Week 1

Tomás Fuenzalida

29-02-2020

## Week 2

Las diversas distribuciones tienen 4 comandos importantes.

### dnorm (Funcion de densidad)

Nos entrega la funcion de densidad.

set.seed(1)  
dnorm(0 , mean = 0, sd = 1) #Probabilidad del quantil 0 con media 0 y sd 1

## [1] 0.3989423

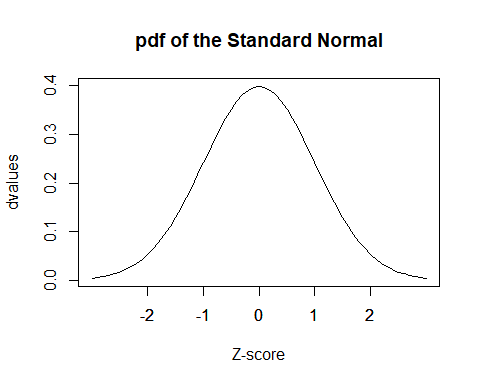
z\_scores <- seq(-3,3,.1); z\_scores

## [1] -3.0 -2.9 -2.8 -2.7 -2.6 -2.5 -2.4 -2.3 -2.2 -2.1 -2.0 -1.9 -1.8 -1.7 -1.6  
## [16] -1.5 -1.4 -1.3 -1.2 -1.1 -1.0 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1  
## [31] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4  
## [46] 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9  
## [61] 3.0

dvalues <- dnorm(z\_scores) ; dvalues

## [1] 0.004431848 0.005952532 0.007915452 0.010420935 0.013582969 0.017528300  
## [7] 0.022394530 0.028327038 0.035474593 0.043983596 0.053990967 0.065615815  
## [13] 0.078950158 0.094049077 0.110920835 0.129517596 0.149727466 0.171368592  
## [19] 0.194186055 0.217852177 0.241970725 0.266085250 0.289691553 0.312253933  
## [25] 0.333224603 0.352065327 0.368270140 0.381387815 0.391042694 0.396952547  
## [31] 0.398942280 0.396952547 0.391042694 0.381387815 0.368270140 0.352065327  
## [37] 0.333224603 0.312253933 0.289691553 0.266085250 0.241970725 0.217852177  
## [43] 0.194186055 0.171368592 0.149727466 0.129517596 0.110920835 0.094049077  
## [49] 0.078950158 0.065615815 0.053990967 0.043983596 0.035474593 0.028327038  
## [55] 0.022394530 0.017528300 0.013582969 0.010420935 0.007915452 0.005952532  
## [61] 0.004431848

#http://seankross.com/notes/dpqr/  
  
plot(dvalues, # Plot where y = values and x = index of the value in the vector  
 xaxt = "n", # Don't label the x-axis  
 type = "l", # Make it a line plot  
 main = "pdf of the Standard Normal",  
 xlab= "Z-score")   
  
# These commands label the x-axis  
axis(1, at=which(dvalues == dnorm(0)), labels=c(0))  
axis(1, at=which(dvalues == dnorm(1)), labels=c(-1, 1))  
axis(1, at=which(dvalues == dnorm(2)), labels=c(-2, 2))  
axis(1, at=which(dvalues == dnorm(0)), labels=c(0))  
axis(1, at=which(dvalues == dnorm(1)), labels=c(-1, 1))  
axis(1, at=which(dvalues == dnorm(2)), labels=c(-2, 2))



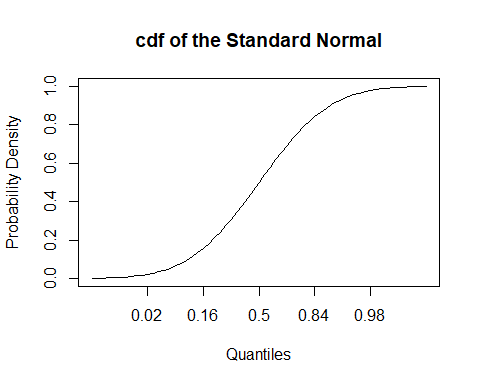
### pnorm(Funcion de densidad acumulada)

Esta funcion entrega la funcion de densidad acumulada

pnorm(2) #Probabilidad que el valor sea menor a 2

## [1] 0.9772499

pvalues <- pnorm(z\_scores)  
  
# Now we'll plot these values  
plot(pvalues, # Plot where y = values and x = index of the value in the vector  
 xaxt = "n", # Don't label the x-axis  
 type = "l", # Make it a line plot  
 main = "cdf of the Standard Normal",  
 xlab= "Quantiles",  
 ylab="Probability Density")   
  
# These commands label the x-axis  
axis(1, at=which(pvalues == pnorm(-2)), labels=round(pnorm(-2), 2))  
axis(1, at=which(pvalues == pnorm(-1)), labels=round(pnorm(-1), 2))  
axis(1, at=which(pvalues == pnorm(0)), labels=c(.5))  
axis(1, at=which(pvalues == pnorm(1)), labels=round(pnorm(1), 2))  
axis(1, at=which(pvalues == pnorm(2)), labels=round(pnorm(2), 2))



### qnorm (Nos entrega el valor del quantil de los quantiles)

Esta funcion nos entrega el valor del quantil y no la probabilidad es la funcion inversa de pnorm

qnorm(.96)

## [1] 1.750686

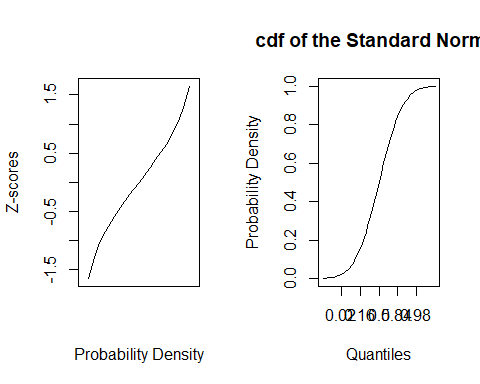
# This is for getting two graphs next to each other  
oldpar <- par()  
par(mfrow=c(1,2))  
  
# Let's make a vector of quantiles: from 0 to 1 by increments of .05  
quantiles <- seq(0, 1, by = .05); quantiles

## [1] 0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70  
## [16] 0.75 0.80 0.85 0.90 0.95 1.00

# Now we'll find the Z-score at each quantile  
qvalues <- qnorm(quantiles); qvalues

## [1] -Inf -1.6448536 -1.2815516 -1.0364334 -0.8416212 -0.6744898  
## [7] -0.5244005 -0.3853205 -0.2533471 -0.1256613 0.0000000 0.1256613  
## [13] 0.2533471 0.3853205 0.5244005 0.6744898 0.8416212 1.0364334  
## [19] 1.2815516 1.6448536 Inf

# Plot the z\_scores  
plot(qvalues,  
 type = "l", # We want a line graph  
 xaxt = "n", # No x-axis  
 xlab="Probability Density",  
 ylab="Z-scores")  
  
# Same pnorm plot from before  
plot(pvalues, # Plot where y = values and x = index of the value in the vector  
 xaxt = "n", # Don't label the x-axis  
 type = "l", # Make it a line plot  
 main = "cdf of the Standard Normal",  
 xlab= "Quantiles",  
 ylab="Probability Density")   
  
# These commands label the x-axis  
axis(1, at=which(pvalues == pnorm(-2)), labels=round(pnorm(-2), 2))  
axis(1, at=which(pvalues == pnorm(-1)), labels=round(pnorm(-1), 2))  
axis(1, at=which(pvalues == pnorm(0)), labels=c(.5))  
axis(1, at=which(pvalues == pnorm(1)), labels=round(pnorm(1), 2))  
axis(1, at=which(pvalues == pnorm(2)), labels=round(pnorm(2), 2))



### rnorm (Vector de numeros random)

Esta funcion nos entrega un vector de numeros que siguen la distribucion señalada

rnorm(5, mean = 0, sd = 1)

## [1] -0.6264538 0.1836433 -0.8356286 1.5952808 0.3295078

## Week 2

### Data Sample

Si quiero el percentil de una muestra de 100 personas de una distribucion normal, debo dividir la desviacion en la raiz de la muestra (sqrt(100))

qnorm(0.95, mean= 1100, sd = 75/sqrt(100))

## [1] 1112.336

## Week 3

### T confidence intervals example

Para comparar la efectividad de un remedio se puede hacer un test-t para saber si en promedio el remedio hace efecto, con t.test. El test-t se utiliza para comparar si las medias de dos grupos son iguales entre si. (Hipotesis nula medias son iguales)

data(sleep)  
head(sleep)

## extra group ID  
## 1 0.7 1 1  
## 2 -1.6 1 2  
## 3 -0.2 1 3  
## 4 -1.2 1 4  
## 5 -0.1 1 5  
## 6 3.4 1 6

g1 <- sleep$extra[1:10]  
g2 <- sleep$extra[11:20]  
difference <- g2 - g1  
mn <- mean(difference)  
s <- sd(difference)  
n <- 10  
mn + c(-1, 1) \* qt(0.975, n - 1) \* s/sqrt(n)

## [1] 0.7001142 2.4598858

t.test(difference) #Como el intervalo no contiene al 0, se puede afirmar la hipotesis

##   
## One Sample t-test  
##   
## data: difference  
## t = 4.0621, df = 9, p-value = 0.002833  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 0.7001142 2.4598858  
## sample estimates:  
## mean of x   
## 1.58

### Independent group T intervals

Lo mismo anterior pero para dos grupos independientes (Placebo y real)

data(sleep)  
head(sleep)

## extra group ID  
## 1 0.7 1 1  
## 2 -1.6 1 2  
## 3 -0.2 1 3  
## 4 -1.2 1 4  
## 5 -0.1 1 5  
## 6 3.4 1 6

g1 <- sleep$extra[1:10]  
g2 <- sleep$extra[11:20]  
t.test(g2,g1, paired = FALSE, var.equal = TRUE) #Como el intervalo contiene al 0, se afirma la hipotesis nula

##   
## Two Sample t-test  
##   
## data: g2 and g1  
## t = 1.8608, df = 18, p-value = 0.07919  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.203874 3.363874  
## sample estimates:  
## mean of x mean of y   
## 2.33 0.75

library(reshape2)  
library(dplyr)

data("chickwts")  
wideCw <- dcast(ChickWeight, Diet + Chick ~ Time, value.var = "weight")  
names(wideCw)[-(1:2)] <- paste("time", names(wideCw)[-(1:2)], sep = "")  
wideCw <- mutate(wideCw, gain = time21 - time0)  
wide14 <- subset(wideCw, Diet %in% c(1,4))  
t.test(gain ~ Diet, paired = FALSE, var.equal = TRUE, data = wide14)$conf #Quiero explicar "gain" con la variable Diet,

## [1] -108.14679 -14.81154  
## attr(,"conf.level")  
## [1] 0.95

#Para esto debe tener solamente dos niveles ya que es una comparacion entre dos cosas  
t.test(gain ~ Diet, paired = FALSE, var.equal = FALSE, data = wide14)$conf #Aqui se asume que la varianza no es igual

## [1] -104.65901 -18.29932  
## attr(,"conf.level")  
## [1] 0.95

#Ya que el intervalo no contiene al 0, podemos decir que se gana menos peso en la dieta 1 que en la 4

### T tests

library(UsingR)

data(father.son)  
t.test(father.son$sheight, father.son$fheight, paired = TRUE) #Un padre, un hijo (PAIRED = TRUE)

##   
## Paired t-test  
##   
## data: father.son$sheight and father.son$fheight  
## t = 11.789, df = 1077, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.8310296 1.1629160  
## sample estimates:  
## mean of the differences   
## 0.9969728

#Como el intervalo no contiene al 0, se rechaza la hipotesis nula

### Two group testing

library(reshape2)  
library(dplyr)

data("chickwts")  
wideCw <- dcast(ChickWeight, Diet + Chick ~ Time, value.var = "weight")  
names(wideCw)[-(1:2)] <- paste("time", names(wideCw)[-(1:2)], sep = "")  
wideCw <- mutate(wideCw, gain = time21 - time0)  
wide14 <- subset(wideCw, Diet %in% c(1,4))  
t.test(gain ~ Diet, paired = FALSE, var.equal = FALSE, data = wide14) #Busco explicar gain a traves de los grupos 1 y 4

##   
## Welch Two Sample t-test  
##   
## data: gain by Diet  
## t = -2.9615, df = 20.937, p-value = 0.007464  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -104.65901 -18.29932  
## sample estimates:  
## mean in group 1 mean in group 4   
## 136.1875 197.6667

#de Diet por eso el "~" (Maximo 2 grupos), las personas del grupo 1 y 4 son disitntas (Paired = FALSE), asumo que la  
#Varianza es distina  
#No contiene al 0, por lo que se rechaza la hipotesis nula  
#El valor T entrega cuantas veces de la desviacion estandar estoy movido

### Calculating Power

“La potencia de una prueba estadística o el poder estadístico es la probabilidad de que la hipótesis nula sea rechazada cuando la hipótesis nula es falsa”

El “poder” me permite saber la probabilidad de encontrar observaciones mas grandes con mayor certeza. Mientras mas observaciones tengo, la probabilidad de encontrar valores de mu mas grandes que la hipotesis nula es mas alta. Los valores extremos son cada vez mas faciles de encontrar

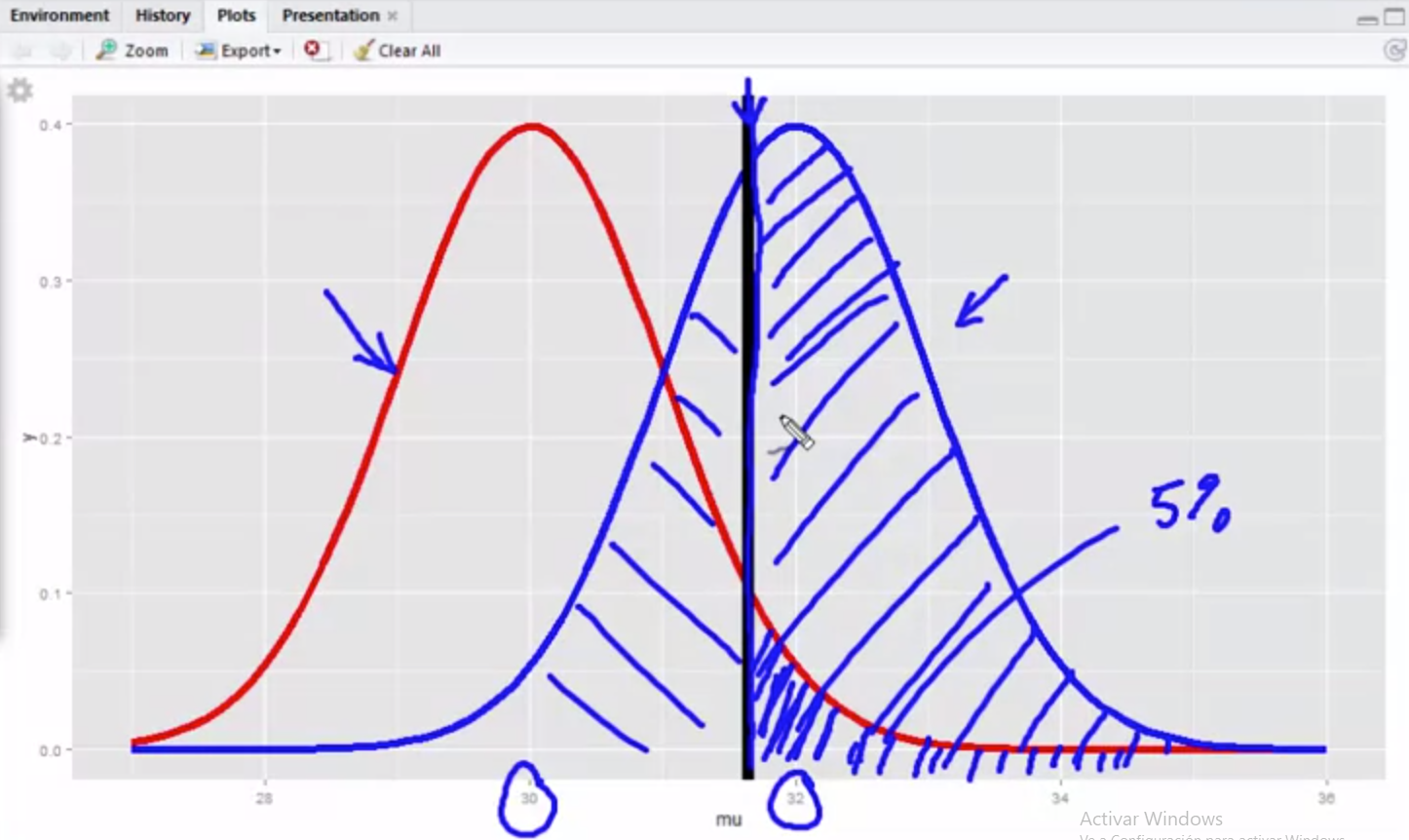
mu0 <- 30  
mua <- 32  
n <- 16  
sigma <- 4  
alpha <- 0.05  
z <- qnorm(1-alpha)  
pnorm(mu0 + z\*sigma/sqrt(n), mean = mu0, sd = sigma/sqrt(n), lower.tail = FALSE)

## [1] 0.05

pnorm(mu0 + z\*sigma/sqrt(n), mean = mua, sd = sigma/sqrt(n), lower.tail = FALSE) #Hay un 64% de probabilidad de detectar

## [1] 0.63876

#una media de 32 o más si llevamos a cabo este experimento



El poder es el area bajo la curva azul y a la derecha de la linea negra, que indica la probabilidad de rechazar la hipotesis nula si es que la curva azul es la correcta. La linea negra corresponde al punto donde la curva roja tiene acumulada un 5% de probabilidad. El otro lado de la curva azul es la probabilidad de un error tipo II. \* Al hacer alpha mas pequeño hacemos que sea mas complicado rechazar la hipotesis nula, se necesita mas evidencia \* Si disminuimos sigma y la variacion es pequeña, la probabilidad de rechazar la hipotesis nula aumenta \* Mientras la hipotesis alternativa esta mas alejada de la nula, la probabilidad es mas alta \* Si aumento n, la varianza disminuye por lo que el poder es mayor

mua-mu0/sigma, es denominado “effect size”

power.t.test(n = 16, delta = 2/4, sd = 1, type = "one.sample", alt = "one.sided") #Delta es la diferencia entre los mu de las hipotesis nulas mua-mu0

##   
## One-sample t test power calculation   
##   
## n = 16  
## delta = 0.5  
## sd = 1  
## sig.level = 0.05  
## power = 0.6040329  
## alternative = one.sided

power.t.test(power = 0.8, delta = 2/4, sd = 1, type = "one.sample", alt = "one.sided")

##   
## One-sample t test power calculation   
##   
## n = 26.13751  
## delta = 0.5  
## sd = 1  
## sig.level = 0.05  
## power = 0.8  
## alternative = one.sided

### Bootstrapping

“Se utiliza para aproximar la distribución en el muestreo de un estadístico. Se usa frecuentemente para aproximar el sesgo o la varianza de un análisis estadístico”

Se usa para aproximar la distribucion estadistica de un muestreo a partir de un expermiento (Tirar el dado 50 veces, y hacerlo solamente una vez)

library(UsingR)

data(father.son)  
x <- father.son$sheight  
n <- length(x)  
B <- 10000  
resamples <- matrix(sample (x, n\*B, replace = TRUE), B,n) #Esto simula nuevamente los datos, con probabilidad 1/n cada uno de los datos  
resampledMedians <- apply(resamples,1,median)