Statistics with R

Bernard Billoud
Morphogenesis of Macro-Algae - UMR8227
Bernard.Billoud@sb-roscoff.fr

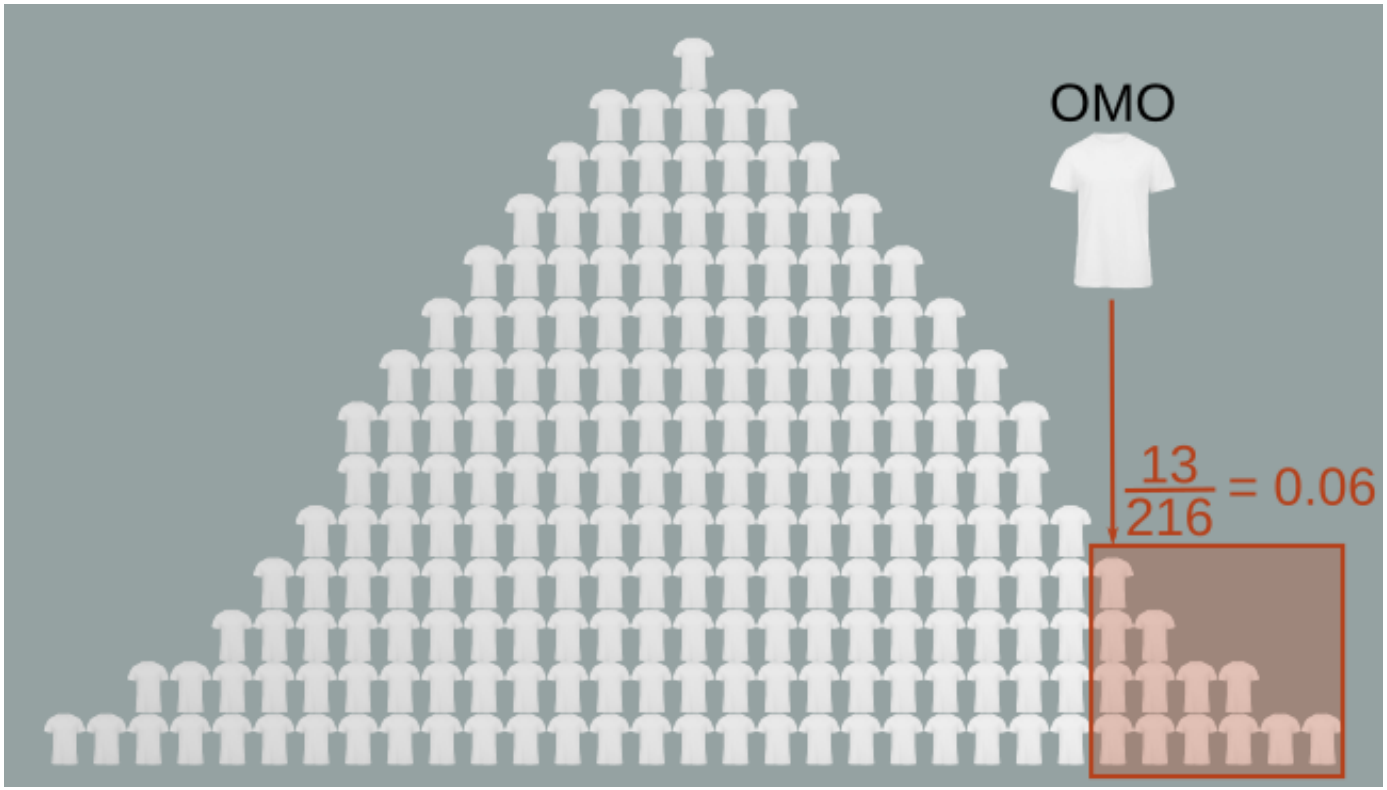
June 1, 2021

Long is the day

The father of all tests: the Gauss test

What proportion of all "usually-washed" T-shirts are whiter than the one washed with the new OMO?

T-Shirt whiteness distribution:
31 shades of white



The T-Shirt test (enhanced version)

Threshold Choose a confidence level, *e.g.* $\alpha = 0.05$;

Null hypothesis H_0 Let us suppose that the new OMO is *not* better than my usual powder;

Variable distribution Knowing the distribution of W, the whiteness of my $N_{tot} = 216$ regular T-Shirts;

Test value Knowing W_O the whiteness of the T-Shirt I washed using the new OMO;

p-value The probability to find a regular T-Shirt as white as (or whiter than) the OMO T-Shirt is:

$$P(W \geq W_O) = \frac{N(W \geq W_O)}{N_{tot}}$$

As $P = 0.06 \geq \alpha$, it can be expected that by chance, a T-Shirt picked at random among the regular ones would be at least as white as the one I washed with the new OMO.

So:

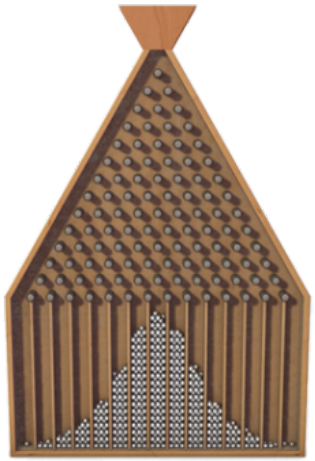
I cannot assert that the new OMO washes whiter than my usual powder.

Code for the T-Shirt test

```
# T-Shirt whiteness distribution -- version 1
tsw <- read.table("RegularWashing.csv",header=T)
B = (min(tsw$whiteness)-0.5):(max(tsw$whiteness)+0.5)
h <- hist(tsw$whiteness,breaks=B)
W = h$mids
N = h$counts

# T-Shirt whiteness distribution -- version 2
N = c(1,1,2,2,3,4,5,7,8,9,10,11,12,13,13,14,13,13,12,11,10,9,8,7,5,4,3,2,2,1,1)
W = seq(0,length(N)-1)

# Whiteness of the OMO-washed T-Shirt and number of whiter T-Shirts
W0 = 25
listOK = N[W>=W0]
nOK = sum(listOK)

# Test
Ntot = sum(N)
cat("\n",Ntot," T-Shirts, in which ",nOK," have whiteness >= ",W0,": ",sep="")
print(listOK)
pv = nOK / Ntot
cat("p-value =",pv,"\n")

# Graphic representation
barplot(N,main="T-Shirt whiteness distribution",density=30-W,
        xlab="whiteness",ylab="# of T-Shirts",names.arg=W)
addOK = c(rep(0,W0),listOK)
barplot(addOK,col="pink",add=TRUE)
```

Counting and computing

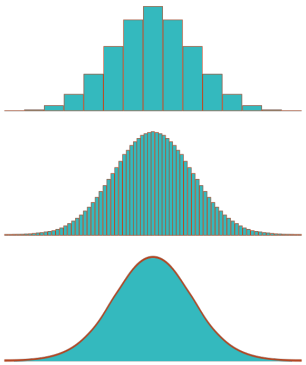
A simple model of allocation in classes: binomial (discrete)



- The value X (position of one bead) depends on multiple (n rows) "choices" (L/R) with probability p ($= 1/2$)
- Distribution among $n+1$ values (bucket $0 \leq k \leq n$):

$$P(X = k) = B_{n,p}(k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$

When n increases:



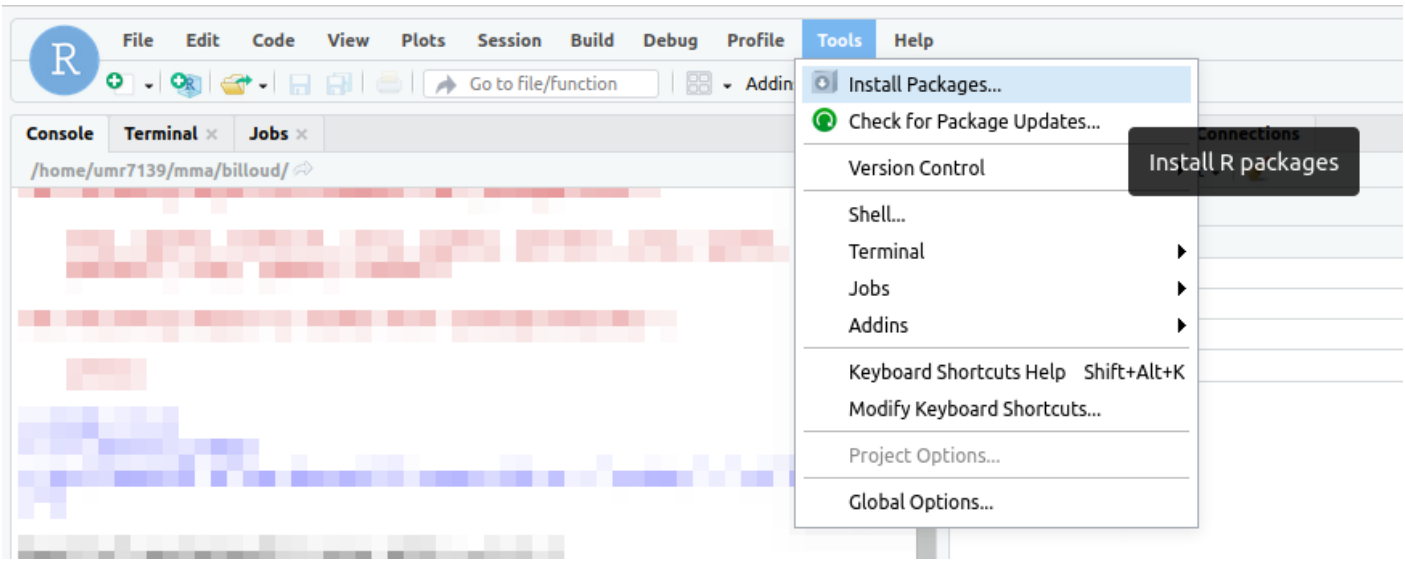
- value range extends: $X \to$ continuous
- $B_{n,p} \to \mathcal{N}_{np,np(1-p)}$
- Density of probability:

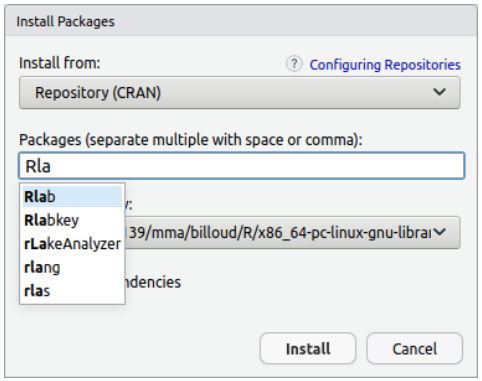
$$P(x) = N_{\mu,\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Interlude: installing a package

RStudio allows any user to – locally – install a package

```
install.packages("Rlab")
```





Code for the Galton board

```
library(Rlab) # Rlab is used for Bernouilli assay (rbern)

# Draw 0 or 1 with proba 0.5 (= 1 pod of the Galton board)
proba = 0.5
d = rbern(1,proba)
cat("\nResult of one Bernouilli assay with p = ",proba,": ",d,"\n",sep="")

# Assuming 0 = L ; 1 =  R, draw 14 times 0 or 1 with proba 0.5 -> bucket number for 1 bead
nlin = 14
x = sum(rbern(nlin,proba))
cat("\nSum of n = ",nlin," Bernouilli assays with p = ",proba,": ",x,"\n",sep="")

# Looping
nb = 10000    # Launch 10000 beads
X <- c()
for (i in 1:nb) {
  X <- c(X,sum(rbern(nlin,proba)))
}

m = mean(X) # expected: 14 × 0.5 = 7
s = sd(X)   # expected: √(14×0.5×0.5) = 1.87
cat(nb," sums of ",nlin," assays: mean = ",m,"; s.d. = ",s,"\n",sep="")
cat("Expected: mean = ",nlin*proba,"; s.d. = ",sqrt(nlin*proba*(1-proba)),"\n",sep="")

# Show
hist(X,breaks=-0.5:(nlin+.5),xlim=c(0,nlin),col="orange",
   main=paste("Binomial law: n = ",nlin,"; p = ",proba," (pop = ",nb,")",sep=""))
```

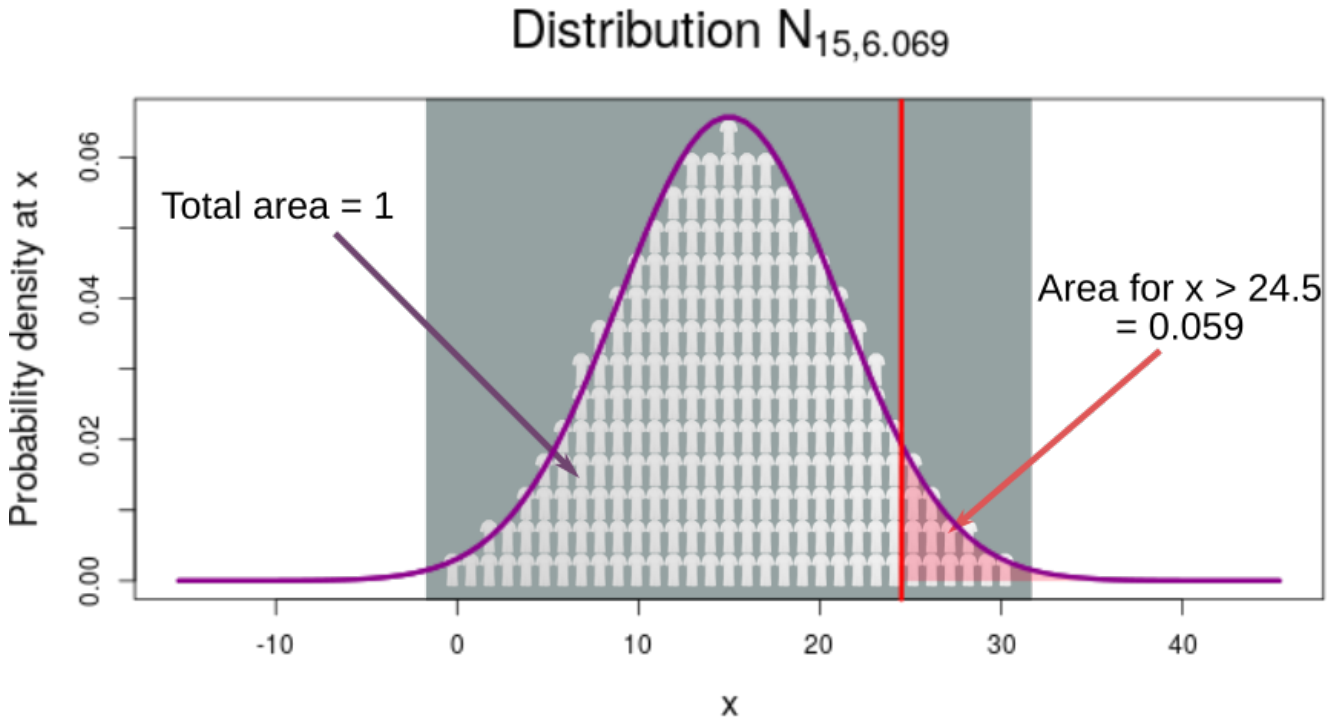
Improve efficiency and converge to the Normal law

```
cat("\nUsing rbinom:\n")

proba = 0.5 ; nlin = 14 ; nb = 10000
# Use the appropriate R function
X = rbinom(nb, nlin, proba)

m = mean(X) # expected: 14 × 0.5 = 7
s = sd(X)   # expected: √(14×0.5×0.5) = 1.87
cat(nb," sums of ",nlin," assays: mean = ",m,"; s.d. = ",s,"\n",sep="")
cat("Expected: mean = ",nlin*proba,"; s.d. = ",sqrt(nlin*proba*(1-proba)),"\n",sep="")

cat("\nIncreasing n:\n")
# Increase nlin (try also with nb = 100000)
nlin = 436
X = rbinom(nb, nlin, proba)
m = mean(X) # expected: 436 × 0.5 = 218
s = sd(X)   # expected: √(436×0.5×0.5) = 10.44
hist(X,breaks=-0.5:(nlin+.5),col="yellow",freq=F,
   xlim=c(nlin*proba-5*sqrt(nlin*proba*(1-proba)),nlin*proba+5*sqrt(nlin*proba*(1-proba))),
   main=paste("Binomial law: n = ",nlin,"; p = ",proba," (pop = ",nb,")",sep=""))

# Compare with Normal
valist = 0:nlin
normd = dnorm(valist,m,s)
points(valist,normd,type="l",lwd=3,col="blue")
cat(nb," sums of ",nlin," assays: mean = ",m,"; s.d. = ",s,"\n",sep="")
cat("Expected: mean = ",nlin*proba,"; s.d. = ",sqrt(nlin*proba*(1-proba)),"\n",sep="")
```

The *real* Gauss test = Z-test

What proportion of all "usually-washed" T-shirts are whiter than the one washed with the new OMO?

T-Shirt whiteness distribution - continuous version

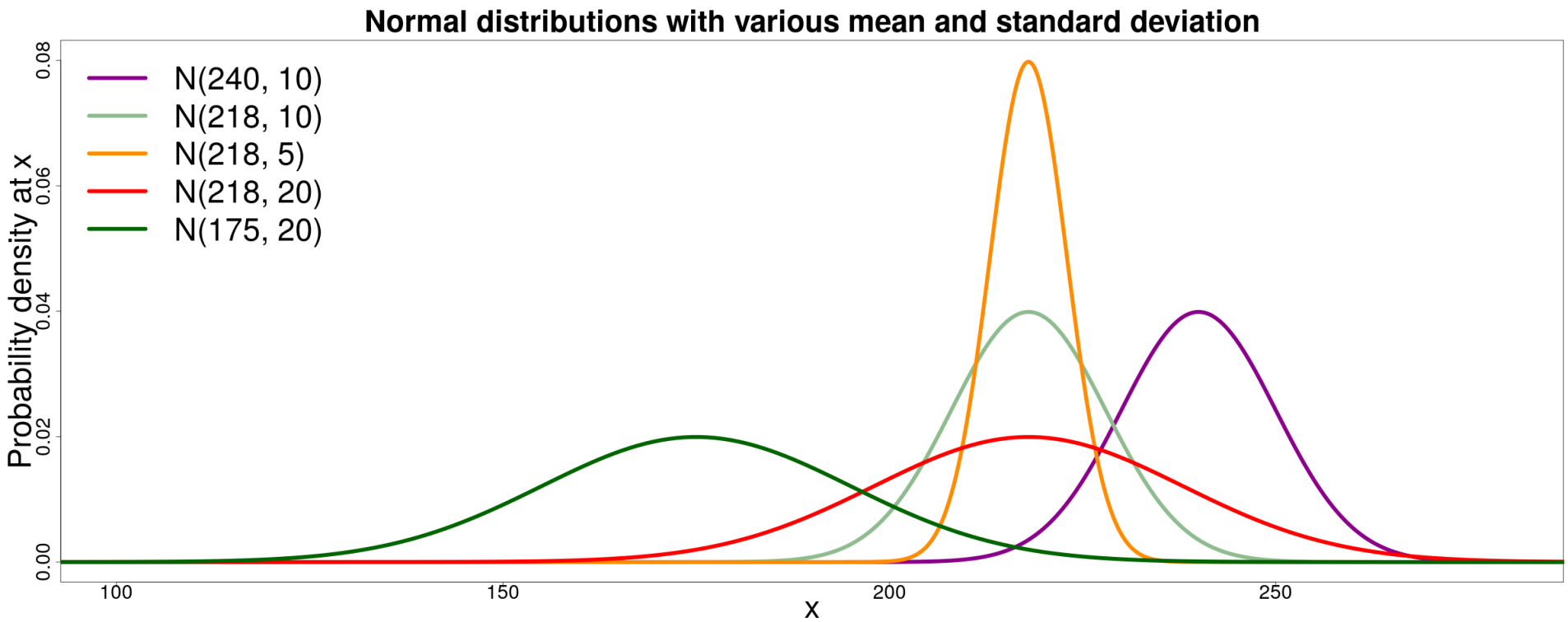


Code for the Z-test

```
# Load the library for Gauss.test
library(compositions)

# T-Shirt whiteness distribution
W = 0:30
N = c(1,1,2,2,3,4,5,7,8,9,10,11,12,13,13,14,13,13,12,11,10,9,8,7,5,4,3,2,2,1,1)
Ntot = sum(N)

# Parameters for the norml law
meanW = sum(W*N) / Ntot
sdW = sqrt(sum(W*W*N) / Ntot - meanW**2)
cat("\nWhiteness: mean =",meanW,"s.d. =",sdW,"\n")

# Discrete: W0 = 25, test for W >= W0; Continuous: test for W > W0, so:
W0 = 24.5

# Test
zt = Gauss.test(W0,mean=meanW,sd=sdW,alternative="greater")
# print(zt)
pv = zt$p.value
cat("p-value for the z-test (W>",W0,"): ",pv,"\n",sep="")

# Graphic representation
valist <- seq(meanW-5*sdW,meanW+5*sdW,length.out=100)
dens <- dnorm(valist,m=meanW,s=sdW)
plot(x=valist,y=dens,type="l",col="darkmagenta",lwd=3,xlab="x",ylab="Probability density at x",
     main=paste("Distribution N(",meanW,",",round(sdW,3),")",sep=""))
abline(v=W0,lwd=3,col="red")
```

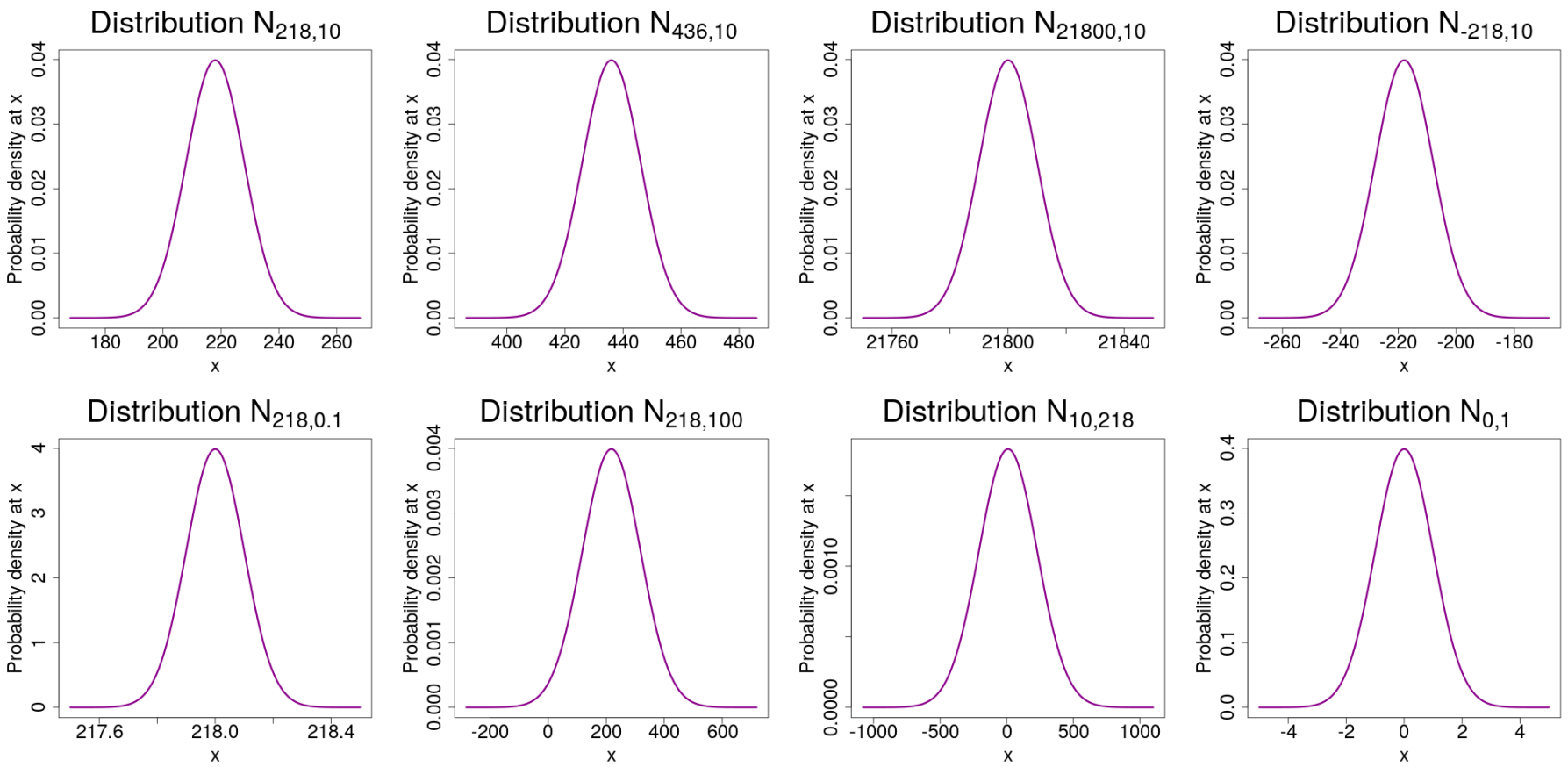
Representative curve

With different values for μ and σ



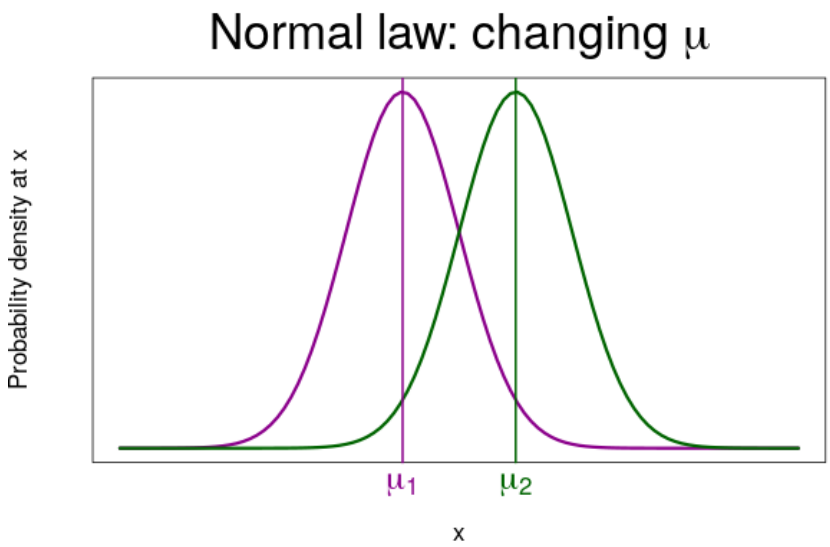
Representative curve

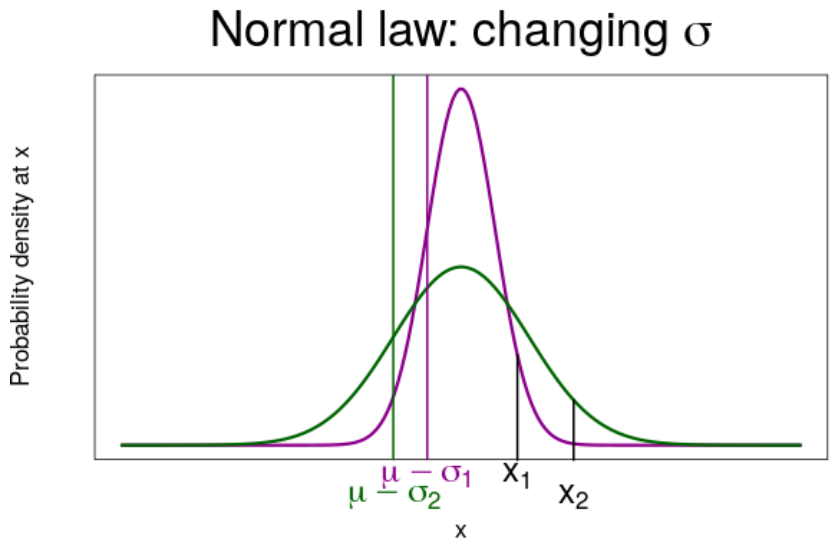
With different values for μ and σ



All curves have the same shape

Parameters μ and σ





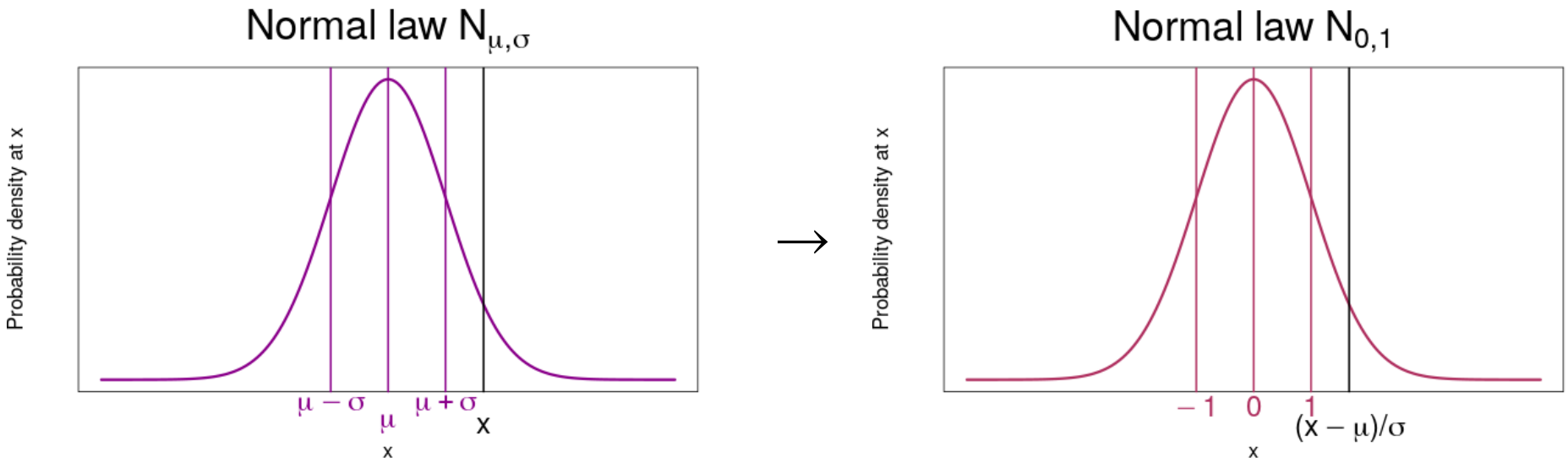
A somewhat useful formula:

$$P_2(x_2) = P_1(x_1) = P_1(x_2 - (\mu_2 - \mu_1))$$

A completely useless formula:

$$P_2(x_2) = \frac{\sigma_1}{\sigma_2} P_1(x_1) = \frac{\sigma_1}{\sigma_2} P_1\left(\frac{\sigma_1}{\sigma_2} x_2 + \mu\left(1 - \frac{\sigma_1}{\sigma_2}\right)\right)$$

Mapping to the standard normal distribution



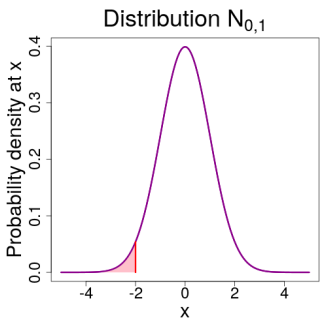
A really useful formula: $P_{\mu,\sigma}(x) = \frac{1}{\sigma} P_{0,1}\left(\frac{x-\mu}{\sigma}\right)$

Or: $X \mapsto N_{\mu,\sigma} \Leftrightarrow X \mapsto \mu + \sigma \times N_{0,1}$

Density of probability and area

Actually, we are interested by the *area* under the curve

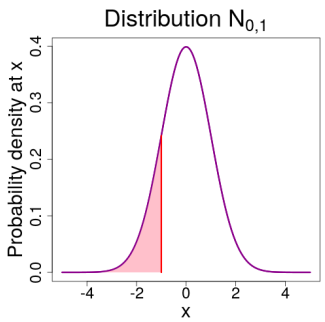
$u = -2$



$P(u) \approx 5.40\ 10^{-2}$

$A(\rightarrow u) \approx 2.275\ 10^{-2}$

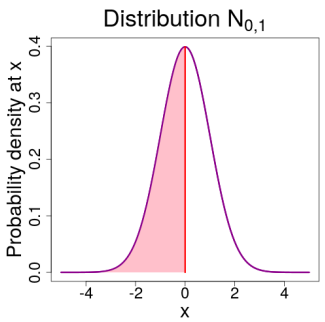
$u = -1$



$P(u) \approx 0.242$

$A(\rightarrow u) \approx 0.159$

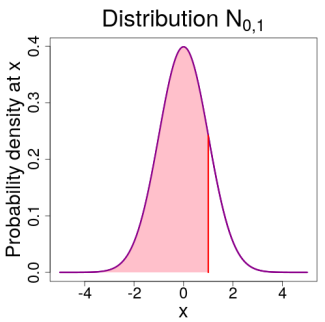
$u = 0$



$P(u) \approx 0.399$

$A(\rightarrow u) = 0.50$

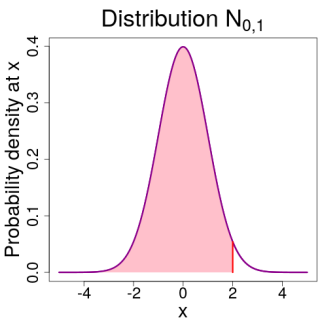
$u = 1$



$P(u) \approx 0.242$

$A(\rightarrow u) \approx 0.841$

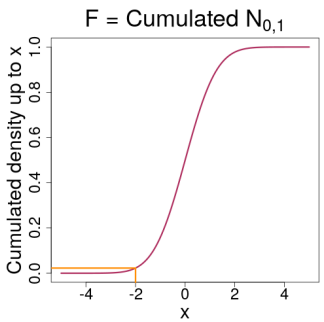
$u = 2$



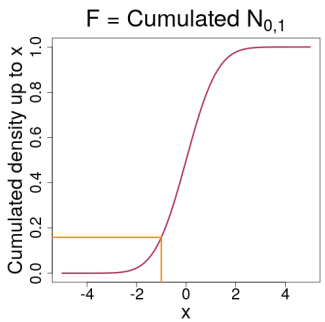
$P(u) \approx 5.40\ 10^{-2}$

$A(x) \approx 0.97725$

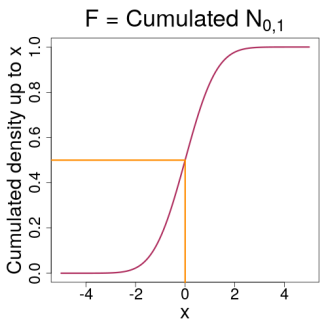
Primitive:



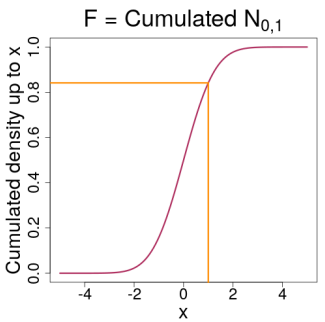
$P(x < u) \approx 2.275\ 10^{-2}$



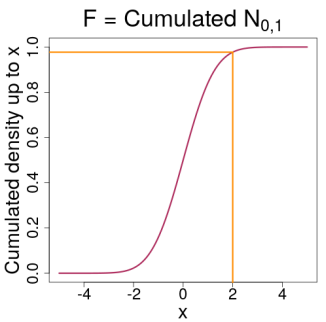
$P(x < u) \approx 0.159$



$P(x < u) = 0.50$



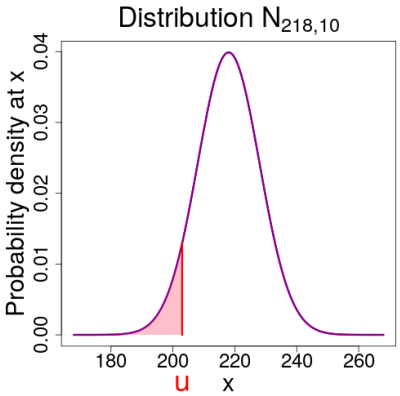
$P(x < u) \approx 0.841$



$P(x < u) \approx 0.97725$

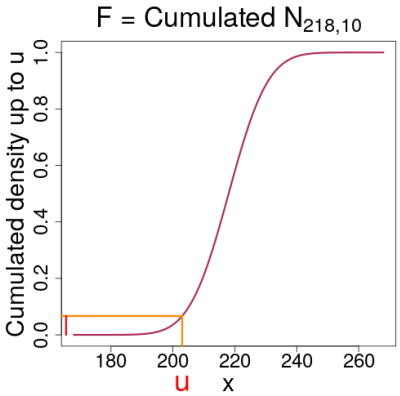
F is versatile

Left



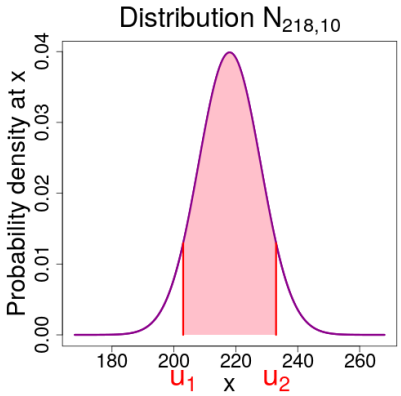
$P(x < u)$

=



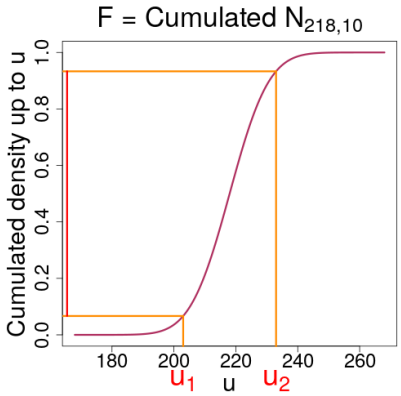
$F(u)$

Inside



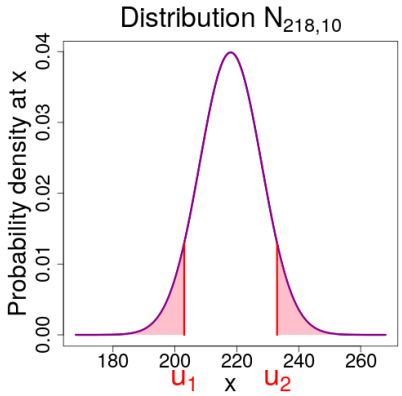
$P(u_1 < x < u_2)$

=



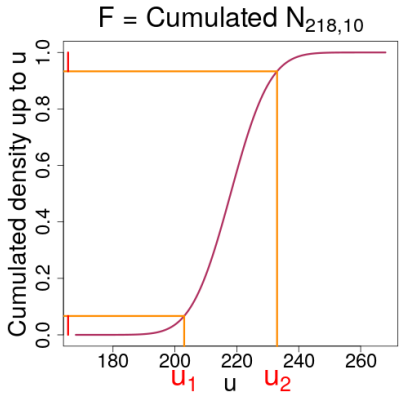
$F(u_2) - F(u_1)$

Outside



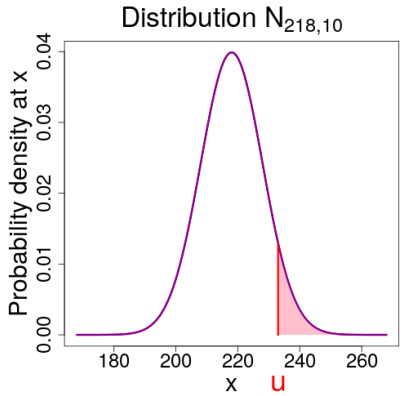
$P(x < u_1 \text{ or } u_2 < x)$

=



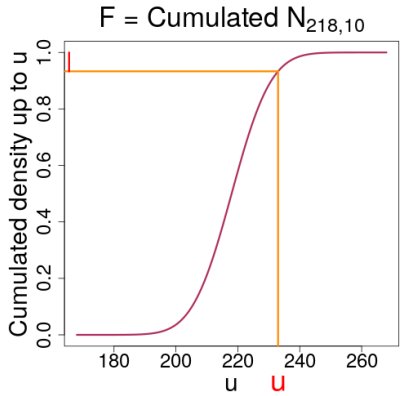
$F(u_1) + 1 - F(u_2)$

Right



$P(u < x)$

=

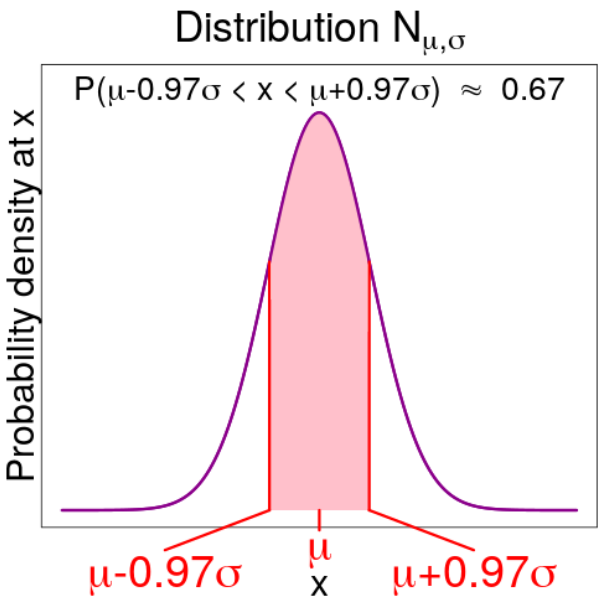


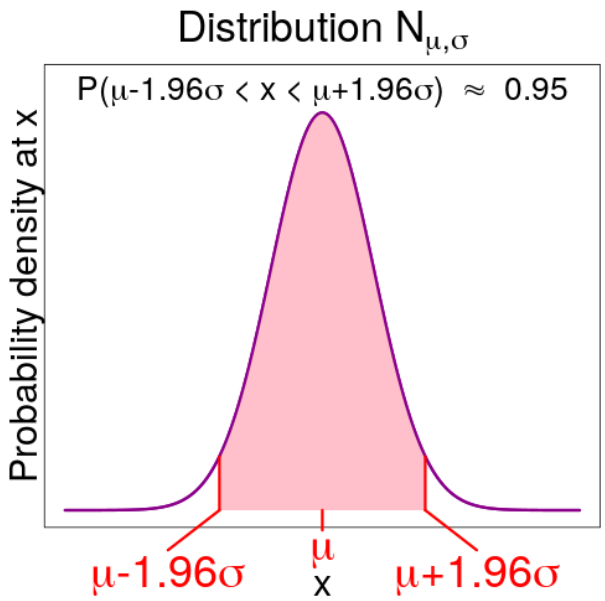
$1 - F(u)$

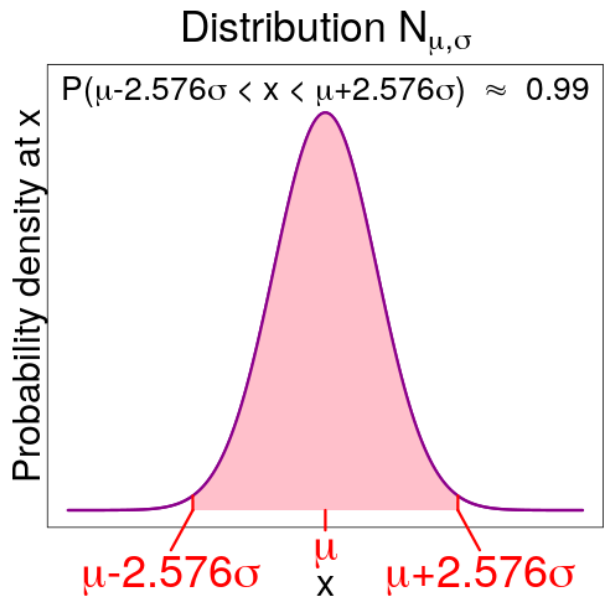
Code for various cases

```
# Suppose we have a population P, for which we know μ and σ
mu = 218
sigma = 10
cat("H0: x drawn from population: N(μ,σ) with μ = ",mu," and σ = ",sigma,"\n",sep="")

valist = c(198, 203, 208, 218, 228, 233, 238)
# pnorm directly gives the probability that x < u
cat("\nLeft test:\n")
for ( u in valist ) {
  cat("Under H0, P(x<",u,") = F(",u,") = ",pnorm(u,m=mu,s=sigma),"\n",sep="")
}

# pnorm indirectly gives the probability that u < x
cat("\nRight test:\n")
for ( u in valist ) {
  cat("Under H0, P(",u,"<x) = 1 - F(",u,") = ",1-pnorm(u,m=mu,s=sigma),"\n",sep="")
}

# pnorm indirectly gives the probability that u1 < x < u2
cat("\nInside interval:\n")
for ( i in 1:(length(valist)-1) ) {
  u1 = valist[i]
  for ( j in (i+1):length(valist) ) {
    u2 = valist[j]
    cat("Under H0, P(",u1,"<x<",u2,") = F(",u2,") - F(",u1,") = ",
                        pnorm(u2,m=mu,s=sigma)-pnorm(u1,m=mu,s=sigma),"\n",sep="")
  }
}
```

Some rules of thumb



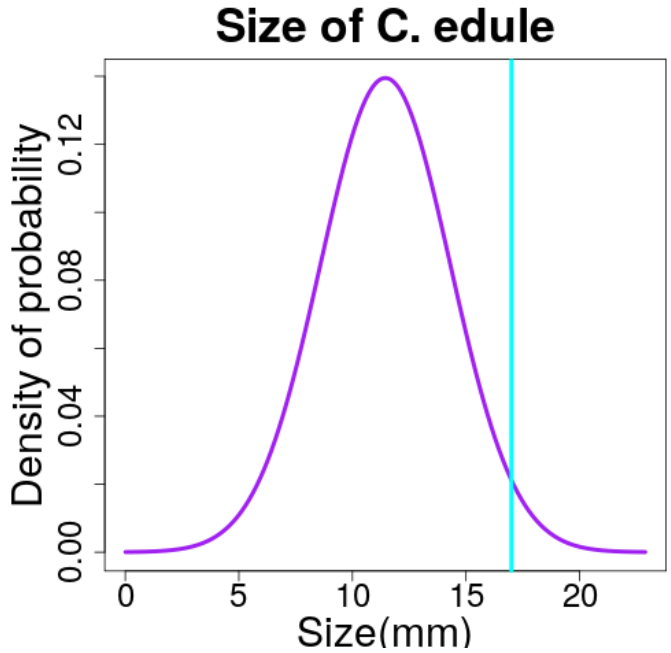




Example of application

The size of one year old *Cerastoderma edule* follows a normal law with $\mu = 11.45\text{mm}$; $\sigma = 2.86\text{mm}$[1].
We find a shell which seems to be a young *C. edule*, and we measure its size: $S = 17\text{mm}$.
Is our specimen too large to be a young *C. edule*?

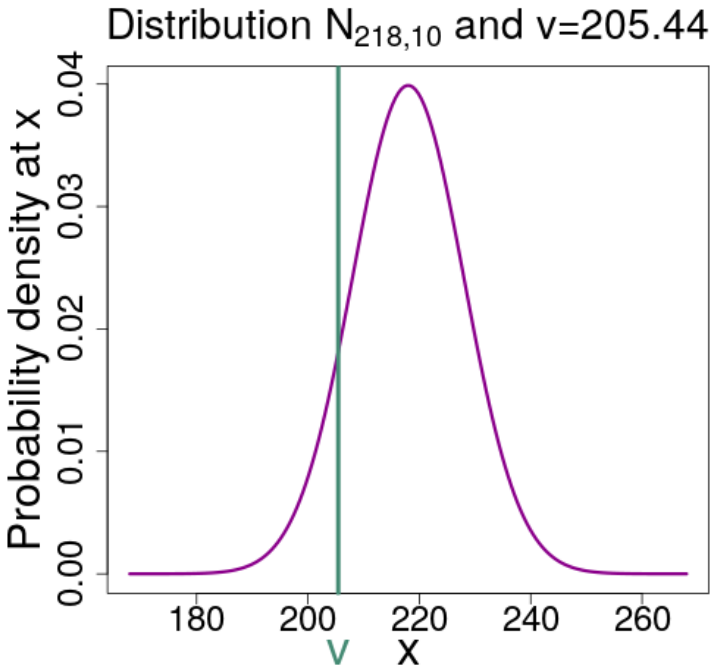

```
muA = 11.45 ; sigmaA = 2.86
siz = 17

cat("H0 : the measured size is not too big for a young C. edule.\n")
pv = 1 - pnorm(siz, m = muA, s = sigmaA)

cat("p-value =",pv,"; ")

alpha = 0.05
if(pv < alpha) {
  cat("H0 can be rejected")
} else {
  cat("One cannot reject H0")
}
cat(" with an accepted risk α =",alpha,"\n")
```



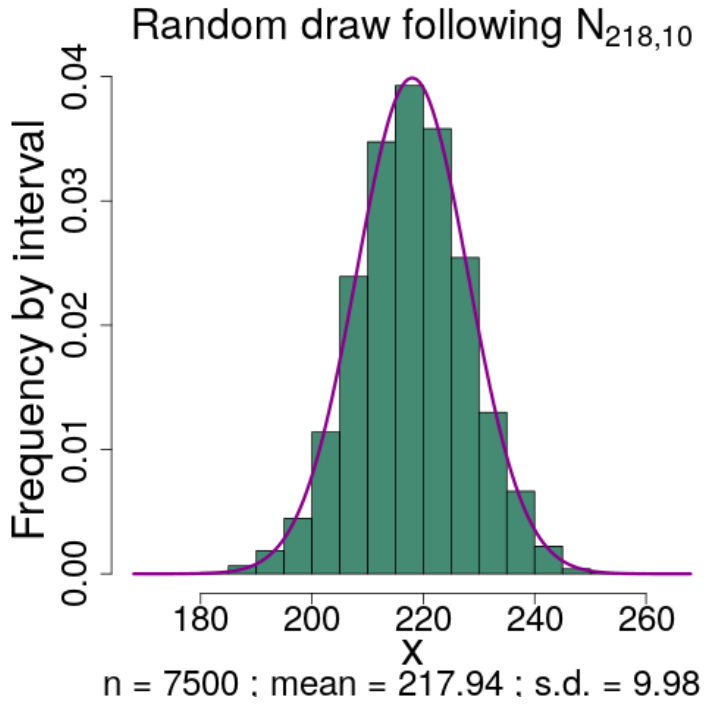
[1] inspired by S.D. Jayakar (1962), *Cah. Biol. Mar.* **3**, 129–136.

Sampling in the normal law

```r
# Draw one random value with p.d.
# following the normal law
mu = 218
sd = 10
# Only once
val <- rnorm(1,m=mu,s=sd)
cat("\nExample of random value:\n")
cat("X↦N(μ =",mu,",σ =",sd,") v =",val,"\n")
```




```r
# Draw many random values with p.d.
# following the normal law
mu = 218
sd = 10
n = 7500
listx <- rnorm(n,m=mu,s=sd)
cat("\n",n,"random values:\n")
cat("mean = ",mean(listx),"\n")
```

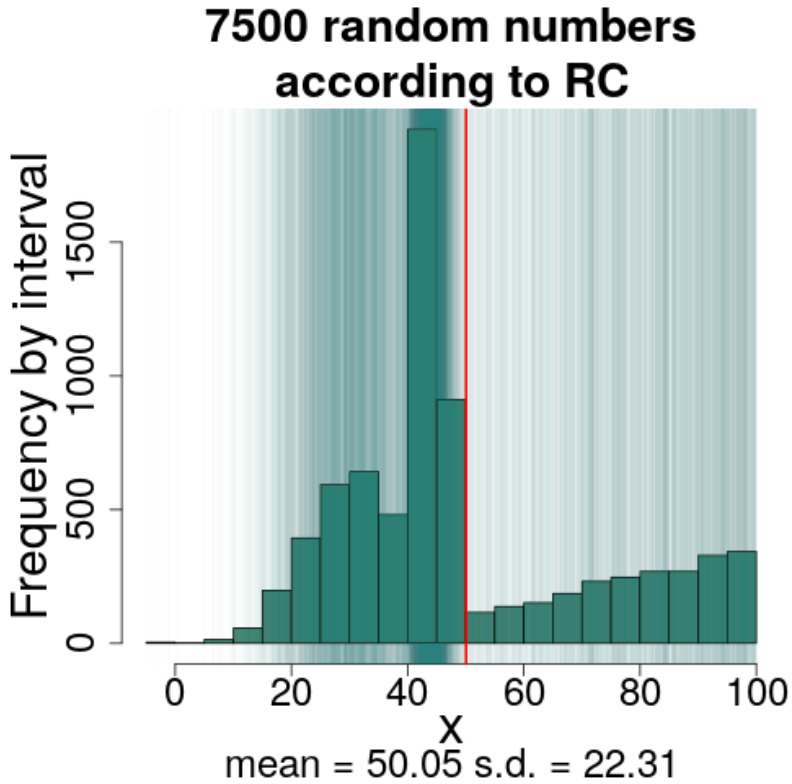


No surprise: $\bar{x} \to \mu$ and $s \to \sigma$

$\hat{\mu} \approx \bar{x}$ and $\hat{\sigma} \approx \sigma$

Let us define a new law: the RC law

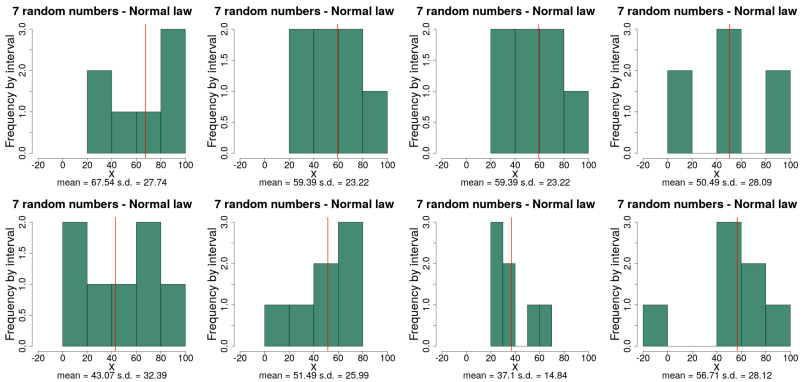
```
rc <- function(nb) {
  x=c()
  rl=sample(c(1,2,3),size=nb,replace=T)
  for(r in rl) {
    if(r==1){
      x=c(x,rnorm(n=1,m=30,s=8))
    }
    if(r==2){
      x=c(x,rnorm(n=1,m=44,s=2))
    }
    if(r==3){
      x=c(x,100-62*abs(rnorm(n=1,m=0,s=0.5)))
    }
  }
  x
}
n = 7500
listx <- rc(n)

hist(listx,breaks=25,col="aquamarine4",
     main=paste(n,"random numbers according to RC"),
     xlab="x",ylab="Frequency by interval",
     sub=paste("mean =",round(mean(listx),2),"s.d. =",round(sd(listx),2))
    )
abline(v=listx,col=adjustcolor("aquamarine4",alpha=0.01),lwd=2)
abline(v=mean(listx),col="red",lwd=2)
```



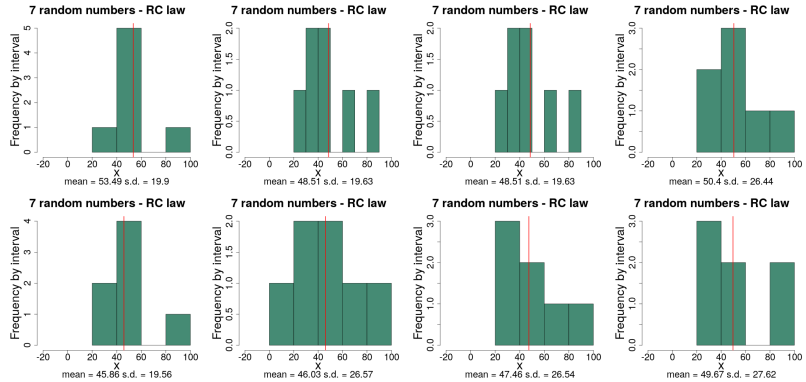
Small samples

Same $\mu = 50.05$ and $\sigma = 22.31$

Normal Law

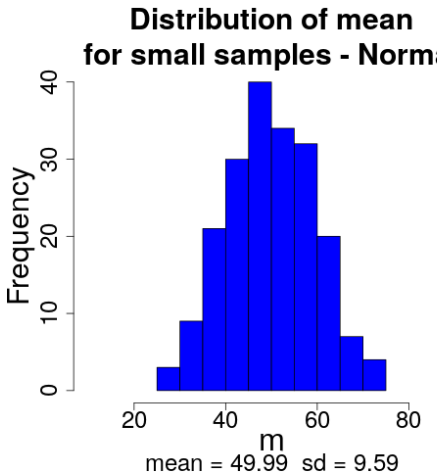


RC Law

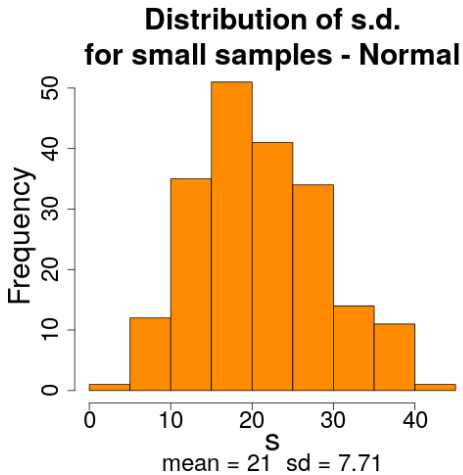


etc. $\rightarrow m = \bar{x}$ and s for each sample

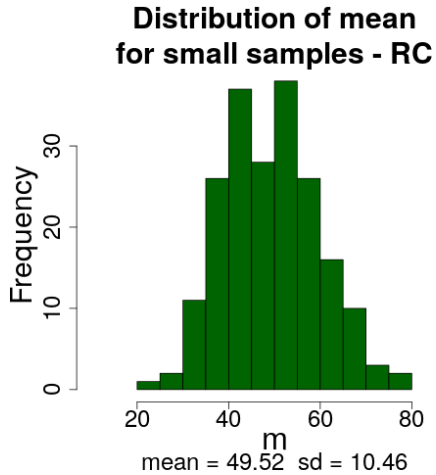
etc. $\rightarrow m = \bar{x}$ and s for each sample



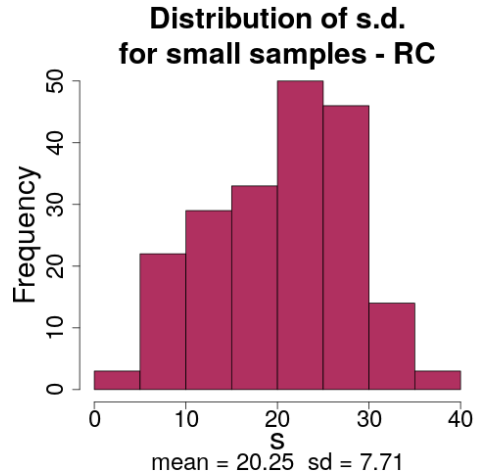
$\bar{m} = 49.99$; $s_m = 9.59$



$\bar{s} = 21.00$; $s_s = 7.71$

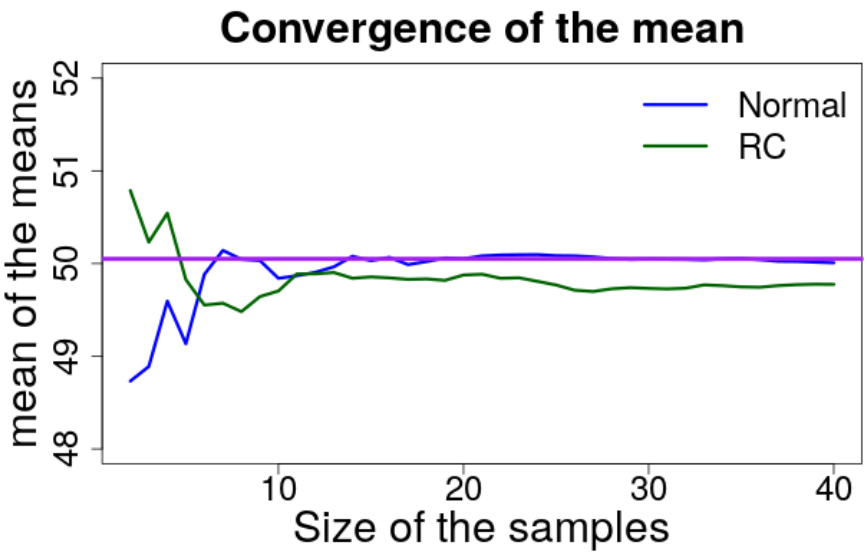


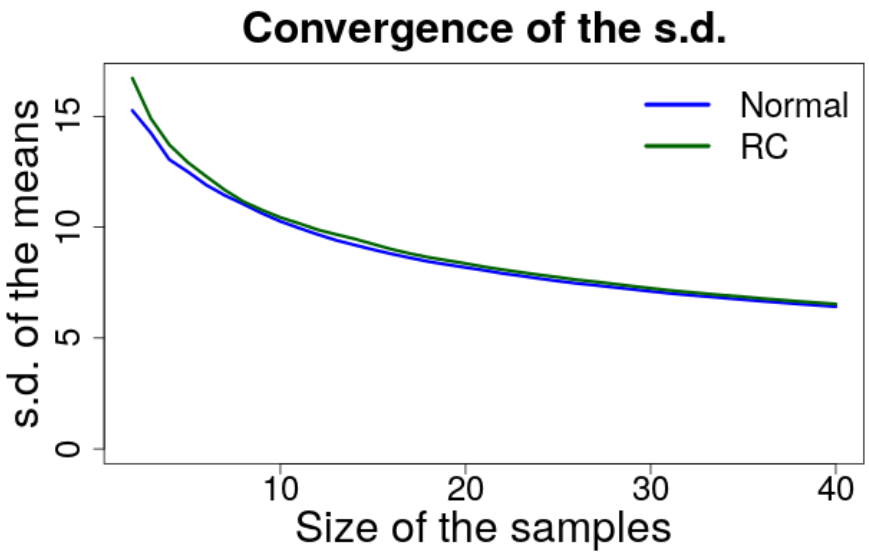
$\bar{m} = 49.52$; $s_m = 10.46$



$\bar{s} = 20.25$; $s_s = 7.71$

Convergence of $\bar{X}$





$\bar{x} \to \mu$

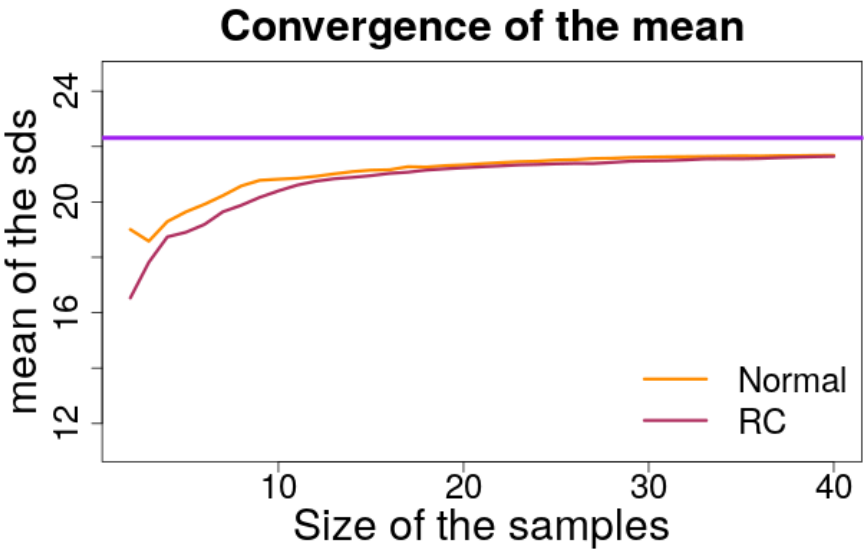
$s_{\bar{X}} \to 0$

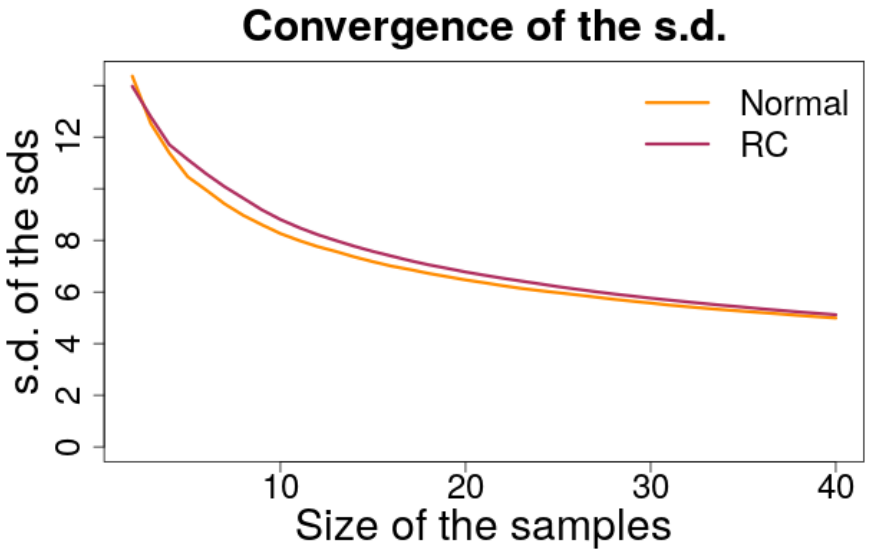
Increasing sample size : Central Limit Theorem

$$\lim_{n \to +\infty} \bar{X} = N_{\mu, \sigma/\sqrt{n}}$$

$$\hat{\mu} = \bar{x}$$

Convergence of $\bar{S}$





$$\bar{s} \to \sqrt{\frac{n-1}{n}}\sigma$$

$$s_{\bar{S}} \to 0$$

Whatever the sample size :

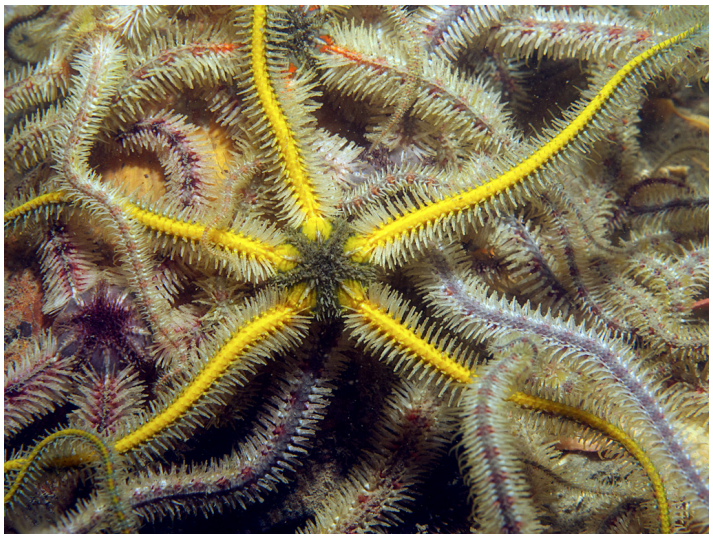
$$\hat{\sigma} = \sqrt{\frac{n}{n-1}}s$$

Confidence Interval

Question[2]: How credible is a parameter estimate?

Example: dry weight of *Ophiothrix fragilis* females

$$n = 161, m = 696.00 \text{ mg}, s = 134.23 \text{ mg}.$$



Estimates for the population:

$$\hat{\mu} = m \qquad \hat{\sigma} = s\sqrt{\frac{n}{n-1}} = 134.65 \text{ mg}$$

Confidence interval at 95%, *i.e.* $\alpha = 0.05$

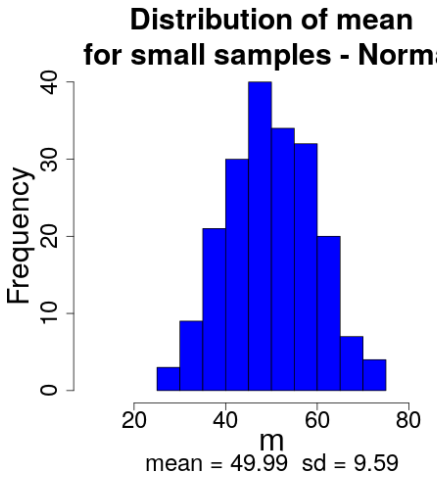
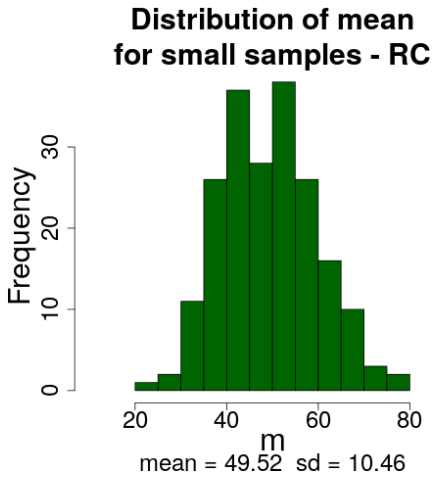
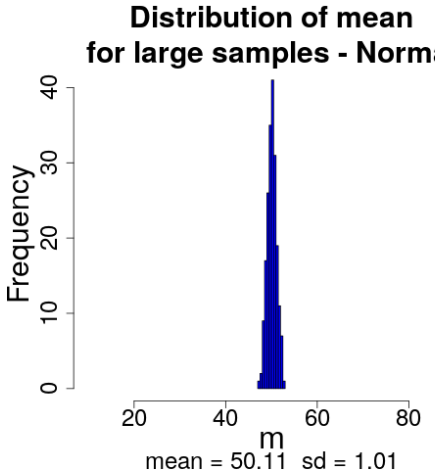
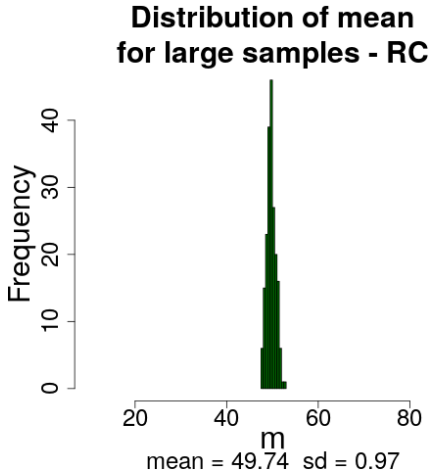
$$\hat{\mu} \pm u_{\alpha/2}\frac{\sigma}{\sqrt{n}}$$

```
# Measures
n0f = 161
m0f = 696.00
s0f = 134.23
# Estimates for the population
mu0f = m0f
si0f = s0f * sqrt(n0f/(n0f-1))
```

```
# Confidence interval
alpha = 0.05
rnglo = qnorm(alpha/2,mu0f,si0f / sqrt(n0f))
rnghi = qnorm(1-alpha/2,mu0f,si0f / sqrt(n0f))
cat("Confidence interval at 95%: [",rnglo,":",rnghi,"]\n")
```

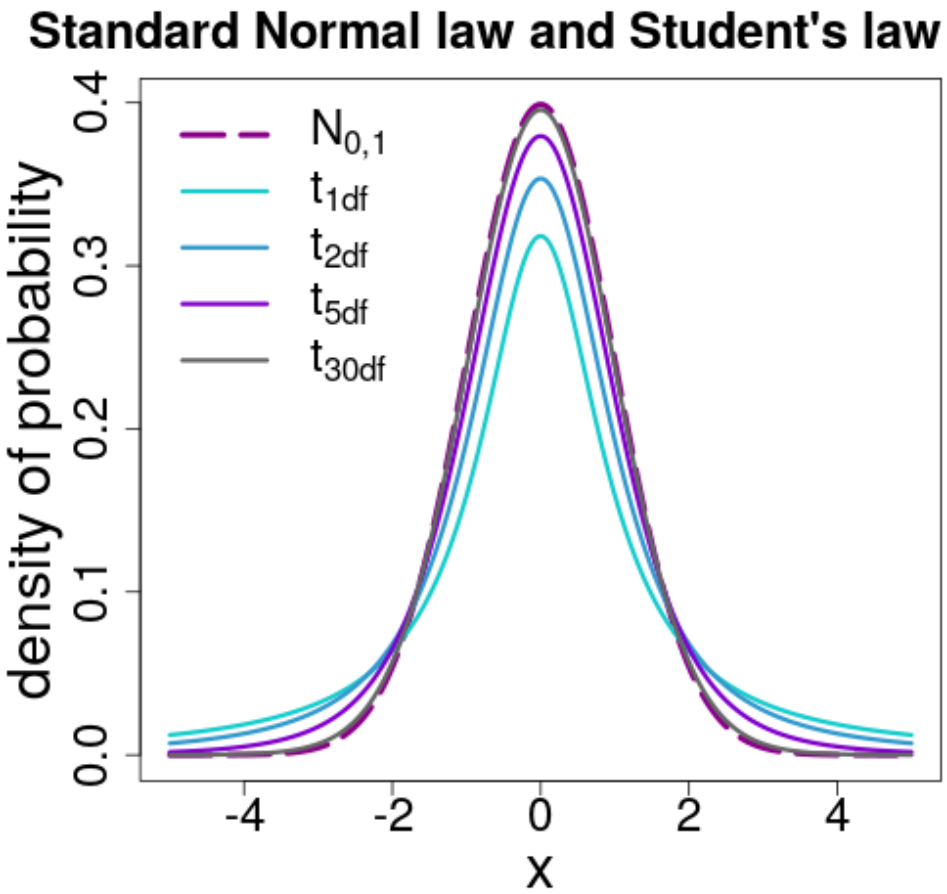
[2] Inspired by A. Lefebvre, D. Davoult, F. Gentil, M. A. Janquin (1999) *Hydrobiologia* **414**:25–34.

Laws and sample size: the sample, that's simple

	Normal Law	Abnormal Law
Small sample		
	$\bar{X} \mapsto \mu + \frac{\sigma}{\sqrt{n}} \times t_{n-1\,df}$	No rule
Large sample		
	$\bar{X} \mapsto N_{\mu,\sigma/\sqrt{n}}$	$\bar{X} \mapsto N_{\mu,\sigma/\sqrt{n}}$

Student's Law

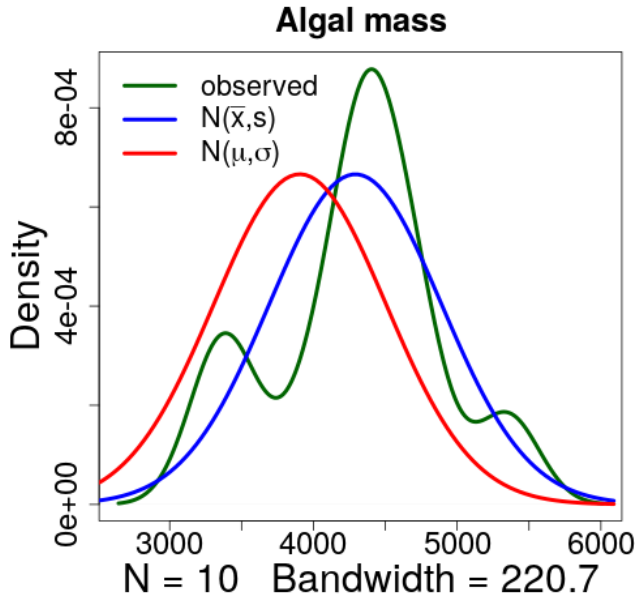
```
valist <- seq(-5,5,length.out=200)
d = 2 # degree of freedom
plot(valist,dt(valist,df=d),type="l")
```



Student's law is like a Normal law with heavy tails (for small d.f.)

Example of application

The biomass of green algae in an estuary is known to follow a normal law with $\mu = 3906$ g/m^2; $\sigma = 599$ g/m^2.
We suspect a pollution to increase this mass.
We perform 10 measures and obtain the data in file `MassAlg01.tab`
Should we confirm that the mass has been increased ?




```
# Under H0, the sample follows the normal law of the population:
cat("H0 : mass does not exceed the usual norm.\n")
muA = 3906 ; sigmaA = 599

# Observed data:
massalg <- read.table("MassAlg01.tab",header=F)
nbmes = nrow(massalg)
measM = mean(massalg$V1)
cat("Mean mass on",nbmes,"measures:",round(measM,2),"\n")
plot(density(massalg$V1),main="Algal mass")    # Rather nice, but not so rigorous

# Under H0, (X̄-μ)/(σ/√n) ↦ t(n-1 df)
alpha = 0.05
pv = 1-pt((measM-muA)/(sigmaA/sqrt(nbmes)),df=nbmes-1)
cat("p-value =",pv,"; ")
if(pv<alpha) {
  cat("we can reject H0")
} else {
  cat("we cannot reject H0")
}
cat(" with a type I error risk α =",alpha,"\n\n")
```

Two changes

What if σ is not known?

If we do not know σ, we replace it by our estimate: $\hat{\sigma} = s\sqrt{\frac{n}{n-1}}$

Warning ! R computes sd with $(n-1)$ so, it is in fact $\hat{\sigma}$!

```
cat("The standard deviation for the sample is : s =",
     sd(massalg$V1)*sqrt((nbmes-1)/nbmes),"\n")
estimsig = sd(massalg$V1)
cat("The estimated population s.d. is", estimsig,"\n")

# recompute p-value using this estimate
pv = 1-pt((measM-muA)/(estimsig/sqrt(nbmes)),df=nbmes-1)

cat("Using this estimate, we find p-value =",pv,"\n")
```

What if we are lazy?

Use The R implementation of the t test!

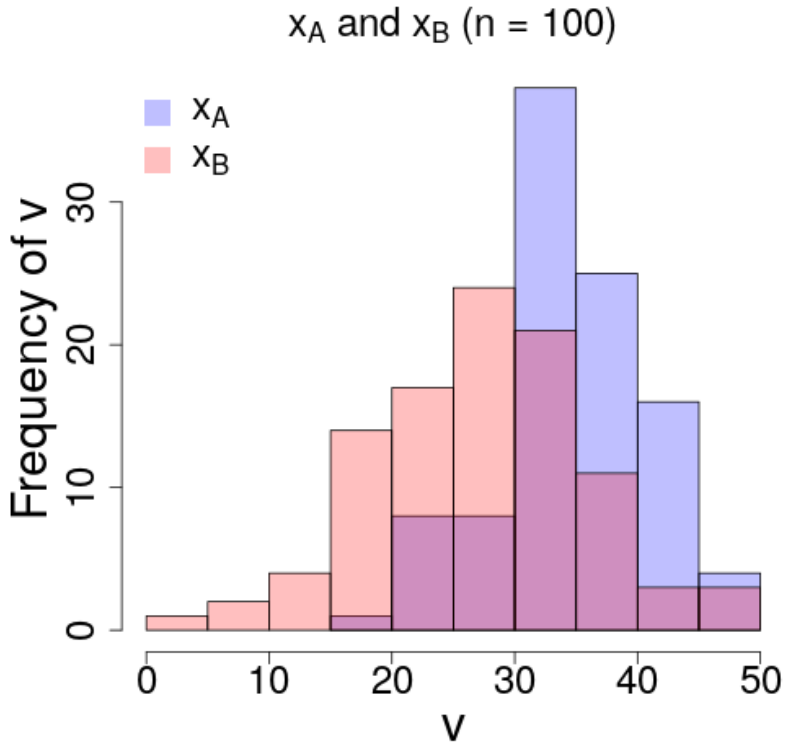
```
cat("\n[R] : function t.test\n")
tt <- t.test(massalg$V1,mu=muA,alternative="greater")
print(tt)
```

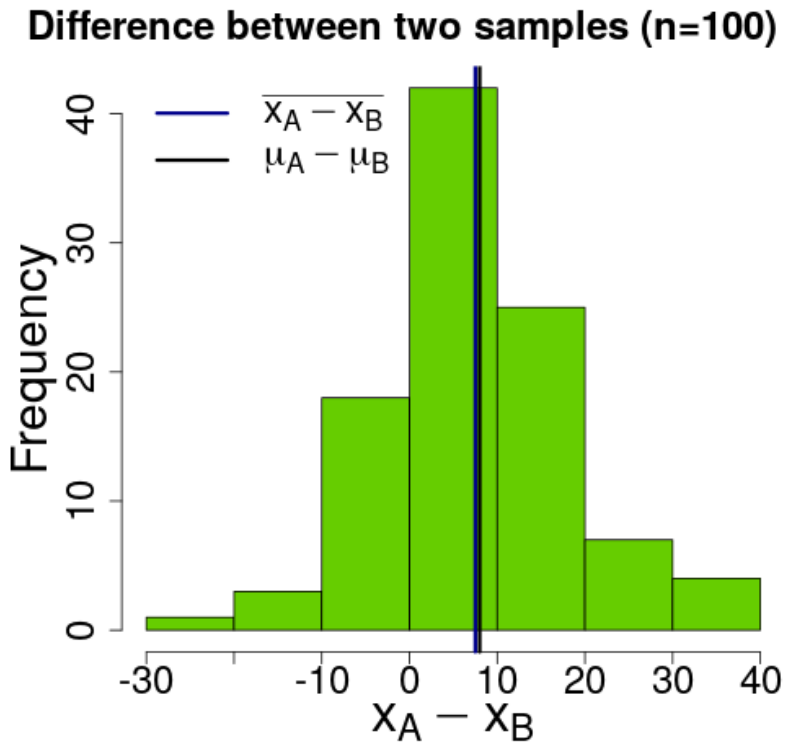
The *real* experimental situation

In real experiments, we usually have two (or more) samples to compare

Example: $\mu_A = 35$; $\sigma_A = 6$; $\mu_B = 27$; $\sigma_B = 9$

Two samples and their difference





Observed $\overline{x_A - x_B} = 7.51$ $s_{x_A - x_B} = 10.67$

Expected $\mu_{x_A - x_B} = \mu_A - \mu_B = 8$ $\sigma_{x_A - x_B} = \sqrt{\sigma_A^2 + \sigma_B^2} = 10.82$

Distribution and field of application

$\overline{x_A - x_B}$ and $\overline{x_A} - \overline{x_B}$ have the same mean and s.d., so how to choose?

- $\overline{x_A} - \overline{x_B}$ can be computed for unpaired samples, even if $n_A \neq n_B$

$$\overline{x_A} - \overline{x_B} \mapsto (\mu_A - \mu_B) + \sqrt{\frac{\sigma_A^2}{n_A} + \frac{\sigma_B^2}{n_B}} \times t_{n_A - 1 + n_B - 1}$$

Usually we do not know σ_A and σ_B:

$$\frac{\sigma_A^2}{n_A} + \frac{\sigma_B^2}{n_B} \approx (n_A s_A^2 + n_B s_B^2) \times \frac{1/n_A + 1/n_B}{n_A + n_B - 2}$$

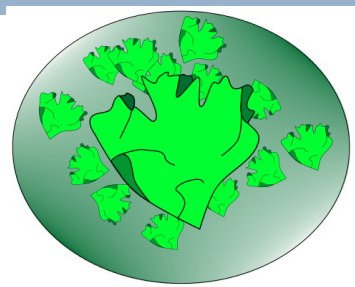
- $\overline{x_A - x_B}$ is good for pairwise differences (thus $n_A = n_B = n$)

$$\overline{x_A - x_B} \mapsto (\mu_A - \mu_B) + \sqrt{\frac{\sigma_A^2 + \sigma_B^2}{n}} \times t_{n-1}$$

Usually we do not know σ_A and σ_B:

$$\frac{\sigma_A^2 + \sigma_B^2}{n} \approx \frac{s_A^2 + s_B^2}{n - 1}$$

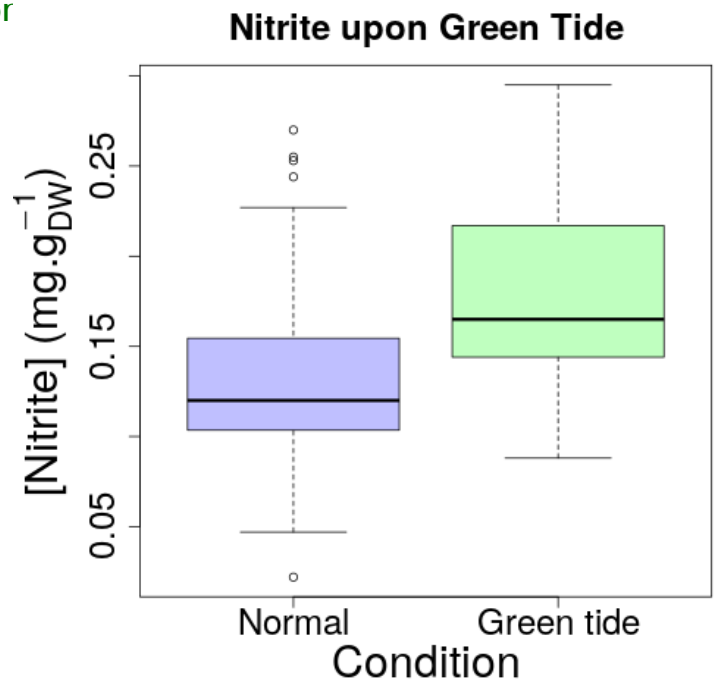
Application: t for two



Nitrite concentration in Ulva strains from (N) Normal and (G) Green tide areas[3]

H_0: areas are samples out of one common Ulva population, $\mu_N = \mu_G$

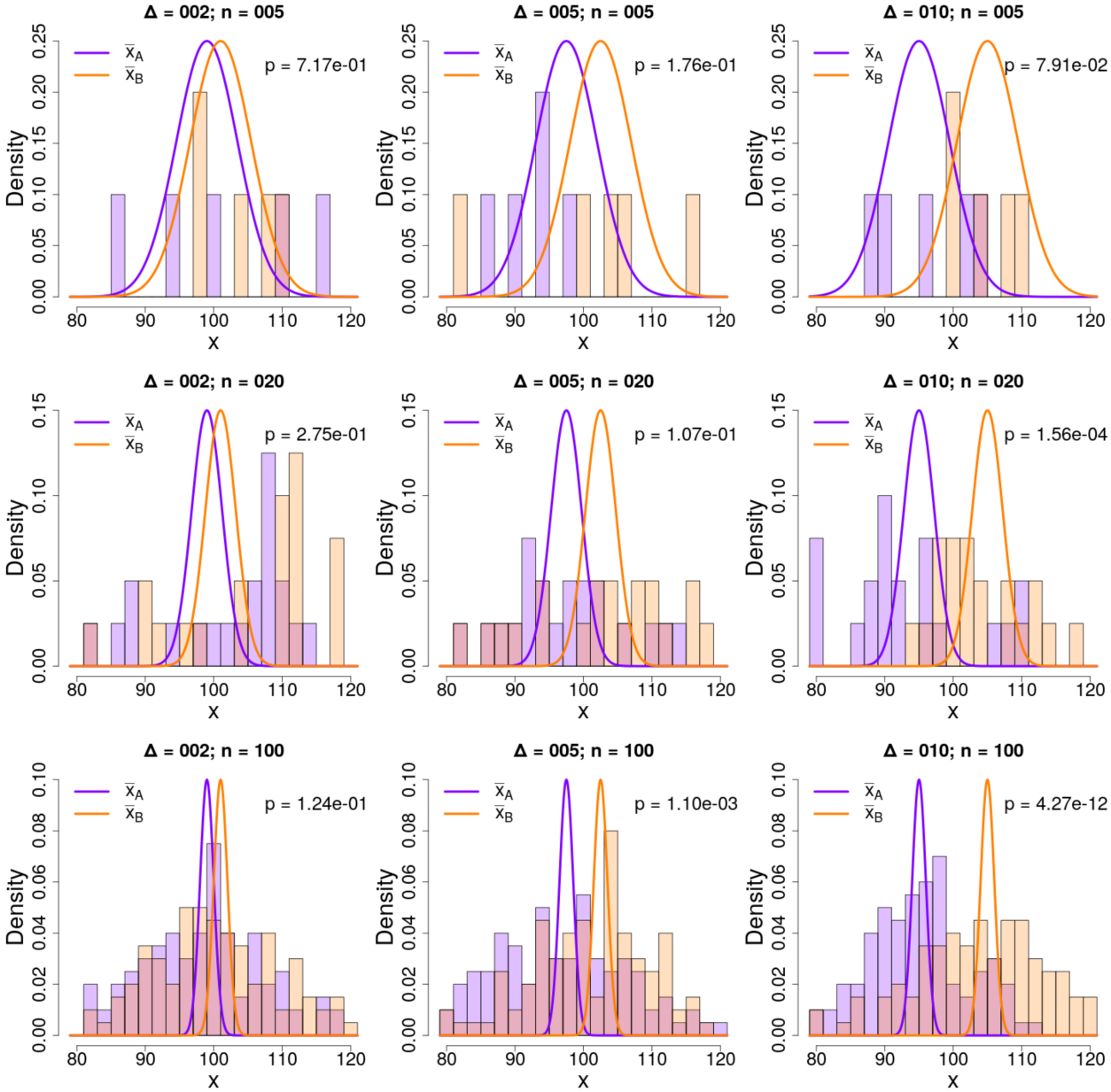
```
nit <- read.table("NitriteGreenTide.tab",header=T,sep="\t")
attach(nit)
boxplot(Nitrite~cond, main="Nitrite upon Green Tide",xlab="Cor
     ylab=expression(paste("[Nitrite] (",mg.g[DW]^{-1},")")))
```




```
# Compute everything
nN = length(Nitrite[type=="Normal"])
mN = mean(Nitrite[type=="Normal"])
sN = sd(Nitrite[type=="Normal"])*sqrt((nN-1)/nN)
nG = length(Nitrite[type=="Green tide"])
mG = mean(Nitrite[type=="Green tide"])
sG = sd(Nitrite[type=="Green tide"])*sqrt((nG-1)/nG)

De = sqrt((nN*sN**2+nG*sG**2)*(1/nN+1/nG)/(nN+nG-2))

# Bilateral test
pv=2*pt(-abs(mN-mG)/De,df=nN+nG-2)
cat("    Difference = ",mN-mG,"; ν = (nN - 1) + (nG -1) = ",nN-1+nG-1,"\n",sep="")
cat("    p-value =",pv,"\n")

# Another, less fun, code
t.test(Nitrite~type,var.equal=T)
```

[3] A. Fort, C. Mannion, J.M. Fariñas-Franco, R. Sulpice (2020) *Sci. Tot. Envir.* **698** 134337

Why experimentalists hate statisticians



We cannot change anything to $\mu_A, \mu_B, \sigma_{x_A}, \sigma_{x_B}$

The only parameter we can tune:

$$\sigma_{\bar{x}} = \frac{\sigma_x}{\sqrt{n}}$$

Increasing n decreases dispersion of means

$\Leftrightarrow$

The larger n, the best we see (small) effects

$\Leftrightarrow$

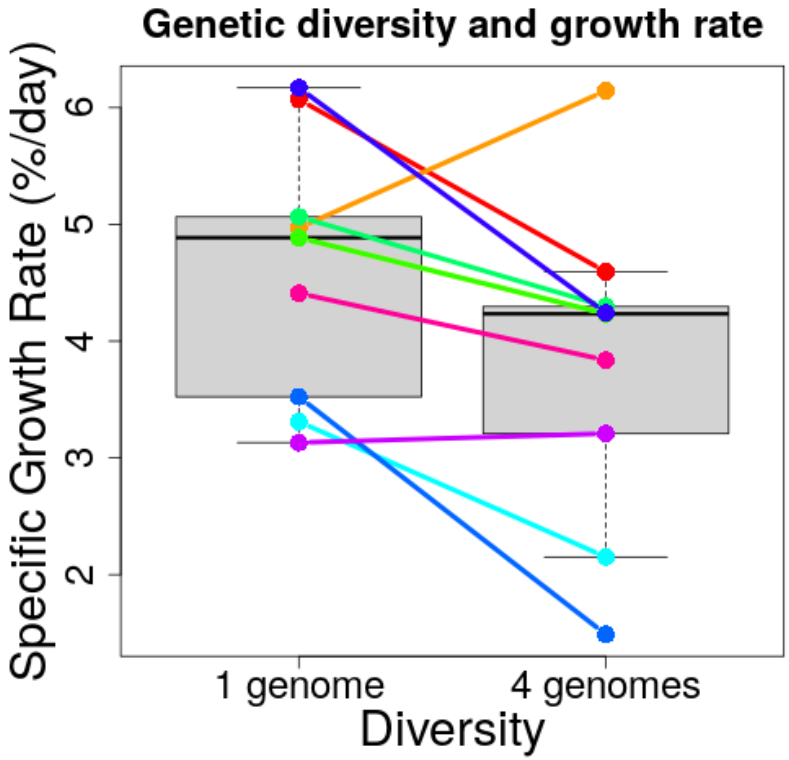
More bench required!

Improve your power by using the pairing information

Example: Genotypic diversity in *Agarophyton chilense*[4]

Question: Effect on growth rate (supposed normally-distributed)?

```
sgrdata <- read.table("SGR_Chaica.tab",header=T)
attach(sgrdata)
summary(sgrdata[,2:3])

# default: unpaired
cat("\nNot taking pairs into account:\n")
cat(" Difference of means =",mean(T1G)-mean(T4G),"\n")
t.test(T1G,T4G,var.equal=T)

# actually, measures on the same genotype => paired
cat("\nTaking pairs (Genotypes) into account:\n")
cat(" Mean difference =",mean(T1G-T4G),"\n")
t.test(T1G,T4G,var.equal=T,paired=T)

# draw
boxplot(T1G,T4G,col=c("purple","orange"),
  main="Genetic diversity and growth rate",
  xlab="Diversity",ylab="Specific Growth Rate (%/day)",
  names=c("1 genome","4 genomes") )
```



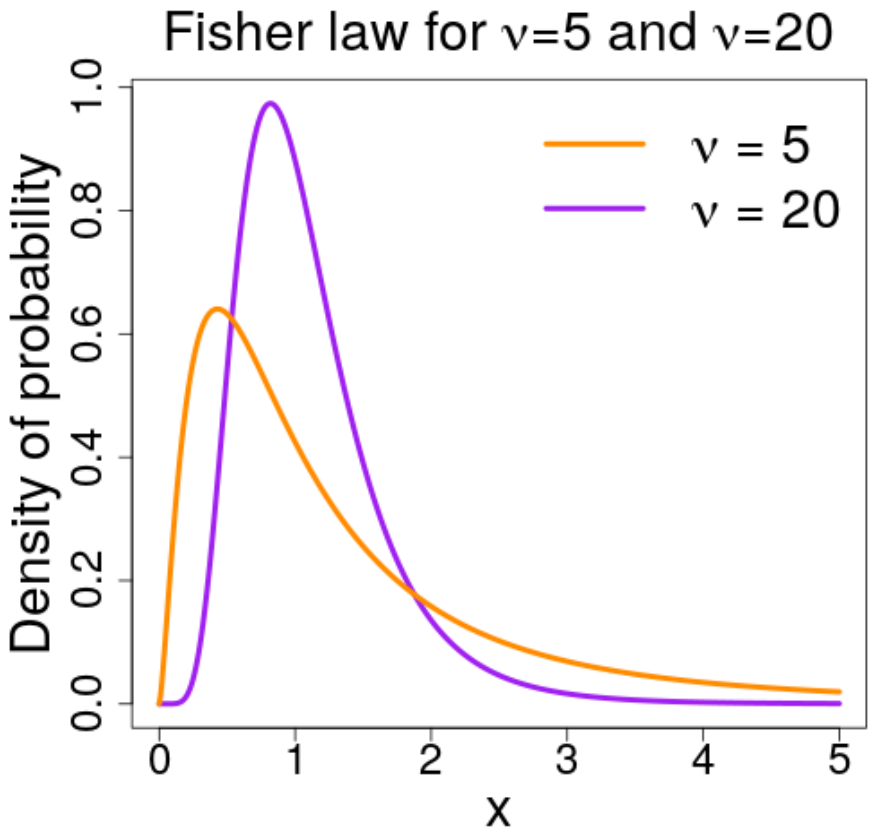
See the difference? (Hint: pay attention to the df)

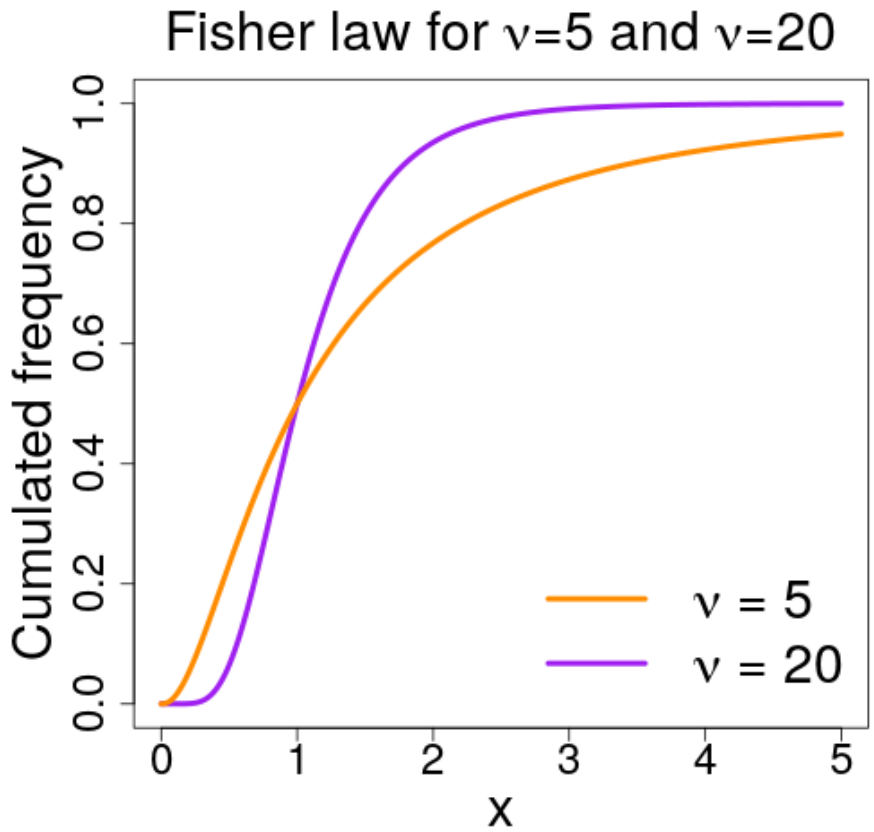
[4] Data dishonestly extracted from S. Usandizaga, A.H. Buschmann, C. Camus, J.L. Kappes, S. Arnaud-Haond, S. Mauger, M. Valero and M.L. Guillemin (2019) *Evol. Appl.* **00**:1–13.

How do we know if variances are equal?

Fisher:

- One population with variance σ^2
- Two samples with variance s_A^2 and s_B^2
- Compute the law for $\frac{s_A^2}{s_B^2}$





Example

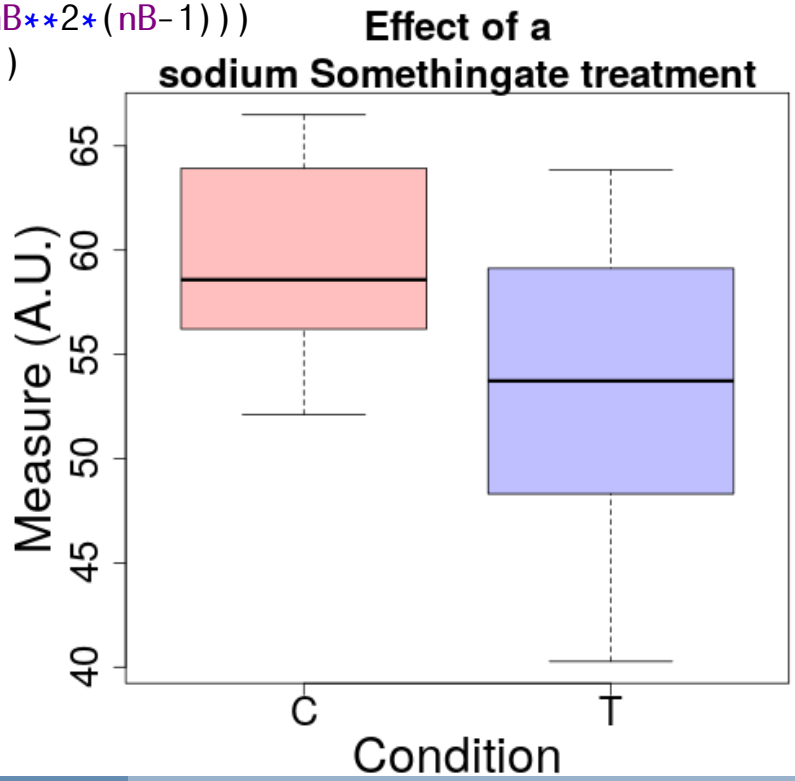
```
muA = 60 ; muB = 55
sigmA = 3 ; sigmB = 6
nA = 12 ; nB =  15

listA <- rnorm(nA, m=muA, s=sigmA)
listB <- rnorm(nB, m=muB, s=sigmB)

mA = mean(listA)
mB = mean(listB)
sA = sd(listA)
sB = sd(listB)
cat("sA =",sA,"; sB =",sB,"; F =",(sA/sB)**2)

# Bilateral test
pv = 2 * pf((sA/sB)**2, df1=nA-1, df2=nB-1)
cat(" p-value =",pv,"\n")

# Too easy
cat("\n[R] : function var.test\n")
var.test(listA, listB)  # test F, same p-value as var.test(listB, listA)
```

So, if variances are not equal?

Welch:

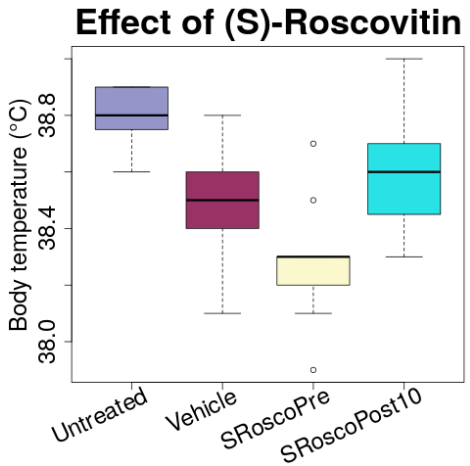
- $\hat{\sigma}_A$ and $\hat{\sigma}_B \rightarrow \sigma^2 = \frac{\hat{\sigma}_A^2}{n_A} + \frac{\hat{\sigma}_B^2}{n_B}$
- change the degree of freedom to something complicated
  $\nu < n_A + n_B - 2$ and $\nu \rightarrow n_A + n_B - 2$ when $s_A \rightarrow s_B$

```
cat("Welch correction\n")
sA = sd(listA)
sB = sd(listB)
nu = (sA**2/nA+sB**2/nB)**2/(sA**4/(nA**2*(nA-1))+sB**4/(nB**2*(nB-1)))
cat("Estimates of σA and σB :",sA,"and",sB,"; ν =",nu,"\n")

De = sqrt((sA**2/nA+sB**2/nB))
pv = 2*pt(-abs(mA-mB)/De,df=nu)
cat("Difference =",mA-mB,"; p-value =",pv,"\n")

# Create Names and Vals as vectors
Names <- c(rep("A",length(listA)),rep("B",length(listB)))
Vals <- c(listA,listB)

t.test(Vals~Names,var.equal=F)

boxplot(Vals~Names,col=c(rgb(1,0,0,0.25),rgb(0,0,1,0.25)),
        main="Effect of a sodium Somethingate treatment",
        xlab="Condition",ylab="Measure (A.U.)"
       )
```



Comparing more than two samples

Example: effect of (S)-roscovitine on metabolic parameters in rats[5].



Body temperature for 4 conditions, with $11 \leq n \leq 13$

Untreated	Vehicle (Control)	(S)-Roscovitin Pre-occlusion	(S)Roscovitin 10 mg/kg/h Post-occlusion
38.9	38.8	37.9	38.7
38.7	38.8	38.5	38.6
38.6	38.3	38.2	38.5
38.8	38.5	38.3	39.0
38.9	38.1	38.3	38.9
38.8	38.8	38.7	38.3
38.9	38.5	38.3	38.6
38.8	38.4	38.3	38.4
38.9	38.6	38.2	38.6
38.7	38.4	38.1	38.7
38.8	38.6	38.3	38.5
38.9	38.6		38.4
	38.4		

First idea: One test for each pair of conditions

```
# Read data
neurokin <- read.table("Neurokin_01.tab",header=T)
attach(neurokin)

# Plot
boxplot(neurokin)
```

```
#perform all 6 tests
t.test(Untreated,Vehicle)
t.test(Untreated,SRoscoPre)
t.test(Untreated,SRoscoPost10)
t.test(Vehicle,SRoscoPre)
t.test(Vehicle,SRoscoPost10)
t.test(SRoscoPre,SRoscoPost10)
```

[5] Data reconstructed from
B. Menn, S. Bach, T. Blevins, M. Campbell, L. Meijer, S. Timsit (2010) *PLoS ONE* **5(8)** e12117.

Interlude: reshaping data

Some functions require an appropriate data format.

"wide" format

Untreated	Vehicle	SRoscoPre	SRoscoPost10
38.9	38.8	37.9	38.7
38.7	38.8	38.5	38.6
38.6	38.3	38.2	38.5
38.8	38.5	38.3	39.0
38.9	38.1	38.3	38.9
38.8	38.8	38.7	38.3
38.9	38.5	38.3	38.6
38.8	38.4	38.3	38.4
38.9	38.6	38.2	38.6
38.7	38.4	38.1	38.7
38.8	38.6	38.3	38.5
38.9	38.6		38.4
	38.4		

"long" format

	Condition	Temperature
1	Untreated	38.9
2	Untreated	38.7
3	Untreated	38.6
...	...	...
10	Untreated	38.7
11	Untreated	38.8
12	Untreated	38.9
14	Vehicle	38.8
15	Vehicle	38.8
16	Vehicle	38.3
...	...	...
24	Vehicle	38.6
25	Vehicle	38.6
26	Vehicle	38.4
27	SRoscoPre	37.9
28	SRoscoPre	38.5
29	SRoscoPre	38.2
...	...	...
35	SRoscoPre	38.2
36	SRoscoPre	38.1
37	SRoscoPre	38.3
40	SRoscoPost10	38.7
41	SRoscoPost10	38.6
42	SRoscoPost10	38.5
...	...	...
49	SRoscoPost10	38.7
50	SRoscoPost10	38.5
51	SRoscoPost10	38.4

From wide to long:

```
neuroCT <- stack(neurokin)
neuroCT <- neuroCT[!is.na(neuroCT$values),c(2,1)]
colnames(neuroCT) <- c("Condition","Temperature")
```

or:

```
library(reshape2)
neuroCT <- melt(neurokin,variable.name="Condition",
                value.name="Temperature",id=NULL,na.rm=T)
```

Performing many t-test is pleasurable, but...

1. H_0 should be: "All samples are drawn out of the same population"
   $\Rightarrow$ same σ for all conditions
   but : each t-test computes its own $\hat{\sigma}$

   p value over- or under-estimated, depending on $\hat{\sigma_{ij}}$ *vs* σ
2. p value says how probable are the data under H_0
   but : each t-test computes how probable is its own subset of two samples
   - Risk $\alpha_p = 0.05$: error on 1 pairwise test
   - $\Rightarrow P_p = 1 - \alpha_p = 0.95$ for each pairwise test to be correct
   - 6 simultaneously correct tests: $P_M = P_p^6 = (1-\alpha_p)^6 \approx 0.735$
   - $\Rightarrow$ we accept a risk $\alpha_M \approx 0.265$ of error for the whole analysis... do we?

   Rule: pairwise p values must be adjusted for multiple testing

```r
# Read data
neurokin <- read.table("Neurokin_01.tab",
                                    header=T)
attach(neurokin)

# Re-format data
stneu <- stack(neurokin)
stneu <- stneu[!is.na(stneu$values),c(2,1)]
colnames(stneu) <- c("Condition","Temperature")
```

```r
#perform all 6 tests
pairwise.t.test(stneu$Temperature,
                g=stneu$Condition)
```

```
    Pairwise comparisons using t tests with pooled SD

data:  stneu$Temperature and stneu$Condition

            Untreated Vehicle SRoscoPre
Vehicle     0.00152   -       -
SRoscoPre   1.3e-07   0.00808 -
SRoscoPost10 0.01697  0.30506 0.00083

P value adjustment method: holm
```

Analysis Of Variance

Reminder: under H_0,

The variation of mean between classes is not higher than the mean of variation within each class

$$N = \sum_{j=1}^{k} n_j \qquad M = \frac{1}{N} \sum_{j=1}^{k} \sum_{i=1}^{n_j} x_{i,j}$$

Mean for each class:

$$m_j = \frac{1}{n_j} \sum_{i=1}^{n_j} x_{i,j}$$

Variance within each class:

$$s_j^2 = \frac{1}{n_j - 1} \sum_{i=1}^{n_j} (x_{i,j} - m_j)^2$$

Variance due to Factor:

$$S_F^2 = \frac{1}{k-1} \sum_{j=1}^{k} n_j (m_j - M)^2$$

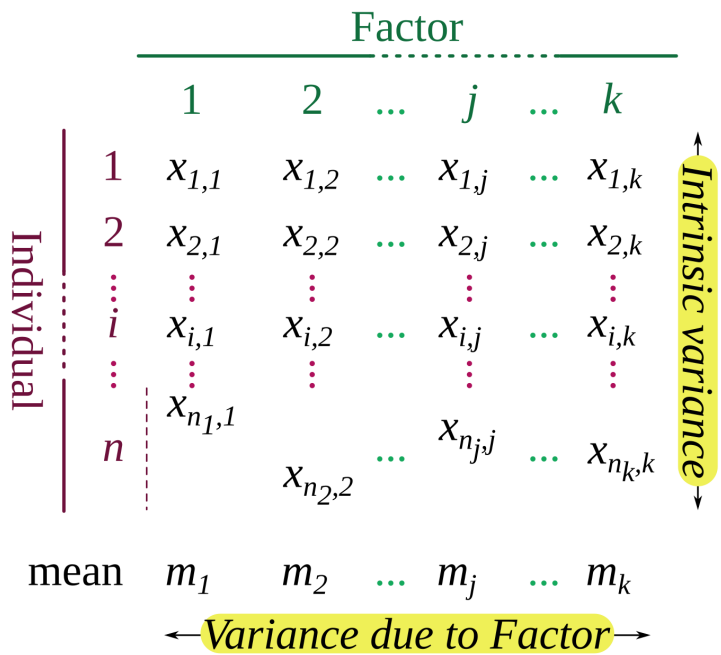
Intrinsic variance:

$$S_I^2 = \frac{1}{N-k} \sum_{j=1}^{k} (n_j - 1) s_j^2$$

Hence the question: is S_F^2 significantly larger than S_I^2?
Hence the question: how do we compare variances?
Hence the answer: with a Fisher test.

Let's write some code!

```
neurokin <- read.table("Neurokin_01.tab",header=T)
# convert to long format
stneu <- stack(neurokin)
stneu <- stneu[!is.na(stneu$values),c(2,1)]
colnames(stneu) <- c("Condition","Temperature")
attach(stneu)

# basic numbers
N = nrow(stneu)
co = levels(Condition)
k = length(co)
M = mean(Temperature)
# loop to sum into S2F and S2I
S2F = 0
S2I = 0
for (j in 1:k) {
  xj <- stneu[stneu[,1]==co[j],2]
  nj = length(xj)
  S2F = S2F + nj / (k-1) * ( mean(xj) - M ) ** 2
  S2I = S2I + (nj-1) / (N-k) * var(xj)
}
# perform a Fisher test
Ffi = S2F/S2I
cat("k =",k," N =",N," k-1 =",k-1," N-k =",N-k,"\n")
cat("S2b =",S2F," S2w =",S2I," Ffi =",Ffi,"\n")
pv <- 1 - pf(Ffi, df1=k-1, df2=N-k)
cat("p-value for AnOVa test: p =",pv,"\n\n")
```



Let's avoid writing so much code!

```
# use the R function aov
summary(aov(Temperature ~ Condition))
```

What we learn from a one-way AnOVa

A typical one-way AnOVa test:

- Choose α
- H_0: All samples come from the same population distibuted according to the same Normal law:
  $\mu_1 = \mu_2 = ... = \mu_k$, $\sigma_1 = \sigma_2 = ... = \sigma_k$
- Compute $F = S_F^2/S_I^2$, then p
  - if $p > \alpha$, then H_0 cannot be rejected
    $\Rightarrow$ No effect can be demonstrated for the factor
  - if $p < \alpha$, then H_0 must be rejected
    $\Rightarrow$ (provided normality and homeosedasticity hold)
    The factor affects μ, at least between two classes

One-way AnOVa test does not tell us which classes differ!

To discover where is/are the difference(s):

```
neurAOV <- aov(Temperature~Condition)
TukeyHSD(neurAOV)
```

The Tukey's "*post-hoc*" test is similar to the pairwise t-test with p-value correction = "Honnest Significant Difference" ($\approx$ Bonferroni).

Testing normality

Is a given sample likely to have been drawn out of a Normal law ?

Skewness

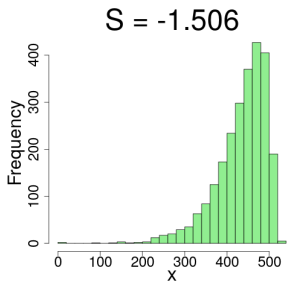
$$S = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^{3/2}}$$

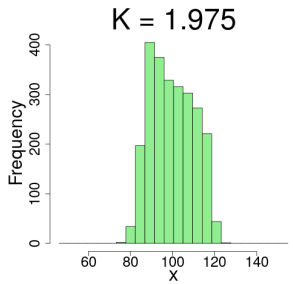
Kurtosis

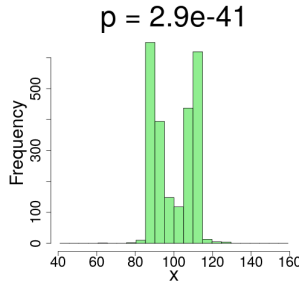
$$K = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^{2}}$$

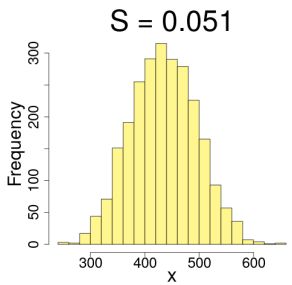
Shapiro-Wilks

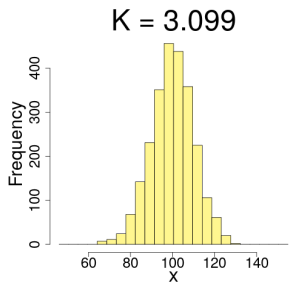
$$W = \frac{\left(\frac{1}{n}\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

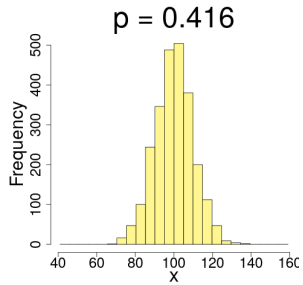


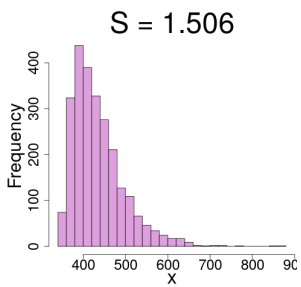


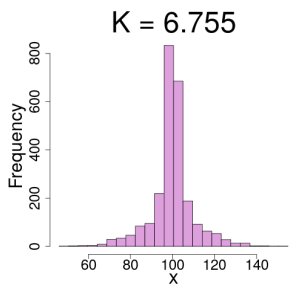


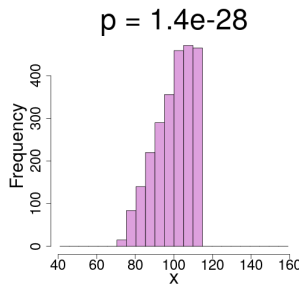












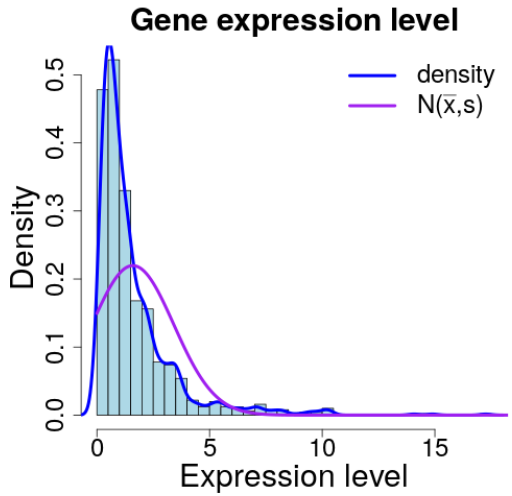
Testing normality

Is a given sample likely to have been drawn out of a Normal law ?

Indices: for Normal law, skewness = 0; kurtosis = 3

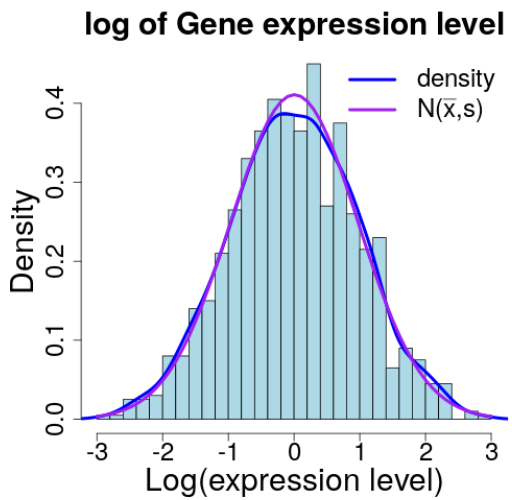
Example: Gene expression level

```
library(moments) # provides skewness an kurtosis

# Is gene expression level likely to follow a normal law?
glev <- read.table("GeneLevel.tab",header = T)
hist(glev$Level,main="Gene expression level distribution",
     breaks=30,xlab="Expression level",freq=F)

cat("  Skewness =",skewness(glev$Level),"\n")
cat("  Kurtosis =",kurtosis(glev$Level),"\n")
stgl = shapiro.test(glev$Level)
cat("  p-value for Shapiro-Wilk test =",stgl$p.value,"\n")

# Is gene expression level likely to follow a log-normal law?
Loglev <- log(glev$Level)
hist(Loglev,main="log of gene expression level distribution",
     breaks=30,xlab="Log(expression level)",freq=F)

cat("\nLog(gene expression level)\n")
cat("  Skewness =",skewness(Loglev),"\n")
cat("  Kurtosis =",kurtosis(Loglev),"\n")
stlgl = shapiro.test(Loglev)
cat("  p-value for Shapiro-Wilk test =",stlgl$p.value,"\n")
```



skew = 3.19, kurt = 17.8

SW test: $p = 1.4^{-39}$



skew = -0.013, kurt = 2.8

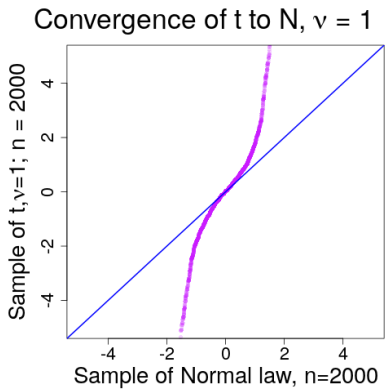
SW test: $p = 0.70$

Testing normality

Is a given sample likely to have been drawn out of a Normal law ?

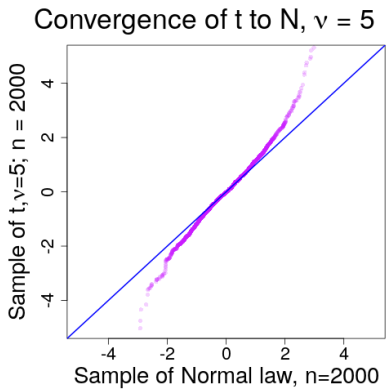
Example: convergence of Student's law to the Normal when ν increases

```
listn <- rnorm(n=2000)
for ( d in c(1,5,30,100) ) {
  listt <- rt(n=2000,df=d)
  qqplot(listn,listt,xlim=c(-5,5),ylim=c(-5,5),pch=16,col=rgb(0.8,0.1,1,0.2),
         main="Convergence of t to N",
         ylab=substitute(paste("Sample of t,",nu,"=",n,"; n = 2000"),list(n=d)),
         xlab="Sample of Normal law, n=2000",
        )
  abline(0,1,col="blue")
  st = shapiro.test(listt)
  cat("ν =",d,"p-value for Shapiro-Wilk test =",st$p.value,"\n")
}
```



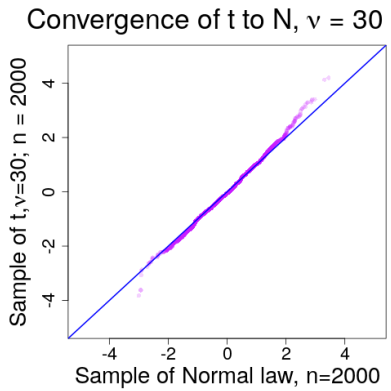
$\nu = 1$

$p = 9.4 \times 10^{-70}$



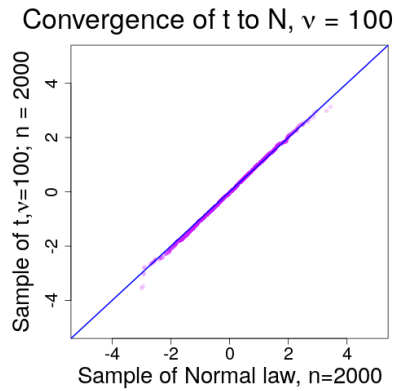
$\nu = 5$

$p = 3.2 \times 10^{-16}$



$\nu = 30$

$p = 3.5 \times 10^{-4}$



$\nu = 100$

$p = 0.96$

Warning

For nearly normal laws, Shapiro-Wilk test result depends on sample size

```
# create functions for Student law
# ... with ν = 20
t20 <- function(n) {rt(n,df=20)}
# ... with ν = 50
t50 <- function(n) {rt(n,df=50)}

# Change sample size
for(ss in c(10,30,100,1000,5000)){

  cat(sprintf(" %4d ",ss))

  for(law in c(rlnorm,t20,t50,rnorm)) {
    pv<-c()
    for(i in seq(1,2000)){
      x<-law(n=ss)
      pv<-c(pv,shapiro.test(x)$p.value)
    }
    # Proportion of normality rejection
    cat(sprintf(" %.3f ",
        length(pv[pv<0.05])/length(pv)))
  }
  cat("\n")
}
```

Proportion of normality rejection

sample size n	log-N	Student $\nu = 20$	Student $\nu = 50$	Normal
10	0.594	0.058	0.053	0.043
30	0.995	0.081	0.064	0.043
100	1.000	0.111	0.077	0.043
1000	1.000	0.400	0.117	0.046
5000	1.000	0.942	0.260	0.036
Expected	← Reject →		← Accept →	
Actual	≈ OK	depends on n		OK

Poisson law for fishermen

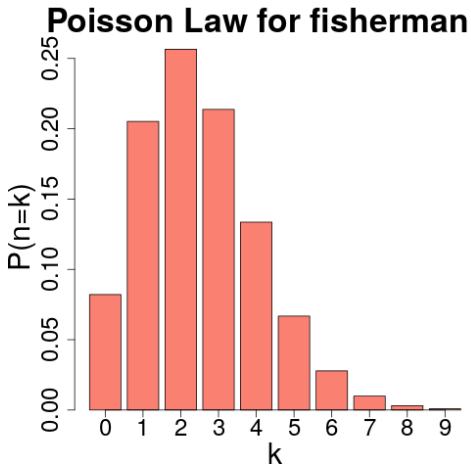
- A fisherman can catch 0 or 1 fish every minute.
- The probability to catch 1 fish during 1 minute is $\beta = 1/12$ (he catches on average 1 fish every 12 minutes).
- We observe him for $\Delta t = 30$ min.

What is the probability he catches $k = 3$ fishes?

$$\text{Poisson law of parameter } \lambda = \beta\Delta t: \quad P_\lambda(n = k) = \frac{\lambda^k}{k!}e^{-\lambda}$$
$$P_\lambda : \mu = \lambda; \sigma^2 = \lambda$$

$$\lambda = \beta\Delta t = 30/12 = 2.5 \text{ fishes during 30 min;}$$
$$P_{2.5}(n = 3) = 2.5^3/3! \times e^{-2.5} \approx 0.214$$

```r
# Poisson probability of 3 with λ = 30/12
dpois(3, lambda = 30/12)

# Plot
k = 0:9
barplot(dpois(k, lambda = 30/12), col = "salmon",
    main = "Poisson Law for fisherman",
    xlab = "k", ylab = "P(n=k)"
)
```



Poisson law for molecular biologists

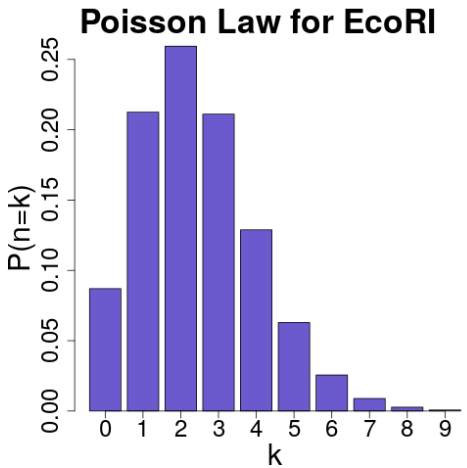
- GAATTC can occur 0 or 1 time at any position in DNA.
- The probability that a 6-mer is GAATTC is $\beta \approx 1/4^6 = 1/4096$ (GAATTC occurs on average 1 time every 4096 bases).
- We observe a sequence of length $l = 10$ kb.

What is the probability we find $k = 3$ occurrences of GAATTC in l?

$$\text{Poisson law of parameter } \lambda = \beta\, l: \quad P_\lambda(n = k) = \frac{\lambda^k}{k!} e^{-\lambda}$$
$$P_\lambda : \mu = \lambda; \sigma^2 = \lambda$$

$$\lambda = \beta\, l = 10000/4096 \approx 2.4414 \text{ cuts in } 10 \text{ kb};$$
$$P_{2.4414}(n = 3) = 2.4414^3/3! \times e^{-2.4414} \approx 0.211$$

```
# Poisson probability of 3 with λ = 10000/4096
dpois(3, lambda = 10000/4096)

# Plot
k = 0:9
barplot(dpois(k, lambda = 10000/4096), col = "slateblue",
    main = "Poisson Law for EcoRI",
    xlab = "k", ylab = "P(n=k)"
)
```



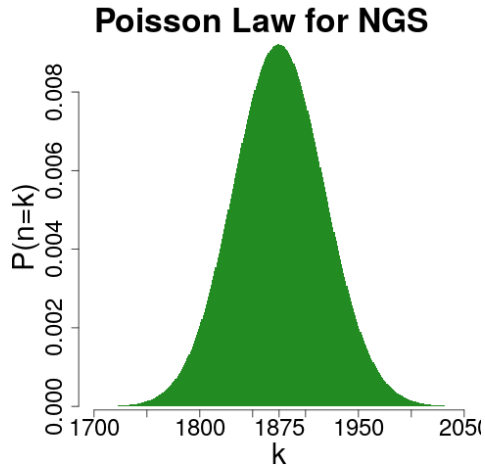
Poisson law for NGS

- 1 read can match on 0 or 1 expressed mRNA (kind of wishful thinking).
- The probability that this read matches mRNA m is $\beta = l_m n_m / \Sigma l_i n_i$ (Proportion of nucleotides in m among nucleotides in transcriptome).
- We observe a pool of $N_r = 10$ millions reads.

Assume

- 5000 expressed genes with average: length = 2kb, number = 800
  $\Rightarrow \Sigma l_i n_i = 5000 \times 2000 \times 800 = 8 \times 10^9$ nucleotides.
- messenger m is $l_m = 2.5$kb long, and is expressed $n_m = 600$times
  No expression level variability $\Rightarrow$ technical replicates
  $\Rightarrow \beta = 1.875 \times 10^{-4} \Rightarrow \lambda = \beta N_r = 1875$ read counts.

```r
# Plot
k = 1700:2050
barplot(dpois(k, lambda = 1875), col = "forestgreen",
    main = "Poisson Law for NGS",
    xlab = "k", ylab = "P(n=k)"
) # ... hmm, this looks like a normal law!

# Compare two technical replicates: same λ?
poisson.test(c(1948,1843)) # does not know λ=1875
```



Wilcoxon W test: principle

Consider two independent samples selected from populations...

with the same distribution	with different distributions
sample A: 54 90 56 22 73 26 17 25 50	sample A: 84 80 90 75 71 79 87 72 89
sample B: 49 79 85 40 20 67 12 72 10 28 53	sample B: 33 82 22 81 50 37 44 78 42 40 10
Merge and order: 10 12 17 20 22 25 26 28 40 49 50 53 54 56 67 72 73 79 85 90	Merge and order: 10 22 33 37 40 42 44 50 71 72 75 78 79 80 81 82 84 87 89 90
Forget values: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	Forget values: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Sum ranks: $W_A = \sum r_A - \sum_{i=1}^{n_A} i = 96 - 45 = 51$	Sum ranks: $W_A = \sum r_A - \sum_{i=1}^{n_A} i = 131 - 45 = 86$
$W_B = \sum r_B - \sum_{i=1}^{n_B} i = 114 - 66 = 48$	$W_B = \sum r_B - \sum_{i=1}^{n_B} i = 79 - 66 = 13$

Under H_0 = samples A and B are issued from the same law:

$$W \mapsto \text{Wilcoxon} - \text{Mann} - \text{Whitney law}\left(E = \frac{n_A n_B}{2}; V = \frac{n_A n_B (n_A + n_B + 1)}{12}\right)$$

$$E = 49.5; V = 173.25; \sqrt{V} = 13.16$$

$p = 0.94$	$p = 4.24 \times 10^{-3}$

Example: shaking oysters

Question:
Does a mechanical stress change Oyster's survival to microbial infection?

Method[6]:
Infect, then leave alone or shake 5 min on Day 3; count mortality on Day 9.

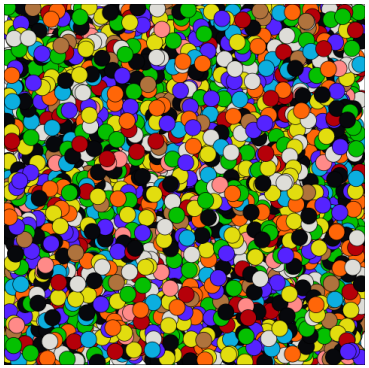
But: Only 4 replicates, no clue about how survival distibutes.

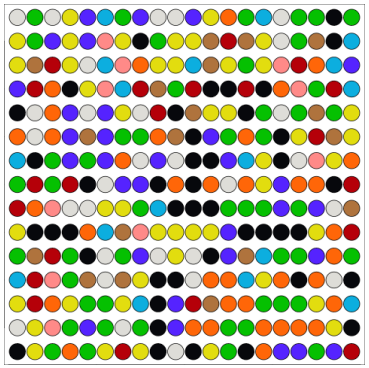
```
# Read data
od <- read.table("OysterDeath.tab",header=T)
attach(od)

#perform the test
wilcox.test(Mortality~Cond)

nA = length(Mortality[Cond=="ChalNoStress"])
nB = length(Mortality[Cond=="Chal5mStress"])
cat("E(W) =",nA*nB/2,"\n")
```

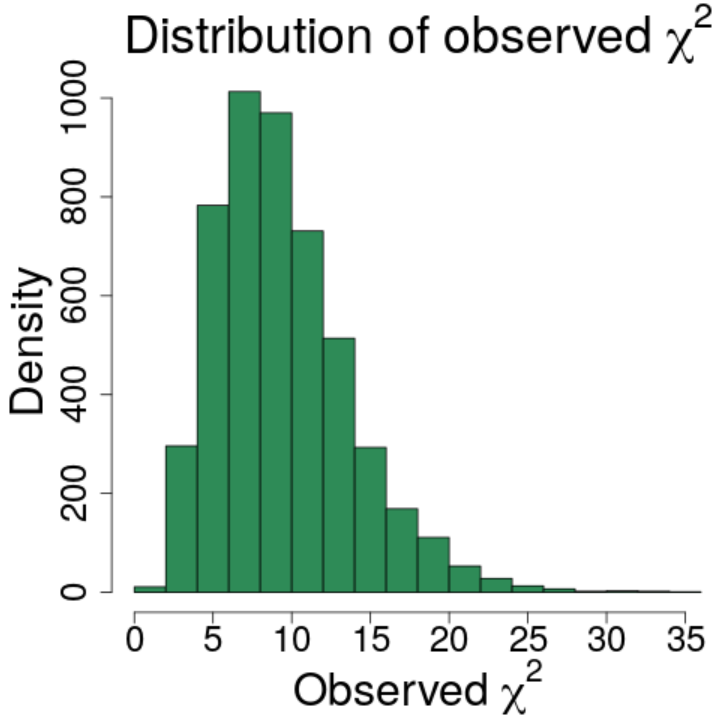
[6] Inspired by A. Lacoste, F. Jalabert, S.K. Malham, A. Cueff, S.A. Poulet (2001) *Appl. Envir. Microbiol.* **67:5** 2304–2309.

Testing numbers of individuals - Scheme

Population (∞) Sample N=300







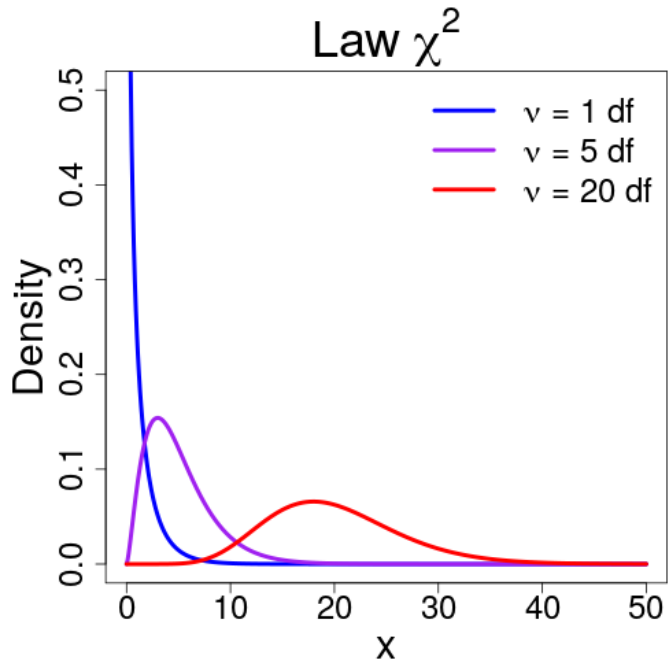
Class	Proba	Expected	Sample	χ^2_i
i	p_i	$e_i = Np_i$	s_i	$\frac{(s_i - e_i)^2}{e_i}$
	0.052	15.6	18	0.3692
	0.097	29.1	29	0.0003
	0.076	22.8	17	1.4754
	0.099	29.7	37	1.7943
	0.144	43.2	45	0.0750
	0.064	19.2	21	0.1688
	0.167	50.1	43	1.0062
	0.033	9.9	10	0.0010
	0.160	48.0	49	0.0208
	0.108	32.4	31	0.0605
$\sum_i$	***1.000***	***300***	***300***	***4.9716***

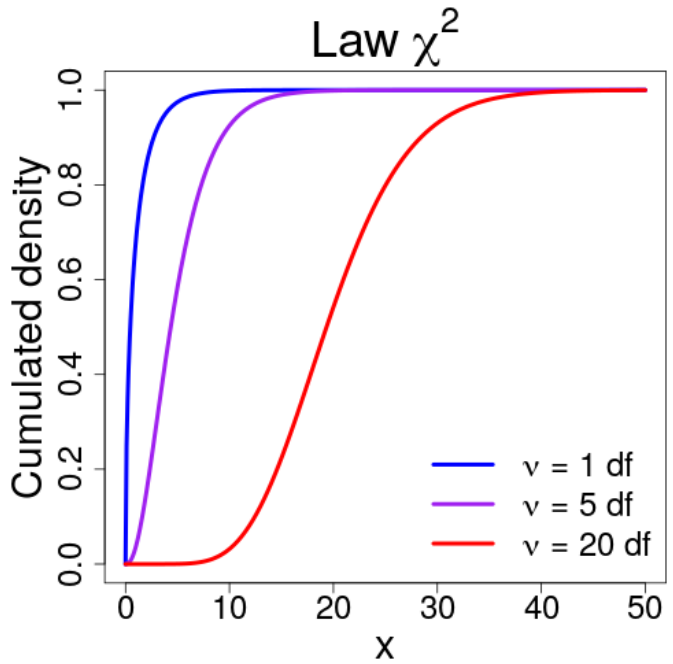
Testing numbers of individuals - Computations

The population P splits into b classes with proportions p_i, so that $\sum_{i=1}^{b} p_i = 1$

We take one sample S of size N out of this population
How does the distribution among classes differ from the expected one?

- Expected: $e_i = Np_i$
- b values $\rightarrow$ 1 indicative value: $\chi^2 = \sum_{i=1}^{b} \frac{(s_i - e_i)^2}{e_i}$





N.B. $\nu = b - 1$; the χ^2 law does NOT depend on N!

Example

In an area, 5 *protostomia* species are known to represent (in %):

Pseudonereis variegata	*Octomeris angulosa*	*Tetraclita serrata*	*Patella granularis*	*Perna perna*
56	25	9	7	3

On one beach in this area, we count:

Pseudonereis variegata	*Octomeris angulosa*	*Tetraclita serrata*	*Patella granularis*	*Perna perna*
298	149	61	32	29

Is this unexpected?

```
cat("\nχ² test: conformity to a known distribution\n")

pr <- c(0.56,0.25,0.09,0.07,0.03)    # sum = 1.00
spcnt <- read.csv("SpeciesCounting.csv",header=T)
attach(spcnt)

cat("H0: the sample comes from the population\n")
# Compute expected numbers
expnb = pr * sum(sample1)
# Compute χ² and p-value
cX2 = 0
for ( i in 1:length(pr) ) {
  cX2 = cX2 + (sample1[i]-expnb[i]) ** 2 / expnb[i]
}
pv = 1 - pchisq(cX2,df=length(pr)-1)  # We want P(χ² > cχ²)
cat(" χ² =",cX2,"\np-value =",pv,"\n")

# Less fun
chisq.test(sample1,p=pr)
```

χ^2 to test for independance

We want to compare three countries for the repartition of algae[7]

Data:

Country	Green	Red	Brown
France	121	403	183
Great Britain	109	347	183
Spain	94	365	155

But:
We have no proportions from a reference population
$\rightarrow$ How can we compute expected counts?

Hypothesis H_0: Algal group and Countries are independant factors.

$$\Rightarrow P(C_i \ \& \ G_j) = P(C_i) \times P(G_j) = \frac{nC_i}{N_{tot} \times \frac{nG_j}{N_{tot}}}; Exp_{ij} = N_{tot} \times P(C_i \ \& \ G_j)$$

Marginal sums and expected counts

C.	Green	Red	Brown	Sum
Fr	116.87	402.20	187.93	707
GB	105.63	363.51	169.86	639
Sp	101.50	349.29	163.21	614
Sum	324	1115	521	1960

$$N.B.\ \nu = (l-1)(c-1) = 4df$$

$$\chi^2 = \sum_{i=1}^{l} \sum_{j=1}^{c} \frac{(obs_{ij} - exp_{ij})^2}{exp_{ij}}$$

```
spfr <- c(121,403,183)
spgb <- c(109,347,183)
spes <- c( 94,365,155)
macroa <- matrix(c(spgb,spes,spfr),
                 byrow=T,nrow=3)
chisq.test(macroa) # matrix, no p=...
```

[7] Data from T. Burel, M. Le Duff, E. Ar Gall (2019) *Cah. Natur. Obs. Mar.* **VII:1**, 1–38.

χ^2 or not χ^2: GO-term enrichment

Among 30 genes, of which 8 belong to GO-term G1, we find 10 significantly up-regulated, of which 6 belong to G1

Is the UP pool enriched in GO-term G1?

Contingency table approach:

	UP	EQ	Σc
G1	6	2	8
Gx	4	18	22
Σl	10	20	30

```
gomat <- matrix(c(6,4,2,18),nrow=2)
rownames(gomat) <- c("G1","Gx")
colnames(gomat) <- c("UP","EQ")
print(gomat)

chisq.test(gomat)
```

We get the answer: reject H_0 with $p = 0.013$, but:

- There is a warning because of low values
- χ^2 says "H_0 can be rejected" but not "enriched / depleted"
  (We can look up for 1 GO-term, but extension to real-size analysis?)

Low-count-friendly approach: Fisher exact test

A very simple idea, indeed:

1. enumerate all possibilities: the number of G1 among 8 can be 0, 1, ..., 8
2. for each, compute the probability: how many ways to get this pattern?
   under H_0 = UP/EQ is independant from G1/Gx
   Example for $x = 6$:
   - Split 30 genes into 10 + 20 (all possible patterns):
     Combinations of 10 genes out of 30: $c_0 = \frac{30!}{10! \times 20!} = 30045015$
   - Split 8 genes into 6 + 2:
     Combinations of 6 genes out of 8: $c_1 = \frac{8!}{6! \times 2!} = 28$
   - Split 22 genes into 4 + 18:
     Combinations of 4 genes out of 22: $c_2 = \frac{22!}{4! \times 18!} = 7315$

   The probability for this pattern is:

$$p_6 = \frac{c_1 \times c_2}{c_0} = \frac{28 \times 7315}{30045015} \approx 6.817 \times 10^{-3}$$

3. sum-um all the proba of cases displaying this enrichment or more
   At least as much enriched: $p = \Sigma p_{x \geq 6}$

$$p = \Sigma p_{x \geq 6} = 7.23 \times 10^{-3}$$

Code for the GO-term enrichment test

```
gomat <- matrix(c(6,4,2,18),nrow=2)
rownames(gomat) <- c("G1","Gx")
colnames(gomat) <- c("UP","EQ")
print(gomat)

m = sum(gomat["G1",])
n = sum(gomat["Gx",])
k = sum(gomat[,"UP"])

c0 = choose((m+n),k) # (m+n)!/(n!m!)

x <- seq(0,m)
c1 = choose(m,x)     # m!/(x!(m-x)!)
c2 = choose(n,(k-x)) # n!/((k-x)!(n-(k-x))!)
p = c1 * c2 / c0
cat("Probabilities (hypergeometric law):\n")
print(p)
cat("Sum =",sum(p),"\n")
cat("P(x≥6) =",sum(p[x>=6]),"\n")
```

Lazy version:

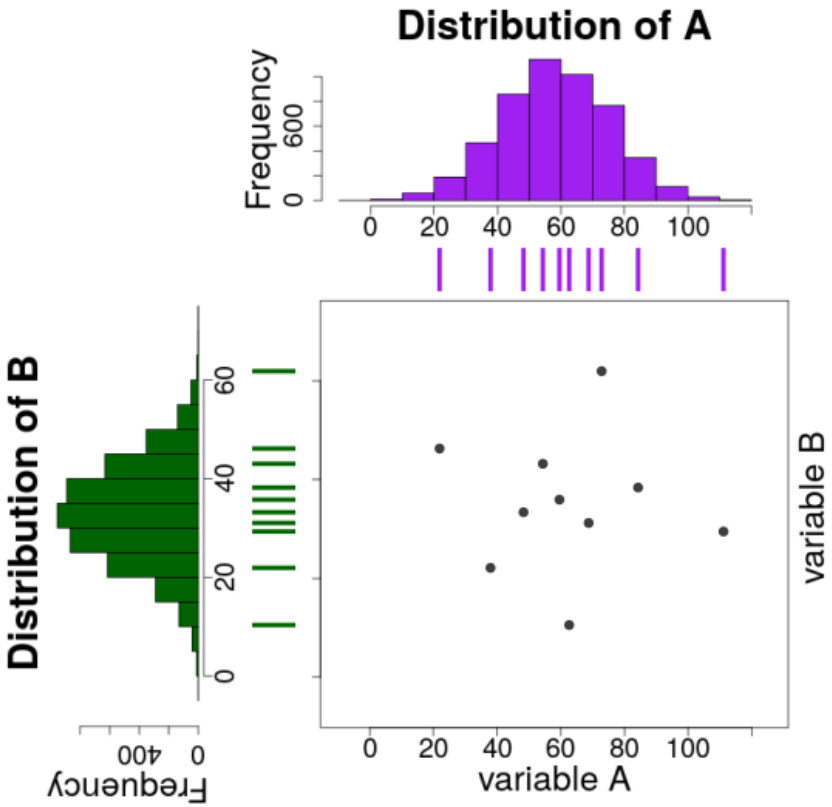
```
dh <- dhyper(x,m,n,k)
print(dh)
cat("Sum =",sum(dh),"\n")
cat("P(x≥6) =",sum(dh[x>=6]),"\n")
```

Super-lazy version:

```
fisher.test(gomat,alternative="greater")
```

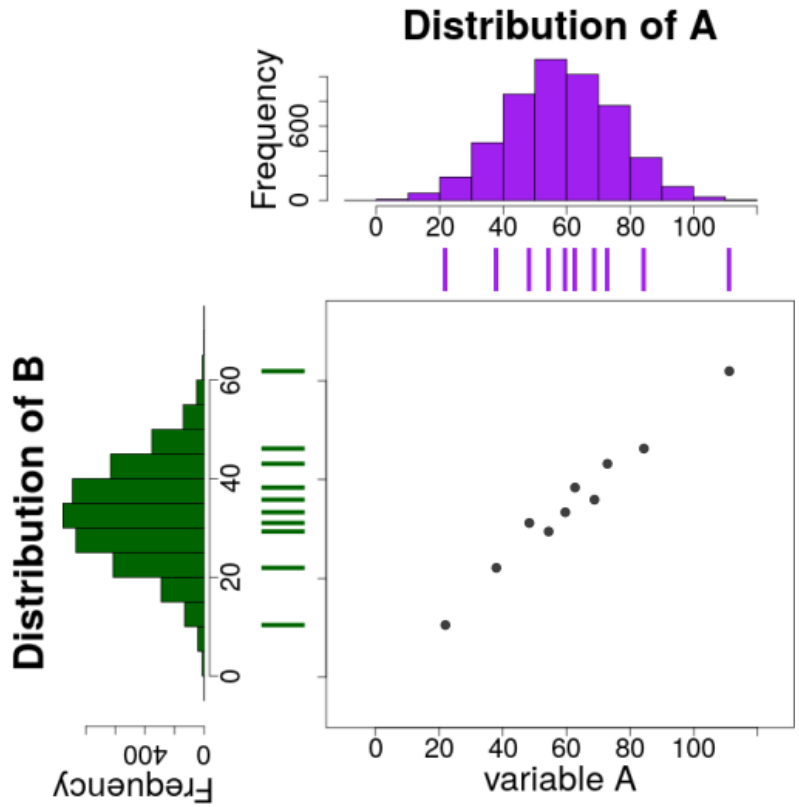
Two variables measured on the same individuals

A and B are independant



$a > \bar{a} \nleftrightarrow b > \bar{b} \Rightarrow s_{A,B} \approx 0$

A and B are correlated



Most often, $a > \bar{a} \leftrightarrow b > \bar{b} \Rightarrow s_{A,B} > 0$

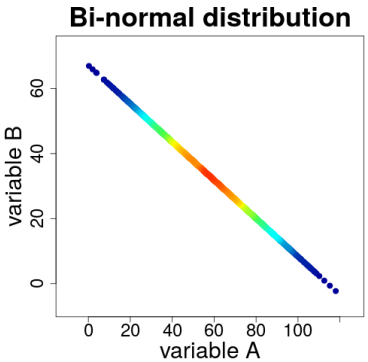
$$\text{Covariance}: \; s_{A,B} = \frac{1}{n-1} \sum_{i=1}^{N} (a - \bar{a})(b - \bar{b})$$

Properties of covariance and correlation coefficient

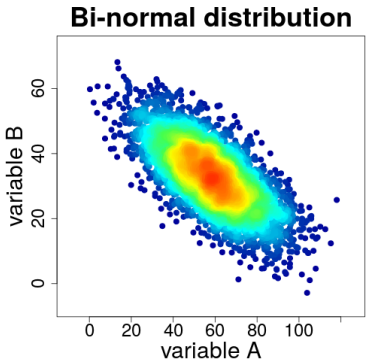
Covariance: $s_{A,B} = \frac{1}{n-1} \sum_{i=1}^{N} (a - \bar{a})(b - \bar{b})$

An index shared by all variables, whatever their s.d. (Pearson):

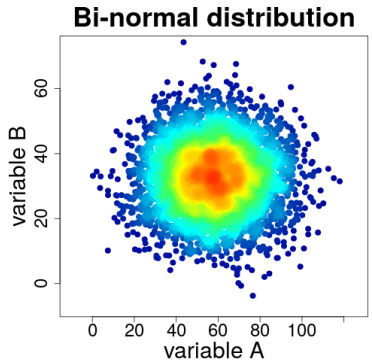
$$r_{A,B} = \frac{s_{A,B}}{s_A s_B}$$



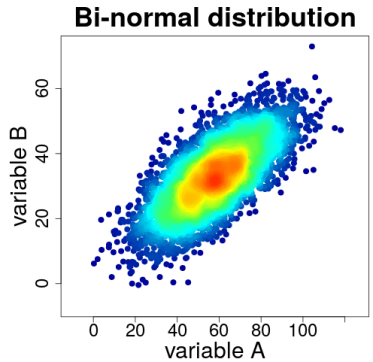
$s_{A,B} = -s_A s_B$
$r_{A,B} = -1$



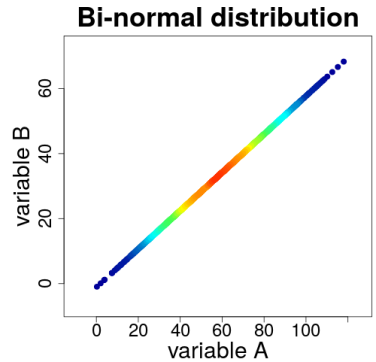
$-s_A s_B < s_{A,B} < 0$
$-1 < r_{A,B} < 0$



$s_{A,B} = 0$
$r_{A,B} = 0$



$0 < s_{A,B} < s_A s_B$
$0 < r_{A,B} < 1$



$s_{A,B} = s_A s_B$
$r_{A,B} = 1$

Properties of Pearson's correlation coefficient

$$s_{A,B} = \frac{1}{n-1}\sum_{i=1}^{N}(a-\bar{a})(b-\bar{b}) \qquad r_{A,B} = \frac{s_{A,B}}{s_A s_B}$$

Under H_0 = "Two uncorrelated variables":

$$r\sqrt{\frac{n-2}{1-r^2}} \mapsto t_{\nu=n-2}$$

if variables are normal or sample is large.

Example

Correlation between location of algae, and their content in Fucose + sulfate[8]

Species	Emersion(%)	Fucose (% D.W.)
Ascophyllum nodosum	9.5	16.1
Bifurcaria bifurcata	26.8	12.6
Fucus cerranoides	63.7	22.6
Fucus serratus	30.1	10.9
Fucus spiralis	81.6	21.4
Fucus vesiculosus	49.6	17.2
Laminaria digitata	12.8	3.6
Pelvetia canaliculata	98.5	34.9

```
# Data
fucdat <- read.table("Fucose.tab",header=T)
attach(fucdat)

# Parameters
sde = sd(Emersion) ; sdf = sd(Fucose)
co = cov(Emersion,Fucose)
r = co/(sde*sdf)
cat("Correlation r =",r,"\n")

# Test
free = nrow(fucdat)-2
t = r * sqrt(free/(1-r*r))

cat("p-value for positive correlation =",dt(t,df=free),"\n")
```

```
# less fun
r = cor(Emersion,Fucose)
cat("Correlation r =",r,"\n")
cor.test(Fucose,Emersion,alternative="g")

# Plot
plot(Emersion,Fucose,
   main="Fucose vs location",
   xlab="Emersion duration (%)",
   ylab="Fucose + Sulfate (%D.W.)" )
```

[8] Data from B. Kloareg (1991), *Bull. Soc. Bot. Fr.* **138:3-4**, 305-318.

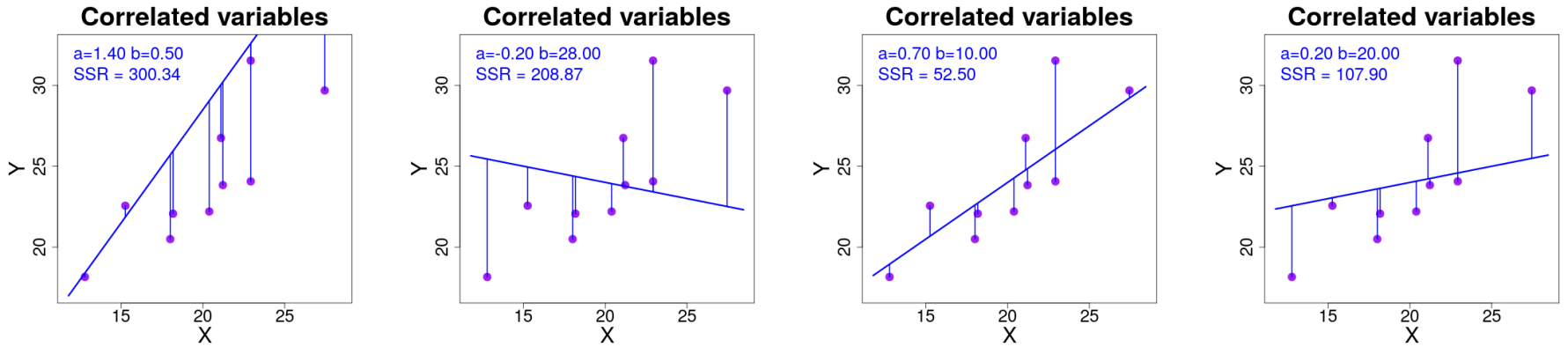
Linear regression

Assuming dependance between variables makes sense...

Linear correlation means $y = ax + b + \epsilon$ where ϵ = "residual"

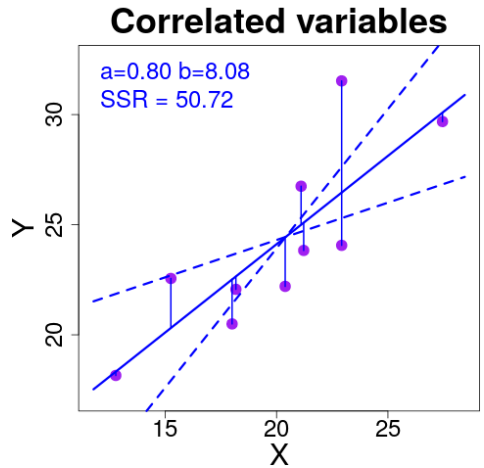
Estimates: $\hat{a}$ and $\hat{b}$ for which $\hat{y} = \hat{a}x + \hat{b}$ predicts y with the lowest error.

$\Rightarrow$ Define error criterion: $SSR = \sum_n \epsilon^2 = \sum_n (y - \hat{y})^2$



Good news: no need to try all possible values!

$$\hat{a} = r\frac{s_y}{s_x} \quad ; \quad \hat{b} = \bar{y} - \hat{a}\bar{x}$$



Confidence intervals:

$$s_a = s_{SSR}/\sqrt{\sum (x_i - \bar{x})^2} \qquad \Delta\hat{a} \mapsto s_a t_{\nu=n-2}$$

$$s_b = s_{SSR}\sqrt{\sum x_i^2 / n \sum (x_i - \bar{x})^2} \qquad \Delta\hat{b} \mapsto s_b t_{\nu=n-2}$$

Example

How do tides impact phytoplankton biomass in Penze?[9]

→ Correlate chorophyl content with water level.

```
# Read data
chloro <- read.table ("ChloroWater.tab",
                                    header=T)
attach(chloro)

# Check linear correlation
rc = cov(H,Chl)/(sd(H)*sd(Chl))

# Estimates for a and b
ac = rc * sd(Chl) / sd(H)
bc = mean(Chl) - ac * mean(H)
cat("Chl = ",ac," H +",bc,"\n")

# Use the linear model function
datac <- data.frame(chloro)
regc <- lm(Chl~H,datac)

cat("Regression coefficients:\n")
print(regc$coef)
cat("Confidence interval:\n")
confint(regc,level=0.95)
```

```
# Plot
minh=1.5
maxh=5.0
minc=1.0
maxc=3.0
plot(Chl~H, pch=20, col="darkgreen",cex=3,
   main="Diurnal variations",
   xlim=c(minh,maxh),ylim=c(minc,maxc),
   xlab="Water H (m)",ylab="Chlorophyl (µg/l)")
lines(c(minh,maxh),c(ac*minh+bc,ac*maxh+bc),
                                col="blue",lwd=3)
text(minh,minc+0.15,pos=4,
 sprintf("a=%.3f b=%.3f\nSSR = %.4f",ac,bc,SSRc))
```

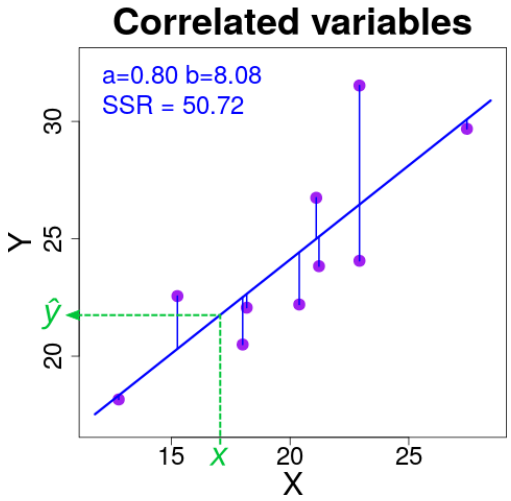
```
cat("  Regression residuals:\n")
print(regc$residuals)
SSRc = sum(regc$residuals**2)
cat("  SSR =",SSRc,"\n")
```

[9] Data from C. Riaux and J.L. Douvillé (1980), *Estuar. Coast. Mar. Sci* **10**, 85-92.

Correlation and prediction

Regression = modeling the relation between x and y

$\Rightarrow$ knowing x, predict y as $\hat{y} = \hat{a}x + \hat{b}$

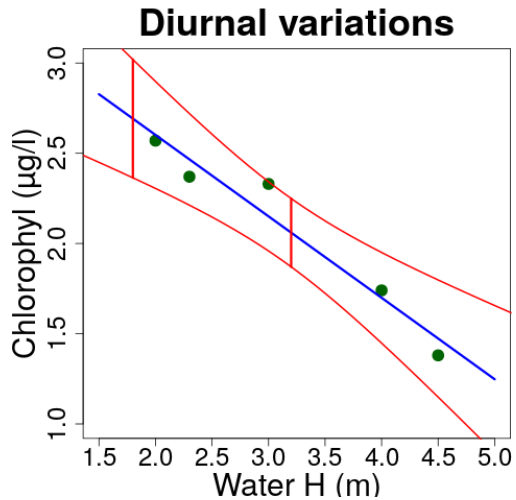


BUT uncertainty $\Delta\hat{a}$ and $\Delta\hat{b} \Rightarrow \Delta\hat{y}$

$$\Delta\hat{y} \quad \mapsto \quad \left(s_{SSR} \sqrt{\frac{1}{n} + \frac{(x-\bar{x})^2}{\sum (x_i - \bar{x})^2}} \right) t_{\nu=n-2}$$

N.B. the more x differs from $\bar{x}$, the highest $\Delta\hat{y}$

```
# Prediction
alpha = 0.05
askx <- c(1.8,3.2)
ansy <- predict(regc,data.frame(H=askx),
            level=1-alpha,interval="confidence")

cat("H =",askx[1],"-> predicted Chl =",ansy[1,"fit"],
"95% interval: ",ansy[1,"lwr"],"< Chl <",ansy[1,"upr"],"\n")
cat("H =",askx[2],"-> predicted Chl =",ansy[2,"fit"],
"95% interval: ",ansy[2,"lwr"],"< Chl <",ansy[2,"upr"],"\n")
```

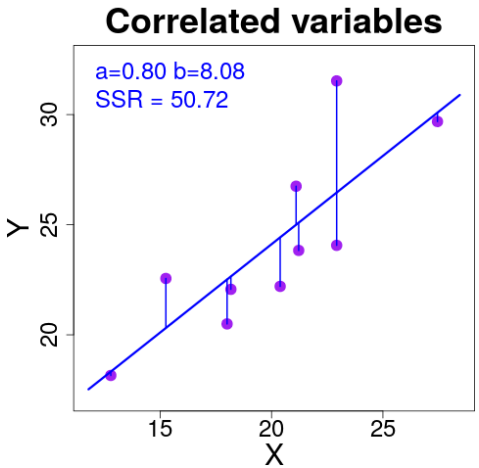


Predict Y *vs* predict X

Regression = modeling the relation between x and y as $y = ax + b$

$\Rightarrow$ knowing x, predict y for x within a reasonable range.

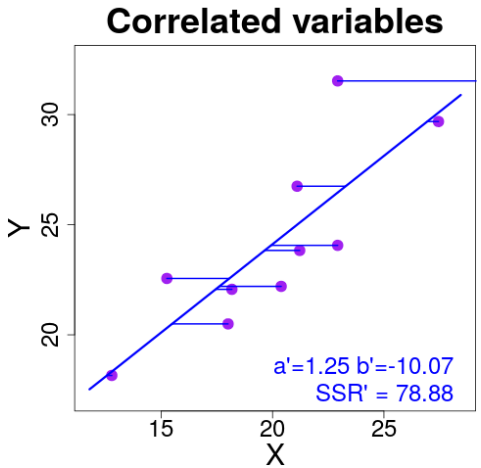
$\Rightarrow$ knowing y, predict x?

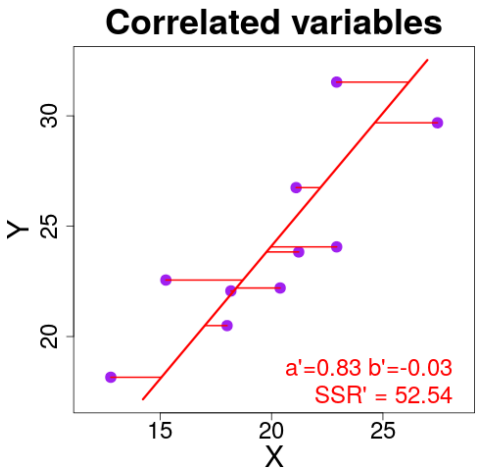


$$y = ax + b \Leftrightarrow x = \frac{1}{a}y - \frac{b}{a}$$

or :

$$x = a'y + b' \qquad a' = \frac{1}{a} \; ; \; b' = -\frac{b}{a}$$



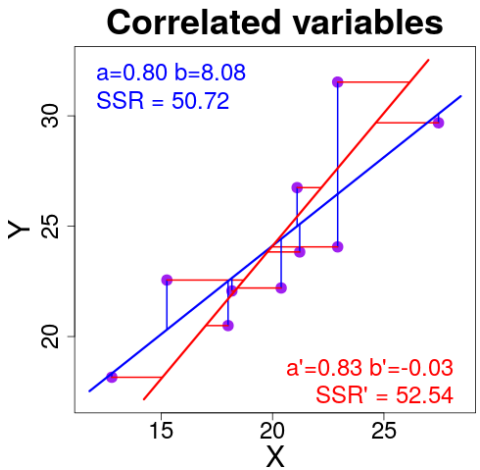


But, as X and Y are symmetric:

$$\hat{a} = r\frac{s_y}{s_x} \Leftrightarrow \hat{a}' = r\frac{s_x}{s_y}$$

So:

$$\hat{a}' \neq 1/\hat{a} \quad \text{... unless } r = 1!$$



More than one factor

i	A	B	C	X
1	A_1	B_1	C_1	X_1
2	A_2	B_2	C_2	X_2
3	A_3	B_3	C_3	X_3
...				

Question:
does the value of X depend on...

- factor A?
- factor B?
- factor C?
- an interaction between A and B?
- an interaction between A and C?
- ...?

Model: for each individual i,

$$X_i = aA_i + bB_i + cC_i + X_0 + \epsilon_i$$

Problem 1:
knowing A_i, B_i, C_i for all i,
estimate a, b, c, X_0 so that
the X_i are well predicted by:

$$\hat{X}_i = \hat{a}A_i + \hat{b}B_i + \hat{c}C_i + \hat{X}_0$$

$$\Leftrightarrow$$

$$\sum \epsilon_i^2 \text{ is minimal}$$

Problem 2:
Which factor(s)
display(s) a significant influence?

Simple answers

Answer to problem 1 = find coefficients:

- Simply use linear regression: `lm` can cope with multiple factors

```
multimodel <- lm( X ~ A + B + C )
```

Answer to problem 2 = effect of each factor:

- Simply use AnOVa: `anova` can cope with multiple factors

```
anova(multimodel)
```

Bonus = combined effect of factors:

- Simply replace + by *

```
combomodel <- lm( X ~ A * B * C )
anova(combomodel)
```

A real, thus less simple, situation

Factors having an effect on brown alga fecundity[10].

Question: how does fecundity depend on the three following factors:

- species considered = *Laminaria ochroleuca*, *Sacchoriza polyschides* ;
- water temperature = 10°C, 15°C, 25°C ;
- time = 0, 2, ... , 26 days

→ How about something like:

```
AlFec <- lm(Fec ~ Spe * Tmp * Day)
anova(AlFec)
```

Problem: Temperature and Time are quantitative,
Species are qualitatitive (categories)

Solution: Turn all variables to qualitative, *i.e.* factors!

```
fectbl <- read.table("SpeDayTmpFec.tsv",header=T)
fectbl[,1:3] <- lapply(fectbl[,1:3],as.factor)
model <- aov(Fec ~ Spe * Day * Tmp, data = fectbl)
summary(model)
```

Bonus: because `model` is made by `aov` – not `lm` – *post-hoc* test applies:

```
TukeyHSD(model,c("Spe","Day","Tmp"))
```

[10] data very approximately reconstructed from T.R. Pereira, A.H. Engelen, G.A. Pearson, E.A. Serrão, C. Destombe, M. Valero (2011) *Cah. Biol. Mar.* **52**:395-403.

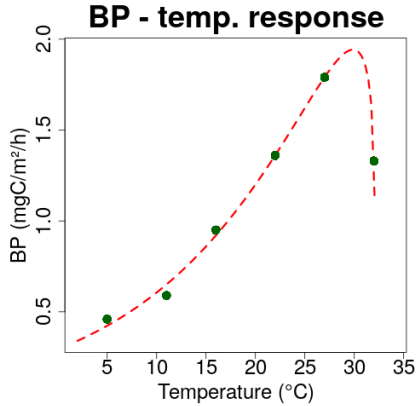
Non-linear parameter optimization

Example: Temperature and Bacterial Production[11]
Bacterial production follows a non-linear law:

$$BP(T) = BP_{max}\left(\frac{T_{max}-T}{T_{max}-T_{opt}}\right)^{\beta}\exp\left(-\beta\left(\frac{T_{max}-T}{T_{max}-T_{opt}}-1\right)\right)$$

with BP_{max}, T_{max}, T_{opt} and β : parameters to estimate.

```
# create the function
metatemp <- function(temp,BPmax,Tmax,Topt,beta) {
  BPmax*((Tmax-temp)/(Tmax-Topt))**beta*
   exp(-beta*(((Tmax-temp)/(Tmax-Topt))-1))
}
# read data and build the data frame
tbptbl <- read.table("TempBPresp.tab",header=T)
attach(tbptbl)
databp <- data.frame(tbptbl)
# optimize parameters
opttbp <- nls(
    BP~metatemp(Temp,BPmax,Tmax,Topt,beta),
    data=databp,
    start=list(BPmax=max(BP),Tmax=max(Temp),
               Topt=27,beta=1) )
opc <- coef(opttbp)
print(opc)
```




```
# plot
tpx<-seq(2,34,length.out=100)
bpy<-metatemp(tpx,opc[1],opc[2],opc[3],opc[4])
plot(tpx,bpy,type="l",lty=2,lwd=3,col="red",
  main="BP - temp. response",
  xlab="Temperature (°C)",ylab="BP (mgC/m2/h)")
points(databp,pch=16,col="darkgreen",cex=2)
```

[11] Data from C. Hubas, L.F. Artigas, D. Davoult (2007). *Mar Ecol Prog Ser* **344**:39-48.

Workflow of statistic test