Formulario

Fórmulas teoría/*Formula en R*

- Mean: Average value

- Integral simple

```
integrate(function(x){(x^2)/3},-1,2)
```

snippet binom
dbinom(${3},${2},${1}) # X~B(${3:numero de exitos},${2:size},${1:probabilidad}) P(X=1)
pbinom(${3},${2},${1}) # X~B(${3:numero de exitos},${2:size},${1:probabilidad})P(X<=1)
1-pbinom(${3},${2},${1}) # X~B(${3:numero de exitos}-1,${2:size},${1:probabilidad})P(X>=2) Pon en nºexitos uno menos que los que al menos necesitas (4⇒3)
pbinom(${5:Y},${2},${1}) - pbinom(${4:X}-1,${2},${1}) # P(x<=X<=y)

snippet poiss
dpois(${3:numero de exitos},${2:constant}
${1:intervalo de tiempo}) #A exact number of exitos
ppois(${4:numero de exitos},${5:constant}
${6:intervalo de tiempo}) # P(X <= x)
dpois(${9:numero de exitos},${8:nºde trials}
${7:probabilidad de exito}) #Desde la binomial
1-dpois(${12:numero de exitos},${11:nºde trials}
${10:probabilidad de exito}) #Desde la binomial y al menos un numero de exito
qpois(${13:percentil},${5:constant}*${6:intervalo de tiempo}) # Percentiles
snippet unicon

```
punif(${3:valueuntil},${1:A_Interval},${2:B_Interval}) #P(X <
${3:valueuntil} )
```

```
1-punif(${3:valueuntil},${1:A_Interval},${2:B_Interval}) #P(X
>=${3:valueuntil} )
```

snippet inteProb
eq = "${1:funcion}+c"
(yac_integral = paste0("Integrate(x,${2:int1}, ${3:int2})-eq"))
Ryacas::yacas_str(yac_integral)
eq_c = #poner lo de antes ya integrada
(yac_eq_c = paste0("Solve(", eq_c, ")=1,c"))
Ryacas::yacas_str(yac_eq_c)
snippet normZ
(${1:X}-${2:mean})/${3:sd} #Z calculus
dnorm((X-mean)/sd) # P(Z=x)
pnorm((${1:X}-${2:mean})/${3:sd}) # P(Z<=x)
1-pnorm((${1:X}-${2:mean})/${3:sd}) # P(Z>x)
pnorm(${5:Y},${2:mean},${3:sd}) - pnorm(${4:X},${2:mean},${3:sd}) # P(x<Z<y)
qnorm(${6:percentil},${2:mean},${3:sd}) # P(X<%percentil%)
qnorm(${7:area_izquierda},${2:mean},${3:sd})
qnorm(1 - ${8:area_derecha},${2:mean},${3:sd})

snippet integralIndef
Define la variable y la función
t ← Sym("${1:variable}")
funcion ← ${2:funcion}

```
# Calcula la integral indefinida
integral_indefinida <- Integrate(funcion, ${1:variable})
print(integral_indefinida)
```

snippet normToBin
ssdd ← sqrt(${1:n}
*${2:p}*(1-${2:p})) #sd
medmed ← ${1:n}*${2:p}

snippet ponderacion
mean = sum(${1:DataVector}*${2:probabilitiesVector})
sqrt(sum(${2:probabilitiesVector} * (${1:DataVector} - mean)^2))

snippet contProb
f ← function(x){${1:funcion}}
fx ← function(x){x*(${1:funcion})}
mean ← integrate(fx, ${2:int1}, ${3:int2})\$value #mean
fxx ← function(x){(x^2)*(${1:funcion})}
EX2 ←integrate(fxx, ${2:int1}, ${3:int2})\$value
var ← EX2 - (mean)^2# varianza
(sqrt(var)) #sd

snippet sterror

```
${1:sd}/sqrt(${2:sizeSample})
```

snippet confInterval
#principio del intervalo o upper bound de la media, en ese caso deja solo el primer porcentaje, el intervalo es desde -INF hasta este numero
${2:mean}-(qnorm(${3:percent}+(1-${3:Percent})/2))
(${4:sd}/sqrt(${1:sizeSample}))
 # el porcentaje se pone en 0.x
#final del intervalo o lower bound de la media, quita el segundo porcentaje, el intervalo es desde este numero a +INF
${2:mean}+(qnorm(${3:Percent}+(1-${3:Percent})/2))
(${4:sd}/sqrt(${1:sizeSample}))
#si nos piden how large is the sample tenemos que despejar la n de la inecuación

```
#For a proportion
stError = (qnorm(${7:percent}+(1-${7:Percent})/2))*sqrt((${6:proportion}*(1-${6:proportion}))/${5:sizeSample})
#ya con el stError se lo sumas y restas a la proporcion para dar el IC

#FOR DIFFERENCE IN POPULATION PROPORTIONS

(${8:p1}-${9:p2})+(qnorm(${7:percent}+(1-${7:Percent})/2))*sqrt((${8:p1}*(1-${8:p1})/${11:n1})+(${9:p2}*(1-${9:p2}))/${12:n2})
```

```
(${8:p1}-${9:p2})-(qnorm(${7:percent}+(1-${7:Percent})/2))*sq
rt((${8:p1}*(1-${8:p1})/${11:n1})+(${9:p2}*(1-${9:p2}))/${12:
n2})

#FOR DIFFERENCE IN POPULATION MEANS

(${13:mean1}-${14:mean2})+(qnorm(${15:percent}+(1-${15:Percen
t})/2))*sqrt(((${16:sd1})^2/${18:n1})+((${17:sd2})^2/${19:n
2}))
(${13:mean1}-${14:mean2})-(qnorm(${15:percent}+(1-${15:Percen
t})/2))*sqrt(((${16:sd1})^2/${18:n1})+((${17:sd2})^2/${19:n
2}))

#si el valor 0 esta dentro del intervalo NO hay diferencia

#CONFIDENCE INTERVALS FOR DIFFERENCE IN POPULATION MEANS (UNK
NOWN VARIANCES)
#SAME VARIANCES DIAP 56 TEMA 6
#DIFFERENT VARIANCES
s <- c(${1:sd1},${2:sd2})
n <- c(${3:n1},${4:n2})
m <- c(${5:mean1},${6:mean2})
num <- ((s[1]/n[1])+(s[2]/n[2]))^2
d1 <- ((s[1]/n[1])^2/(n[1]-1))
d2 <- ((s[2]/n[2])^2/(n[2]-1))
v <- num/(d1+d2)
print(paste("V=",v))
ci <- .95
t <- qt(1-(1-ci)/2,v)
print(paste("t=",t))
ci_low <- (m[1]-m[2])-t*sqrt(
(s[1]/n[1])+(s[2]/n[2]))
ci_up <- (m[1]-m[2])+t*sqrt(
(s[1]/n[1])+(s[2]/n[2]))
print(paste("(",ci_low,"-",ci_up,")"))
```

```
snippet tDist
#IF n<30 & UNKNOWN sd
qt(${1:Percent}+(1-${1:Percent})/2),${2:sizeMuestra}-1) #se le resta 1 porque
#son los grados de libertad
#IF n<30
```

```
snippet SampleSize
((qnorm(${3:percent}+(1-${3:Percent})/2)
${1:sd})/${2:error})^2 #sin aproximar
ceiling(((qnorm(${3:percent}+(1-${3:Percent})/2)
${1:sd})/${2:error})^2) #aproximado
PROPORTION
If we are unsure about the error we can accept we
should maximize the error. That happens with ^p=0.5
(qnorm(${6:percent}+(1-${6:Percent})/2))^2*(${5:proportion}*
(1-${5:proportion}))/(${4:error})^2
```

```
snippet PairedSamples
diff ← c(0.16,0.38,0.17,0.31,0.19,0.35,0.43, #aquí pon los valores de los que quieras
trabajar
-.21,0.34,0.2)
diff_m ← mean(diff)
s← sd(diff)
n ← length(diff)
t ← qt(1-(1-${2:%})/2,n-1)
ci_low ← diff_m - (t*(s/sqrt(n)))
ci_high ← diff_m + (t*(s/sqrt(n)))
print(paste("(",ci_low,"-",ci_high,")")
```

```
snippet ChiCI
ci = ${1:ci}
sample ← c(46.4, 46.1, 45.8, 47.0, 46.1,
45.9, 45.8, 46.9, 45.2, 46.0)#aqui mete los datos pa trabjar
sample_m ← mean(sample)
s← sd(sample)
n ← length(sample)
chi_low ← qchisq(1-((1-ci)/2),(n-1))
chi_high ← qchisq((1-ci)/2,n-1)
```

ci_low ← ((n-1)*s^2)/chi_low
ci_high ← ((n-1)*s^2)/chi_high
print(paste("(",ci_low,"-",ci_high,")"))

snippet SampleVariance
n = ${1:n}
sumx = ${2:sumx}
sumx2= ${3:sumx2}
mean = sumx/n
sva = (sumx2/n - mean^2)*(n/(n-1))
ssd = sqrt(sva)

snippet fDist
n = ${1:n1}
m = ${2:n2}
F_high← qf((${3:percent}+(1-${3:Percent})/2),n-1 ,m-1)
F_low ←qf((1-${3:Percent})/2,n-1 ,m-1)
print(paste("(",F_low,"-",F_high,")"))

```
s1 =  # poner aqui la s1
s2 =  # poner aqui la s2
s12 = s1/s2
ci_high <- s12/F_high
ci_low <- s12/F_low
print(paste("(",ci_low,"-",ci_high,")"))
```