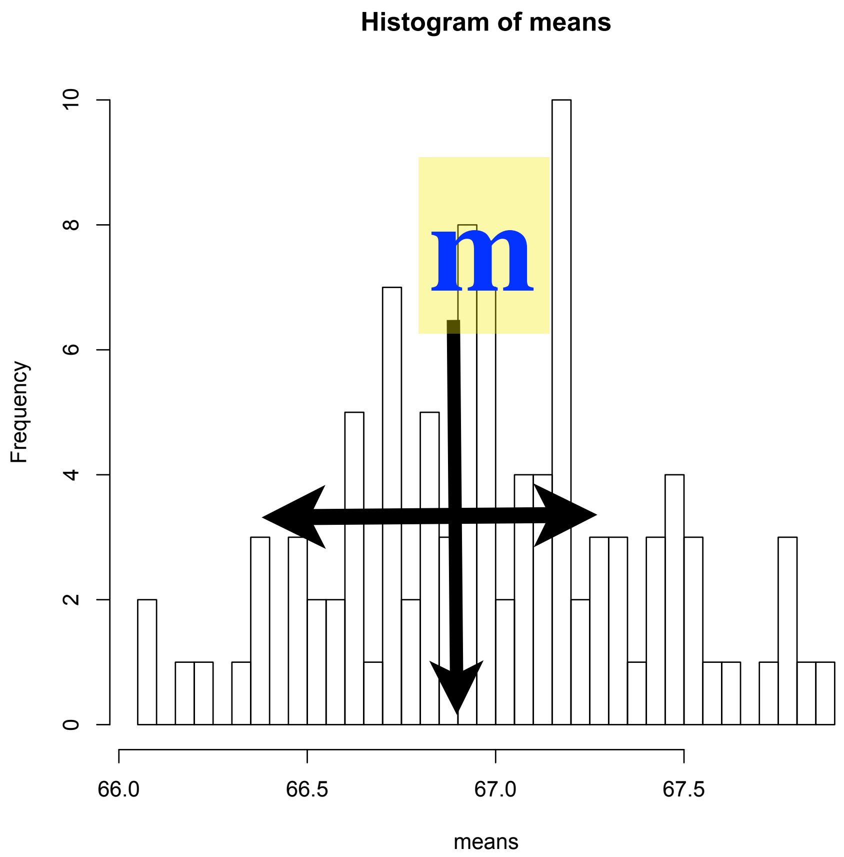


Each plot is individual

I did not use multiple plots per page

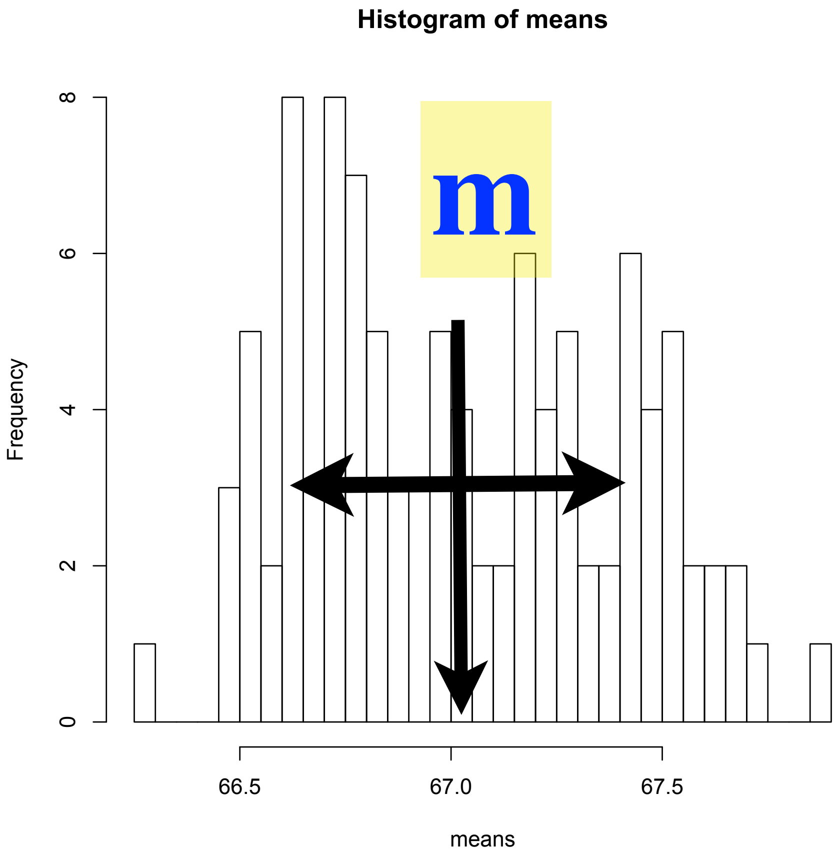
There are 30 breaks for the histogram





m is the mean of the distribution of means

s is the standard deviation of the distribution of means

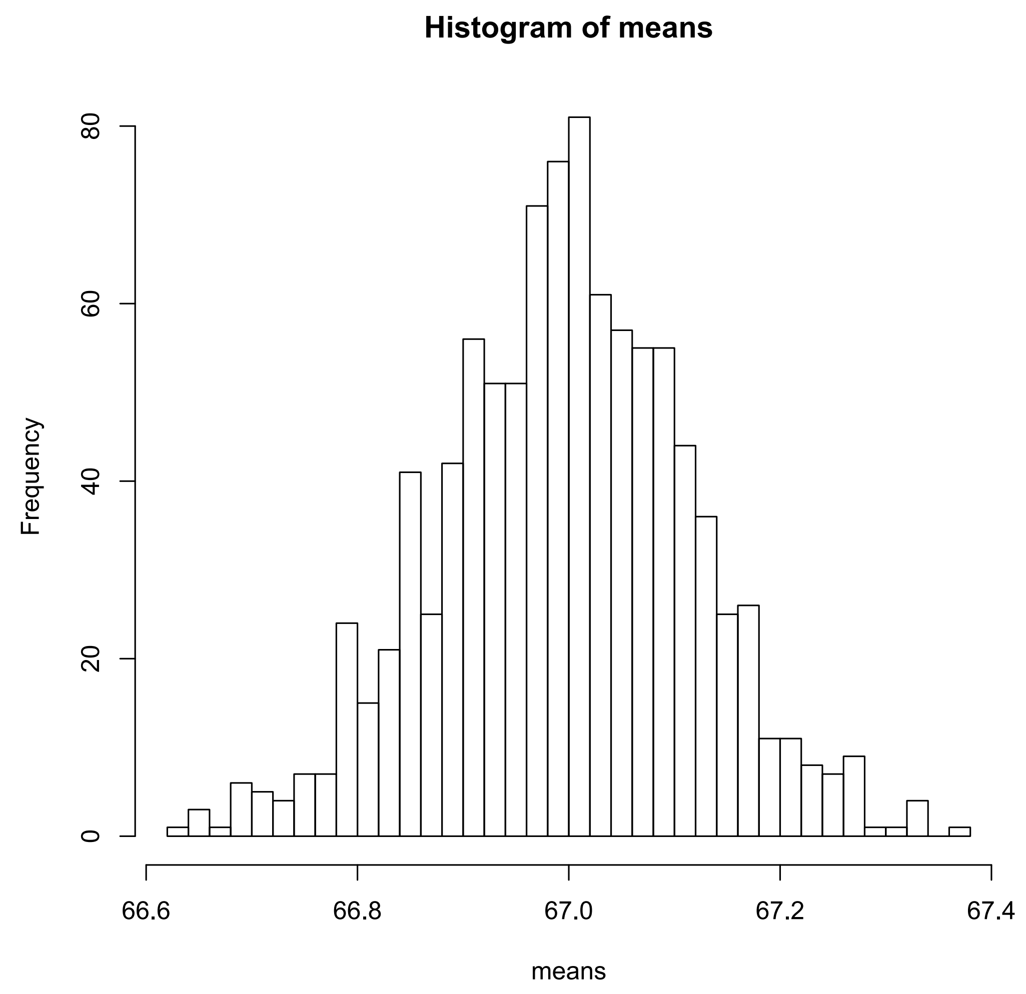


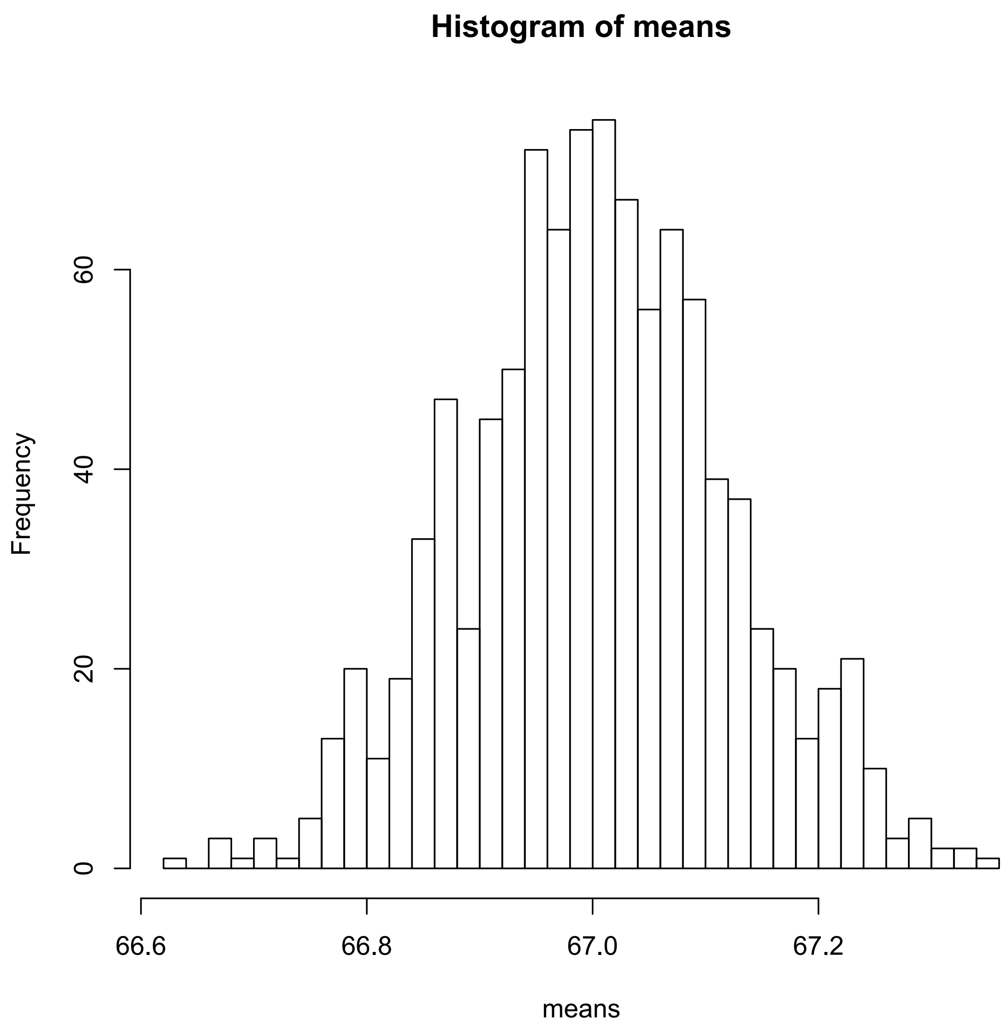
Let us get a smoother plot

nb.samples and sample.size increase by factor of 10

```
nb.samples = 1000
sample.size = 10000
all.samples = rnorm(sample.size*nb.samples,mean=67,sd=12)
mat.samples = matrix(all.samples, ncol=nb.samples)
means = apply(mat.samples,MARGIN=2,FUN=mean)
hist(means,breaks=30)
```

The distributions are closer to normal





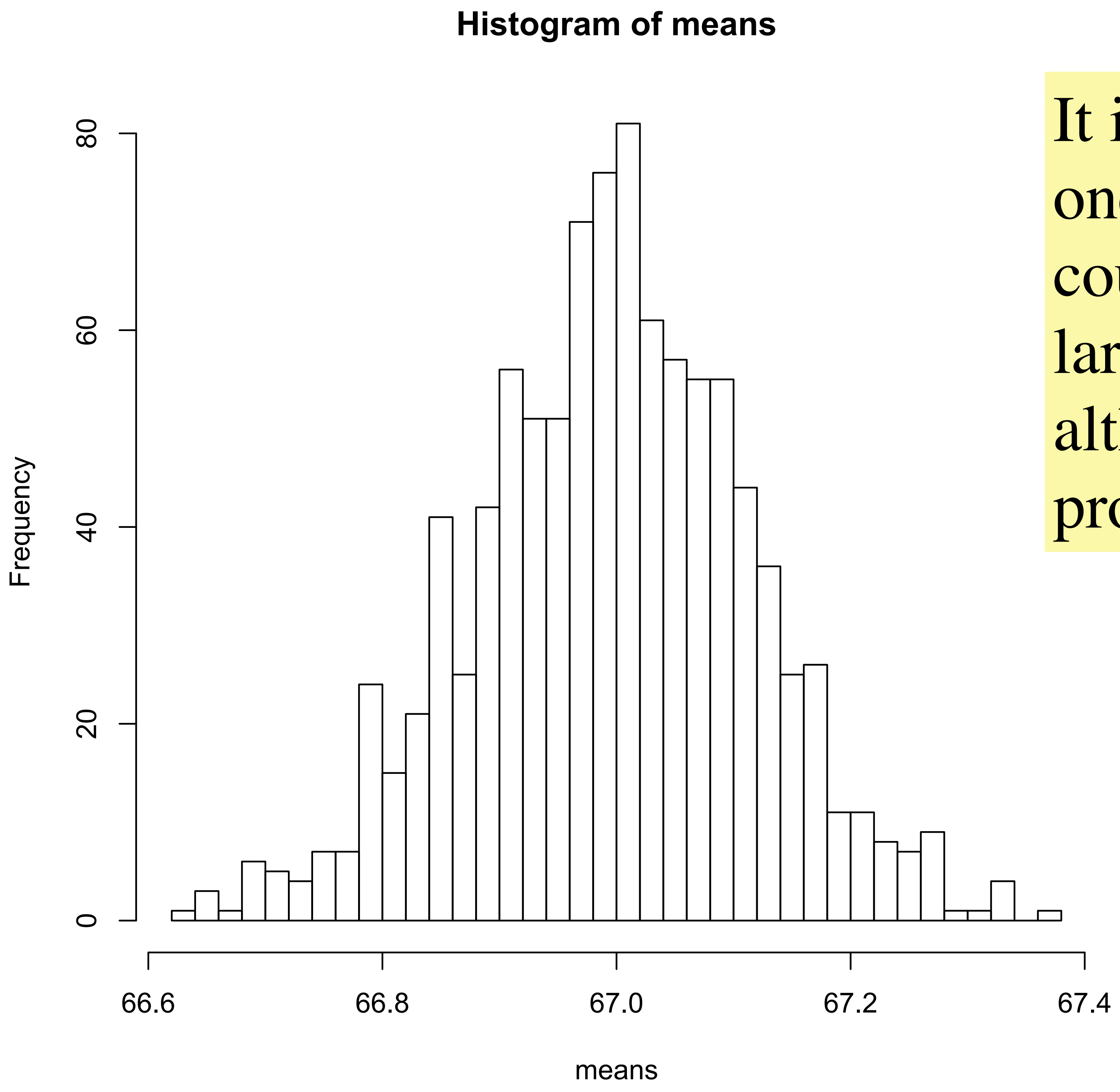
```
> source("height_samples.r")
mean(means)=  67.00137
sd(means)=  0.1159333
```

```
> source("height_samples.r")
mean(means)=  66.99516
sd(means)=  0.123773
```

More Generally

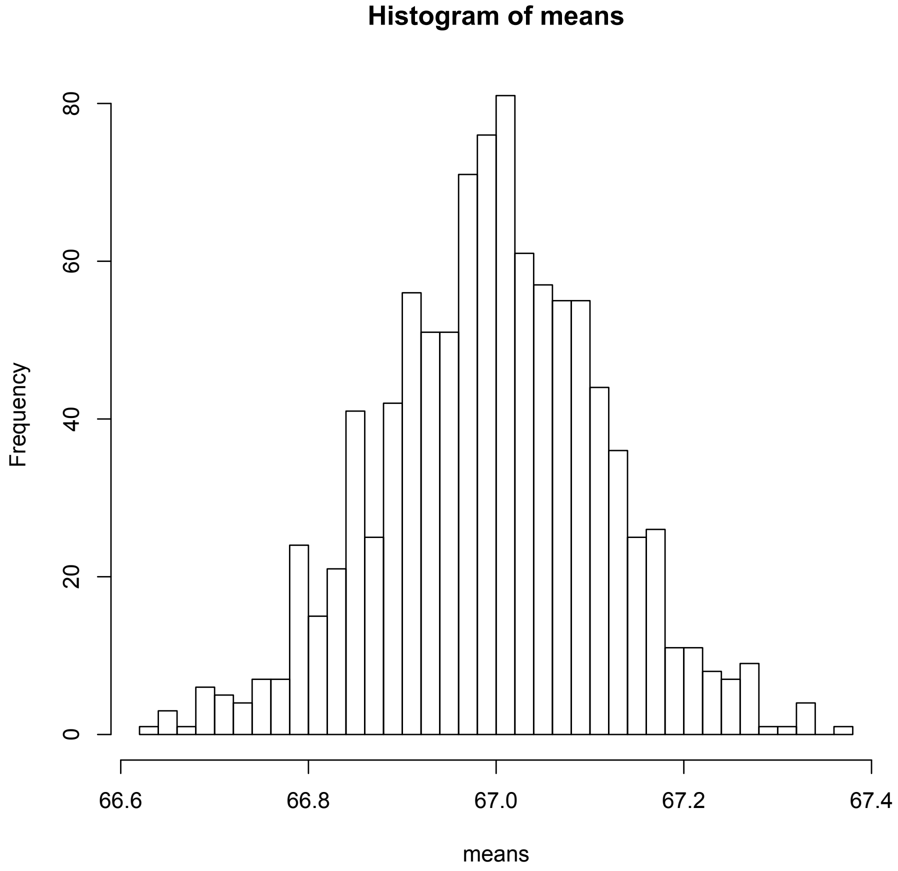
- In the previous example, I postulated that the height distribution in the US population followed a normal distribution
  - But that may not be the case
- It so happens that *whatever* the distribution of the population, the *sample means* will go to a normal distribution as its size becomes larger and larger
  - This is stated more precisely in the **Central Limit Theorem**

- Given a sample of size ***n***,
  - as *n* gets larger, the variance of the distribution of sample means is the population variance divided by *n*
  - with *n=10000*, the standard deviation of the previous example should go to
    12/sqrt(10000) = 0.12
  - with *n=1000*, 12/sqrt(1000) = 0.38 (more spread)



It is possible that
one of the sample means
could be very much
larger than 67,
although with very low
probability

It is possible that one of the sample means could be very much larger than 67, although with very low probability

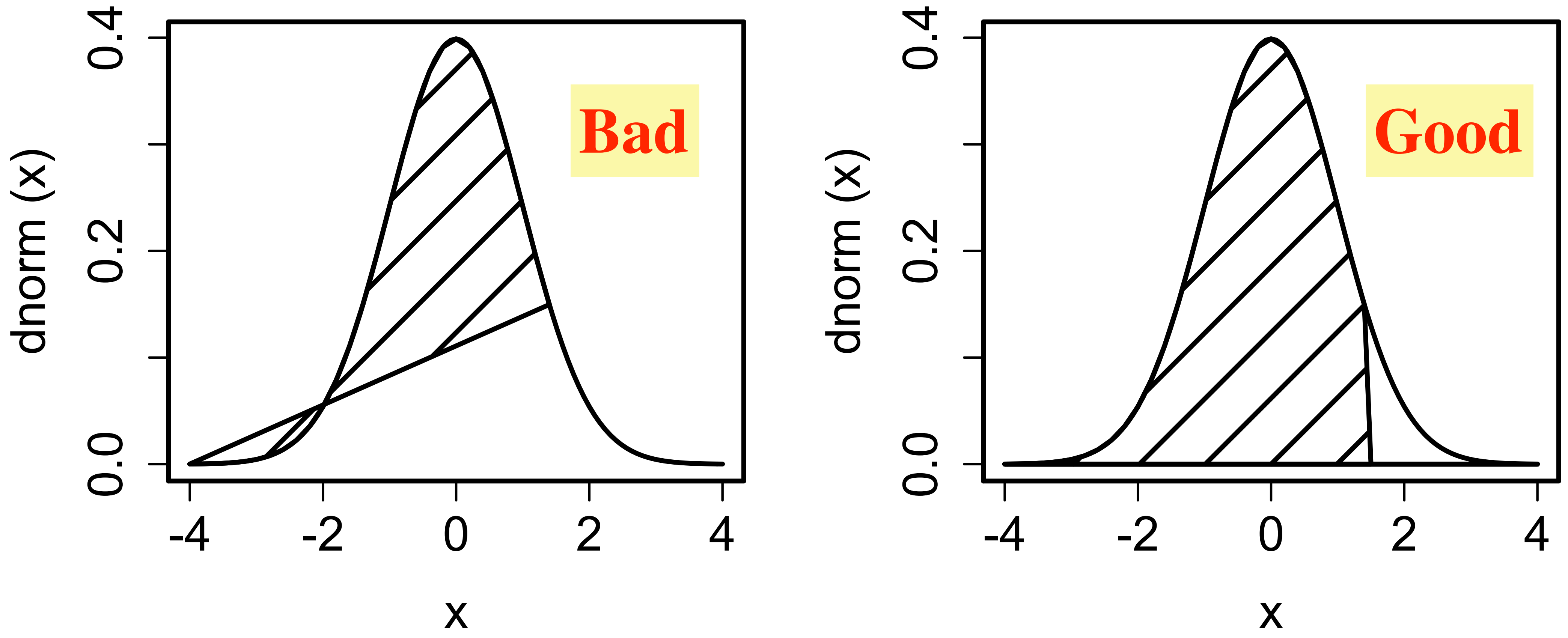


This leads to the question: Given a **single sample** of size (**n**), and its mean (**m**), how confident am I that it is a sample from a population with a specific with mean μ?

Confidence Intervals

- Given a normal distribution **N(0,1)**, and a random variable **X**∈**N(0,1)**, **what is the probability that x < 3?**
- We answer that graphically

Draw normal distribution with hatched polygon



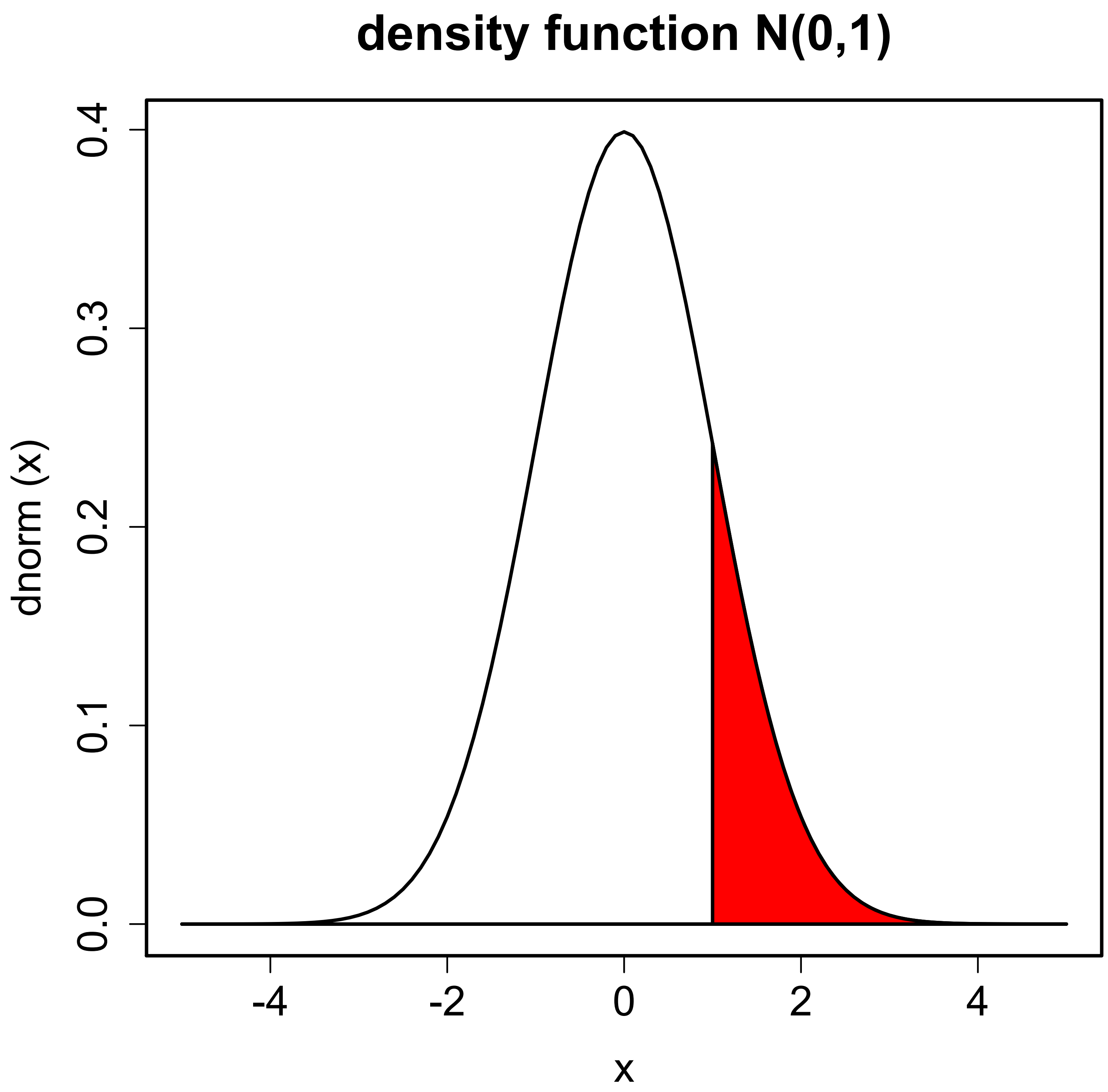
```
par(ps=18, lwd=2)
par(mfrow=c(2,2))
plot(dnorm, from=-4, to=4)
xp = seq(-4,1.5,.2)
polygon(xp,dnorm(xp), angle=45,
   density=5)
```

```
more = c(1.5,-4);
yp = append(dnorm(xp), c(0.,0.))
xp = append(xp,c(1.5,-4))
plot(dnorm, from=-4, to=4)
polygon(c(-4,4),c(0,0))
polygon(xp,yp, angle=45,
density=5,col='black')
```

Create my own function

```
# Draw normal distribution with hatched polygon

filled.normal <- function(from=-5,to=5, mean, sd,
  hatch.from=-5, hatch.to=1.5, col='red', angle=45, density=5)  {
    xp = c(seq(hatch.from,hatch.to,(hatch.to-hatch.from)/50), c(hatch.to, hatch.from))
    yp = c(dnorm(xp)[1:(length(xp)-2)], c(0., 0.))
    plot(dnorm, from=from, to=to)
    polygon(c(from,to),c(0,0))
    # without density argument, I get filled polygon
    polygon(xp,yp, col='red', border="black")
}
```

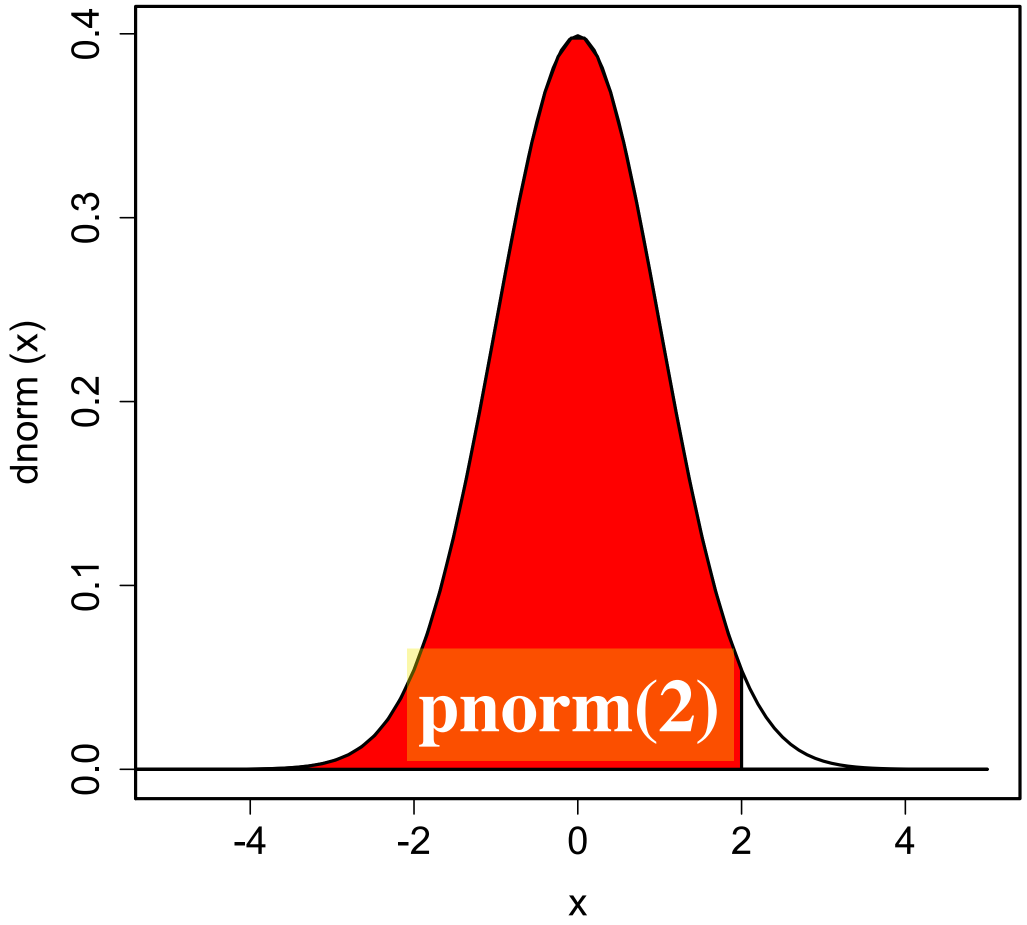
```
> source("hatched_function.r")
> fill.normal(hatch.from=1,hatch.to=5)
> title(main="density function N(0,1)")
```



The red area represents the probability that the random variable X lies between 1 and 5

The probability that X lies between -∞ and ∞ is one (certainty)

pnorm(...)



```
filled.normal(hatch.from=-10.,hatch.to=2)
```

```
> qnorm(.95)
[1] 1.644854
```

- The area under the density plot is the probability that X ≤ 2
- So let us ask another question:
  - find the value X* of X such that the probability that X ≤ X* is .95
  - Use qnorm(.95) and find X* =1.644854

qnorm()

- The probability that x < Infinity is one!!
  - qnorm(1) returns Inf

pnorm(...)

- pnorm(x) is the opposite of qnorm()
- Given x, pnorm(2.) is the probability that $X \leq 2$ if X is a random normal variable

```
> pnorm(2.)
[1] 0.9772499
> qnorm(.9772499)
[1] 2.000001
> qnorm(pnorm(2.))
[1] 2
```

```
> pnorm(qnorm(.6))
[1] 0.6
```

pnorm represents an area (between zero and one) under dnorm(x)

Given an area (between zero and one), qnorm returns a value of x

pnorm(...)

- the probability that x < Infinity is obviously one!

  pnorm(Inf) is 1

  pnorm(0) is 0.5

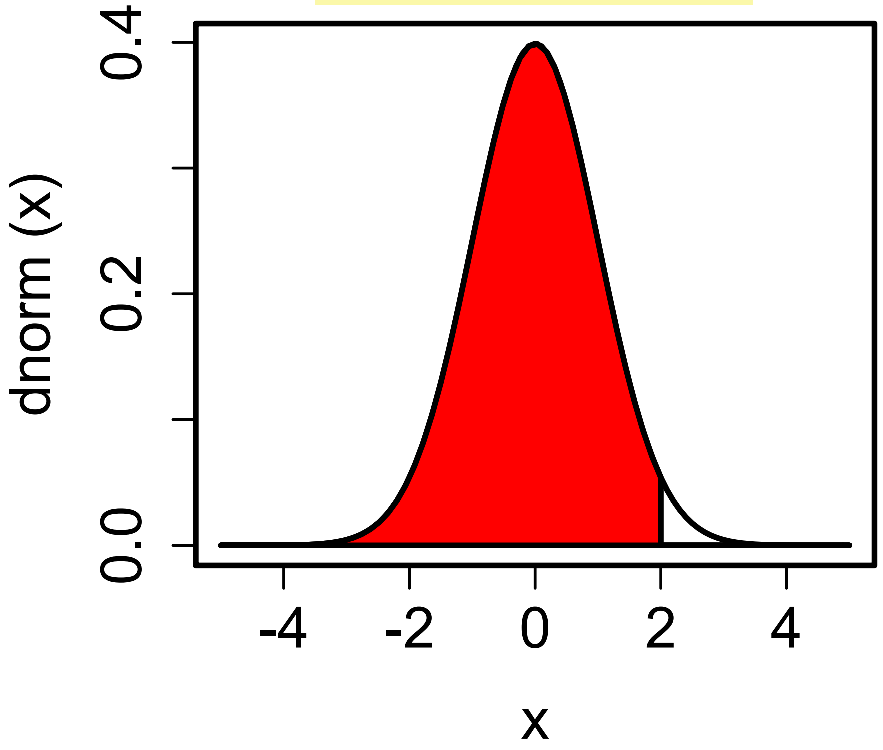
  pnorm(0,mean=1) returns 0.1586...

  pnorm(0,mean=5) returns 2.866...*10^{-7}

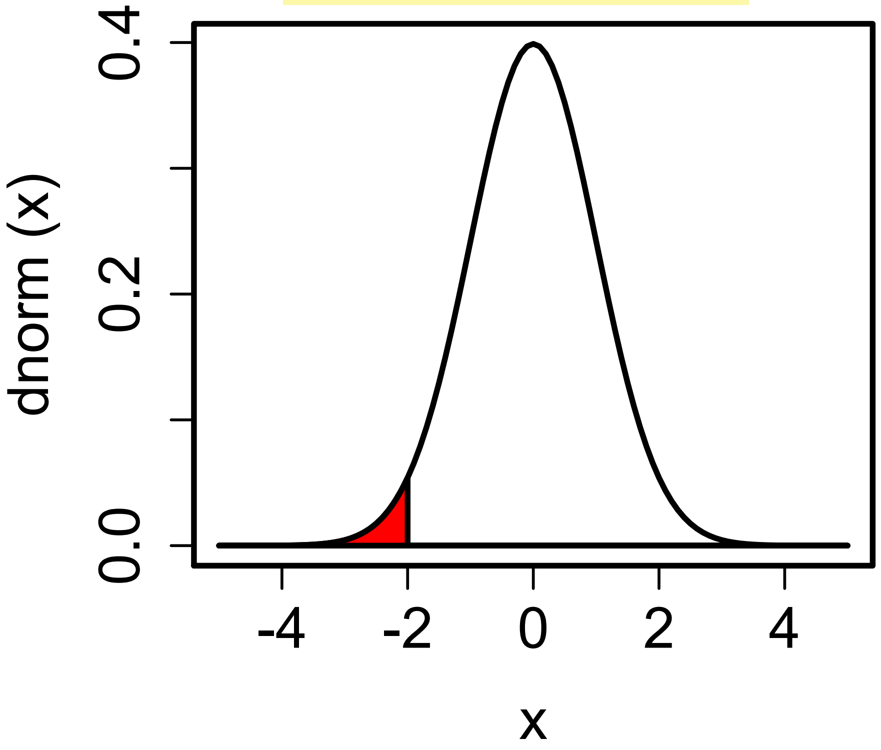
Probability interval

- Choose a random value of X using rnorm(1)
- What is the probability that X lies between -2 and 2?
- Graphical solution:
  - 1) compute the probability that X < 2
  - 2) compute the probability that X < -2
  - subtract the second from the first

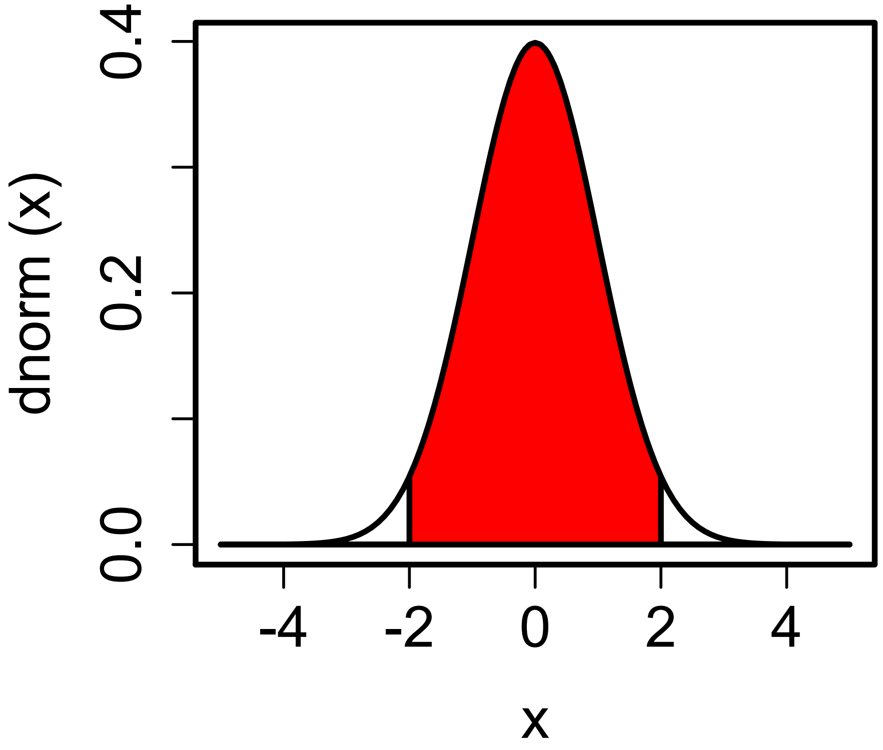
Prob(X < 2)



Prob(X < -2)



Prob(-2 < X < 2)



Prob(-2 < X < 2)

=

Prob(X < 2) - Prob(X < -2)

=

pnorm(2) - pnorm(-2)

which returns 0.954

Code for plots on previous slide

```
> source(“hatched_function.r”)
> par(mfrow=c(2,2))
> filled.normal(hatch.from=-5,hatch.to=2)
> filled.normal(hatch.from=-5,hatch.to=-2)
> filled.normal(hatch.from=-2,hatch.to=2)
```

Content of “hatched_function.r”

```
filled.normal <- function(from=-5,to=5, mean, sd,
  hatch.from=-5, hatch.to=1.5, col='red', angle=45, density=null)  {
    xp = c(seq(hatch.from,hatch.to,(hatch.to-hatch.from)/50), c(hatch.to,
hatch.from))
    yp = c(dnorm(xp)[1:(length(xp)-2)], c(0., 0.))
    plot(dnorm, from=from, to=to)
    polygon(c(from,to),c(0,0))
    # without density argument, I get filled polygon
    polygon(xp,yp, col='red', border="black",density=density)
}
```

Confidence Interval

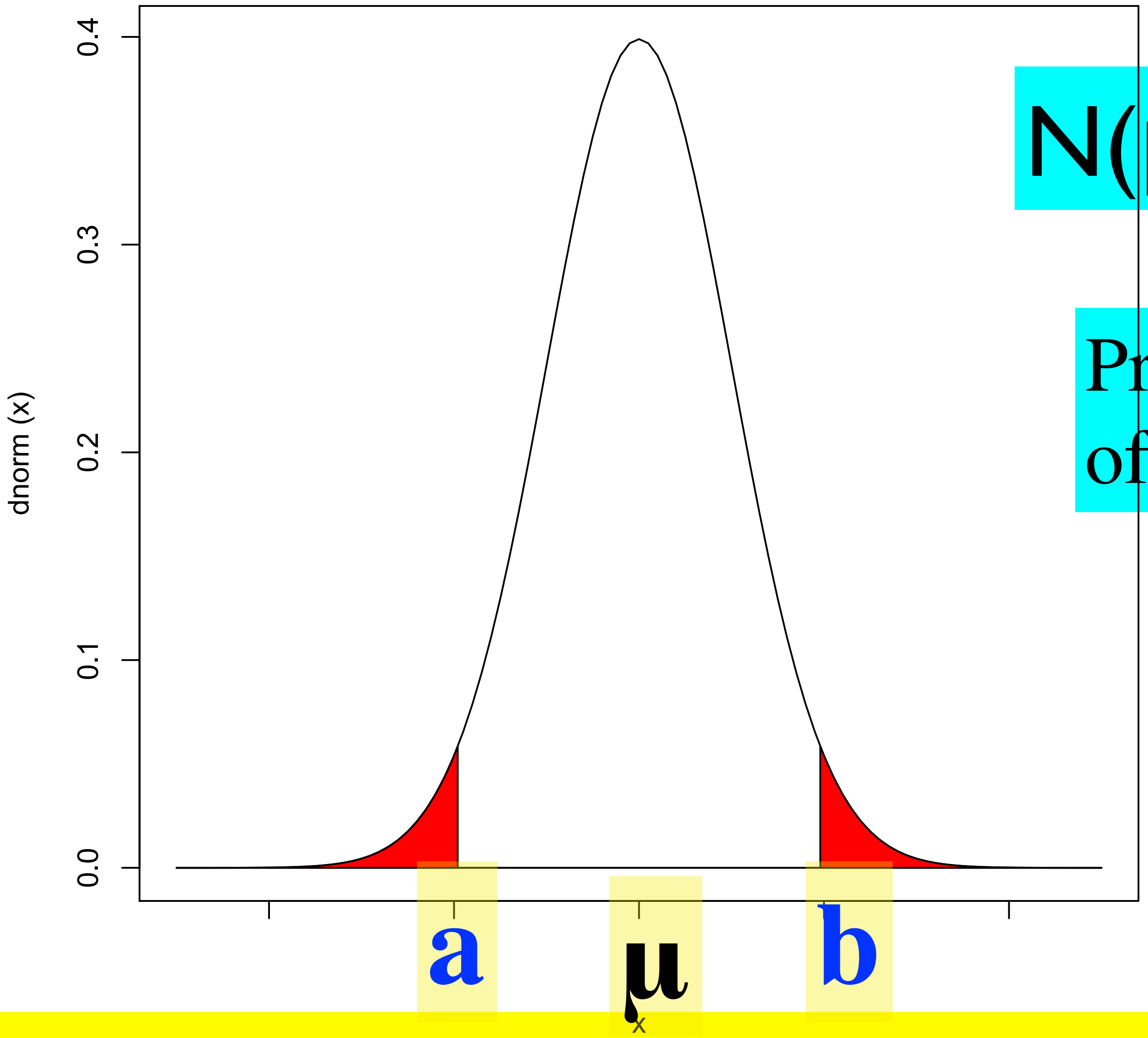
- Given a normal distribution N(0,1)
- Pick a sample with n=20 elements
  - samp = rnorm(20)
- Compute the mean of this sample
  - samp.mean = mean(samp)
- Question:
  - Given only the **sample mean s** and the sample size, what can I say about the **population mean?**

H0 and H1

- H0: the population mean is μ=s
  - s is the known sample mean
- Ha: (or H1): alternative hypothesis: the population mean μ≠s

Confidence level α

- **If H0 is true,** the sample mean equals the population mean
- What is the distribution of the sample mean m?
  - Answer: $m \in N(\mu,\sigma^2/n)$
- When is H0 true? See next slide.



$N(\mu,\sigma^2/n)$

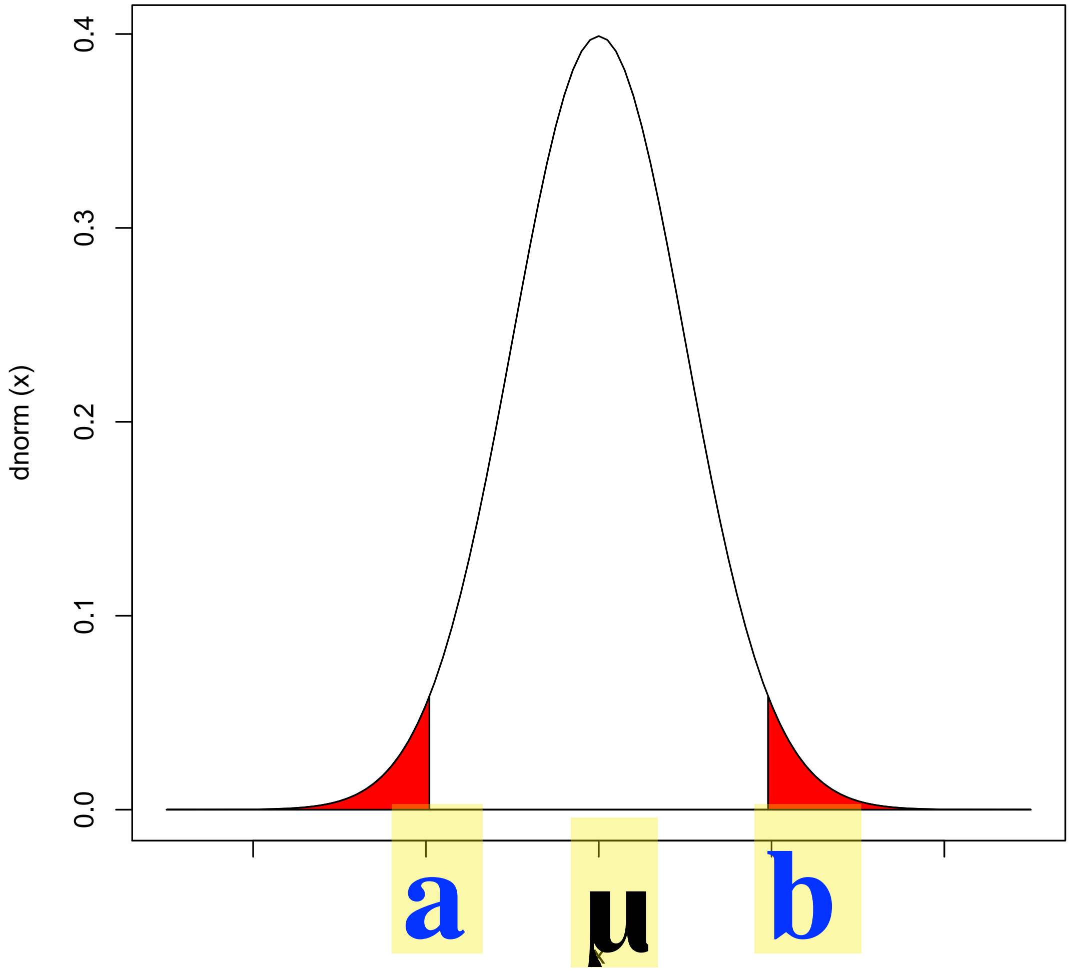
Probability distribution of sample means

As long as m falls *outside* the red region (called rejection region), H0 is considered to be true. That happens with a probability of 95% (i.e., for 95 samples out of 100 on average)

Each red region has an area of 0.025 (2.5 percent), for a total of 5 percent.

Variance of population

- Given a sample, we wish to know whether it comes from a particular population of mean μ
- We do not know the variance of this population
- The best we can do is estimate it.
  - we base the estimate on the sample data
  - we use an unbiased estimate



N(μ,σ^2/n)

X is a random variable

In this case, **X** is the sample mean which follows N(μ,σ^2/n)

$$\text{Prob}(a \leq X \leq b) \quad = \quad 1 - \alpha$$

z-normal statistic

$$z \quad = \quad \frac{X - \mu}{\sigma/\sqrt{n}}$$

$$\text{Prob}(a \leq \frac{z\sigma}{\sqrt{n}} + \mu \leq b) \quad = \quad 1 - \alpha$$

C(...): Confidence interval

$$C(a - \frac{z\sigma}{\sqrt{n}} \leq \mu \leq b - \frac{z\sigma}{\sqrt{n}}) \quad = \quad 1 - \alpha$$

$$C(a - \frac{z\sigma}{\sqrt{n}} \leq \mu \leq b - \frac{z\sigma}{\sqrt{n}}) \quad = \quad 1 - \alpha$$

population mean μ and
population standard deviation σ
are constant
a and b are also constant and a function of the
confidence level 1-α

Each sample generates a new *z* in *N(0,1)*
The confidence level C(...) is a function of the sample.

For a large number of samples, the population mean is within this confidence interval (1- α) percent of the time.

Usually, α=0.05, so the population mean is within the confidence interval 95% of the time.

Experiment in R

- We will consider a normal population of mean 5 and standard deviation 2
  - (sample.size,mean=5,sd=2)
- We will consider a single sample of size 30:

  ```
  sample.size = 30
  sampl = rnorm(30)
  ```

t.test

Performs one and two sample t-tests on vectors of data.

```
## Default S3 method:
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)
```

x is a vector

```
sample.size = 30
mean = 5
sd = 2
sampl = rnorm(sample.size, mean=mean, sd=sd)
test1 = t.test(sampl, mu=mean)
test2 = t.test(sampl, mu=0)
print(test1)
print(test2)
names(test1)
```

Let us look at the output to test1

```
sample.size = 30
mean = 5
sd = 2
sampl = rnorm(sample.size, mean=mean, sd=sd)
test1 = t.test(sampl, mu=mean)
print(test1)
print(names(test1))
```

R code in single_test.r

Output from code

```
        One Sample t-test

data:  sampl
t = -0.2611, df = 29, p-value = 0.7959
alternative hypothesis: true mean is not equal to 5
95 percent confidence interval:
 4.147606 5.659397
sample estimates:
mean of x
 4.903501

[1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
[6] "null.value"  "alternative" "method"      "data.name"
```

```
data:  sampl
t = -0.2611, df = 29, p-value = 0.7959
alternative hypothesis: true mean is not equal to 5
95 percent confidence interval:
 4.147606 5.659397
sample estimates:
mean of x
 4.903501
```

As long as the p-value is greater than 0.05, the H0 hypothesis is assumed to be true.

In this example, therefore, the true mean could be equal to 5. The 95% confidence interval includes 5.

Example Problem

Criminals v. Cambridge Men

criminal_cambridge.RData

Do criminals and Cambridge men differ in height?

Really asking is are the means of the two groups the same (assuming the variance is, too).

H0: the mean of the two groups is equal

H1: the means are not equal, so must differ

Use t.test(...) to compare the mean of two different samples (which can have different sizes)

Criminals v. Cambridge Men

```
> X = read.table("criminal_cambridge.RData")
> criminals = subset(X,source=="criminal") #or X[X$source==”criminal,]
> cambridge = subset(X,source=="cambridge")
> t.test(criminals$height.cm, cambridge$height.cm)
```

```
        Welch Two Sample t-test


data:  criminals$height.cm and cambridge$height.cm
t = -36.1876, df = 1705.635, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -9.051622 -8.120879
sample estimates:
mean of x mean of y
 166.3014  174.8877
```

So,...

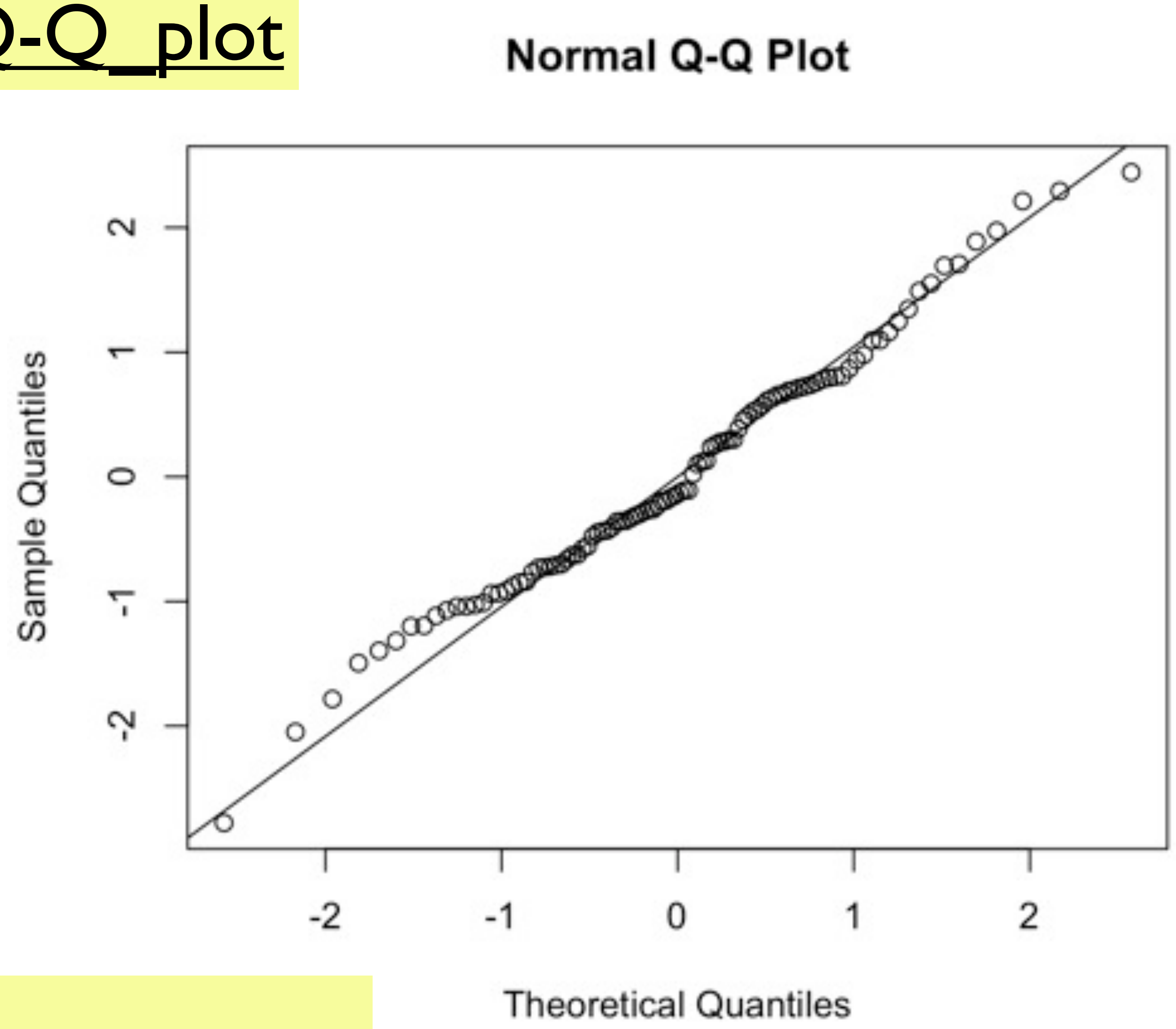
The average criminal is significantly shorter than the average Cambridge man!

Better keep an eye on those short people.

Is this normal?

http://en.wikipedia.org/wiki/Q-Q_plot

```
> y = rnorm(100)
> qqnorm(y)
> qqline(y)
```



?qqline

line through 1st and 3rd quantiles
of normal distribution and of data

qnorm(.25) and qnorm(.75) # -.666 and .666

shapiro.test()

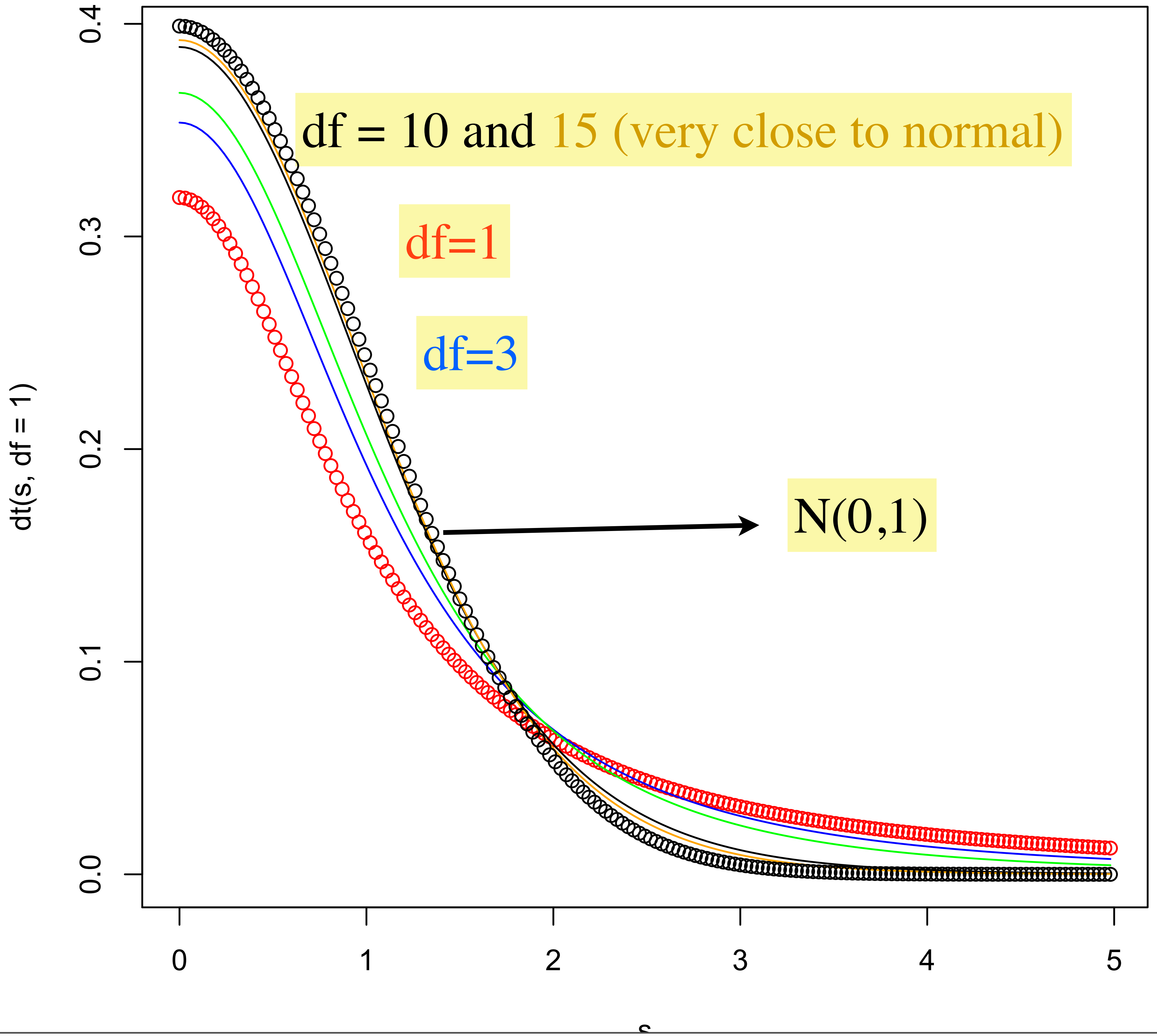
```
> shapiro.test(y)

    Shapiro-Wilk normality test

data:  y

W = 0.9871, p-value = 0.4431
```

if $p < 0.05$, the proposed sample is not considered to be normal.

The y vector is considered normal

Student Distributions

dt(df=...)



df = 10 and 15 (very close to normal)

df=1

df=3

N(0,1)

Shapiro test with student distribution

```
shapiro.test(rt(100,1))

shapiro.test(rt(100,3))

shapiro.test(rt(100,5))

shapiro.test(rt(100,8))
```

p-value in results increase
beyond 0.05 when degrees
of freedom is slightly beyond 5

Distribution of sample mean, revisited

- If the population is $N(\mu,\sigma^2)$, a sample of size *n* is composed of *n* random variables, which change value for each sample
- The sum of independent normal random variables is a normal random variable. Therefore, the sample mean is a random variable with mean μ (also called expected value: **E(sample mean) = μ**)
- Each of these random variables has s.d. σ
- The sample mean follows **$N(\mu, \sigma^2/n)$**
- **HOWEVER:** we do not know μ or σ

Unbiased Variance of Sample

- Given a sample (stored as a vector of numbers), for example:
  - sampl = sample(1:1000,size=100)
    stdev = sd(sample); mean = mean(sample)
  - sd(sample) is identical to
    ss = sum(sampl^2-mean(sampl)^2)
    s = stdev = sqrt(ss/99)
  - So we work with
    $N(m, s^2/n)$ instead of **$N(\mu, \sigma^2/n)$**

N(m, s²)

$$
\begin{aligned}
m &= (X_1 + X_2 + \cdots + X_n)/n \\
s^2 &= [(X_1 - m)^2 + (X_2 - m)^2 + \cdots + (X_n - m)^2]/(n-1)
\end{aligned}
$$

n degrees of freedom

n-1 degrees of freedom

All the X’s are random variables taken from the population. Thus, the confidence interval is not calculated based on

$$z = \frac{m - \mu}{\sigma/\sqrt{n}}$$

but

$$z = \frac{m - \mu}{s/\sqrt{n}}$$

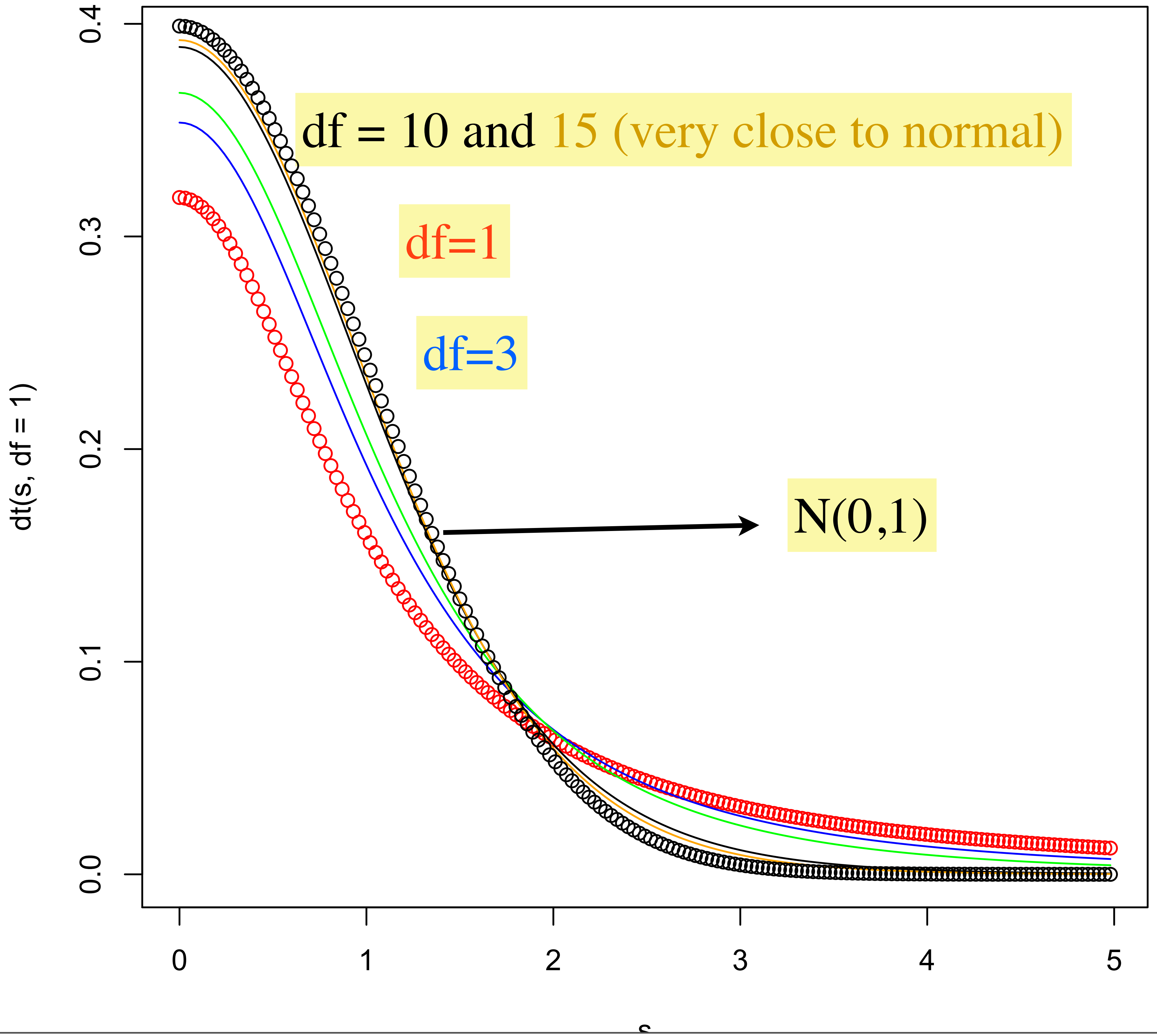
which follows a Student distribution t(n-1) with n-1 degrees of freedom

R commands for Student Distribution

- Normal Distribution
    - **rnorm, pnorm, qnorm, rnorm**
- Student Distribution
    - **dt, pt, qt, rt**
- F Distribution
    - **df, pf, qf, tf**

Student Distributions

dt(df=...)



df = 10 and 15 (very close to normal)

df=1

df=3

N(0,1)

Large sample sizes

- As the sample size becomes larger than 10, one can safely replace the Student distribution of sample means by a normal distribution
- Use t.test(...) for hypothesis testing.

Experiment in R

- We will consider a normal population of mean 5 and standard deviation 2
  - (sample.size,mean=5,sd=2)
- Run 1000 samples and compute confidence intervals for each, with 95% confidence interval ($\alpha = 0.05$)
- Measure (with R) the number of intervals that do not contain the mean $\mu = 5$

R code: monitor_confidence_intervals.r

```
sample.size = 30
mean = 5
sd = 2

low.count = 0
high.count = 0
nb.samples = 1000

# Consider 1000 samples from N(5,4)
# In how many cases does the
# confidence interval
# not contain the mean?
```

```
for (i in 1:nb.samples) {
    sampl = rnorm(sample.size,
                  mean=mean, sd=sd)
    test1 = t.test(sampl, mu=mean)
    low = test1$conf.int[1]
    high = test1$conf.int[2]
    cat(low, high, "\n")

    if (low < 5 && high < 5) {
        low.count = low.count + 1
    }
    if (low > 5 && high > 5) {
        high.count = high.count + 1
    }
}

cat("low.count= ", low.count, "\n")
cat("high.count= ", high.count, '\n')
```