

Estadística II en R

Fiona Franco Churruarín

30 December, 2019

Contents

Probabilidad y Variables Aleatorias	2
Variables aleatorias y sus distribuciones	2
Casos discretos	2
Distribuciones muestrales	5
Momentos	17
Propiedades de FGM:	17
FGM de distintas distribuciones	18
Ley de los Grandes Números y Teorema Central del Límite	18
Estimadores	18
Propiedades	18
Métodos de Estimación	19
Intervalos de Confianza	19
Una población	19
Dos poblaciones	19
Test de Hipótesis	19
Regresión Lineal	19
Mínimos Cuadrados Ordinarios	19
Inferencia estadística en modelos de regresión	19
ANOVA	19
Bondad de Ajuste e Independencia de Atributos	19
Test Ji-Cuadrado	19
Test Kolmogorov-Smirnov	20
Otras medidas de bondad del ajuste	20
Inferencia Bayesiana	22

Probabilidad y Variables Aleatorias

Una variable aleatoria le asigna un número a cada elemento del espacio muestral, es decir, a cada elemento del conjunto de todos los posibles resultados de un experimento aleatorio.

Variables aleatorias y sus distribuciones

Recordemos que una variable aleatoria (en adelante, v.a.) es discreta cuando la imagen de x o su recorrido R_x es un conjunto finito o infinito numerable, y que una v.a. es continua cuando su En esta sección se verá como se implementan las distribuciones de probabilidad que se utilizan en el curso. Es simple notar que todas las distribuciones tienen un patrón común en \mathbb{R} .

Las v.a. discretas tienen asociada una *función de probabilidad*, que devuelve la probabilidad *puntual*, de que la variable aleatoria x tome un determinado valor x_i ,

$$p(x_i) = p(x = x_i)$$

Donde siempre se cumple que

1. $0 \leq p(x_i) \leq 1$
2. $\sum_{i=1}^n p(x_i) = 1$

Las v.a. aleatorias continuas tienen asociadas una *función de densidad*. Como el recorrido de la v.a. es continuo, no podemos encontrar la probabilidad de ocurrencia de un sólo punto, es decir, $f(x = a) = 0$. Para entender esto, recuérdese que no se puede calcular el área bajo un sólo punto de una curva, porque es una sola línea. Por lo tanto, se debe trabajar con la probabilidad de que la variable tome valores entre dos puntos, para poder hallar el área debajo de la función de densidad,

$$p(a \leq x \leq b) = \int_a^b f(x)dx.$$

En donde se cumple

1. $f(x) \geq 0$
2. $\int_{R_x} f(x)dx = 1$

Cada variable aleatoria también tiene una función de distribución $F(x)$

Casos discretos

Distribución de Bernoulli

$x \sim \text{Bern}(p)$ Se utiliza para conocer la probabilidad de ocurrencia de sucesos dicotómicos, es decir, aquellos en los que sólo hay dos resultados posibles. Si llamamos x a la v.a., entonces podemos denotar el espacio muestral de la variable de la siguiente forma,

$$x = \begin{cases} 1 & \text{si éxito} \\ 0 & \text{si fracaso} \end{cases} \quad (1)$$

Denominamos la probabilidad de ocurrencia del éxito $p(x = 1)$ como p , y la probabilidad de éxito del fracaso como $1 - p = q$. Así, podemos escribir la función de probabilidad de una v.a. con distribución de Bernoulli de la siguiente forma,

$$p(x) = p^x q^{1-x}$$

En R esta distribución está codificada con los siguientes comandos

```
#dbinom(x, size, prob, log = FALSE)
#pbinom(q, size, prob, lower.tail = TRUE, log.p = FALSE)
#qbinom(p, size, prob, lower.tail = TRUE, log.p = FALSE)
#rbinom(n, size, prob)

dbinom(x = 1, size = 1, prob = 0.5, log = FALSE)

## [1] 0.5

pbinom(q = 1, size = 1, 0.5, lower.tail = TRUE, log.p = FALSE)

## [1] 1

qbinom(p = 0.5, size = 1, 0.5, lower.tail = TRUE, log.p = FALSE)

## [1] 0

rbinom(n = 1, size = 1, 0.5)

## [1] 1
```

Distribución Binomial

Una v.a. x con distribución binomial, $x \sim Bi(m, p)$, describe el número de éxitos en m ensayos de Bernoulli independientes.

$$p(x) = \binom{m}{x} p^x q^{m-x}$$

```
m <- 5

dbinom(x = 1, size = m, prob = 0.5, log = FALSE)

## [1] 0.15625

pbinom(q = 1, size = m, prob = 0.5, lower.tail = TRUE, log.p = FALSE)

## [1] 0.1875

qbinom(p = 0.5, size = m, prob = 0.5, lower.tail = TRUE, log.p = FALSE)

## [1] 2

#rbinom(n = 1, size = m, prob = 0.5,)
```

Distribución geométrica

```
#dgeom(x, prob, log = FALSE)
#pgeom(q, prob, lower.tail = TRUE, log.p = FALSE)
#qgeom(p, prob, lower.tail = TRUE, log.p = FALSE)
#rgeom(n, prob)
```

$x \sim Ge(p)$ describe el número de ensayos de Bernoulli independientes necesarios hasta obtener el primer éxito.

$$p(x) = pq^{x-1}$$

Distribución de Poisson

$x \sim P(\lambda)$ describe el número de ocurrencias en un continuo (tiempo, volúmen, etc). Se supone que la cantidad de ocurrencias es directamente proporcional al continuo.

$$p(x) = \frac{e^{-\lambda} \lambda^x}{x!}$$

```
#dpois(x, lambda, log = FALSE)
#ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)
#qpois(p, lambda, lower.tail = TRUE, log.p = FALSE)
#rpois(n, lambda)
```

##Casos continuos

Distribución Normal Estándar

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}$$

```
#dnorm(x, mean = 0, sd = 1, log = FALSE)
#pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
#qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
#rnorm(n, mean = 0, sd = 1)
```

Como los valores por default son los de la distribución normal estándar, podemos omitirlos cuando queramos calcular probabilidades provenientes de esta distribución

```
dnorm(0)
```

```
## [1] 0.3989423
```

```
dnorm(2)
```

```
## [1] 0.05399097
```

```
dnorm(c(-1.963, 1.963))
```

```
## [1] 0.05809806 0.05809806
```

```
pnorm(1.963)
```

```
## [1] 0.9751769
```

```
pnorm(-1.963)
```

```
## [1] 0.02482309
```

```
qnorm(0.95)
```

```
## [1] 1.644854
```

```
qnorm(0.975)
```

```
## [1] 1.959964
```

```
qnorm(0.05)
```

```
## [1] -1.644854
```

```
qnorm(0)
```

```
## [1] -Inf
```

```

set.seed(285)
rnorm(5)

## [1] 0.56819281 -0.03564371 1.19478613 -2.61908879 0.56540354

rnorm(2)

## [1] 0.7279098 -0.4013947

rnorm(10)

## [1] 1.1464764 -1.2346869 1.8002508 0.7072287 -0.1439041 -1.7690106
## [7] 1.3905327 -1.4716124 0.4264682 -0.3068373

```

Distribución Normal

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Distribución Uniforme Continua

Distribución Gamma

Casos especiales de gamma:

- $\alpha = 1 \Rightarrow$ Exponencial
- $\alpha = n/2, \quad \beta = 2 \Rightarrow$ Ji-Cuadrado

Distribución Beta

Distribuciones muestrales

Distribución Ji-Cuadrado

Como se vio anteriormente, la distribución Ji-Cuadrado con n grados de libertad es un caso especial de la distribución Gamma, cuando $\alpha = n/2$ y $\beta = 2$.

Para calcular las probabilidades de una distribución Ji-Cuadrado podemos utilizar los siguientes códigos:

```

#df = degrees of freedom = grados de libertad
#dchisq(x, df, ncp, log = FALSE)
#pchisq(q, df, ncp, lower.tail = TRUE, log.p = FALSE)
#qchisq(p, df, ncp, lower.tail = TRUE, log.p = FALSE)
#rchisq(n, df, ncp)

```

La distribución Ji-Cuadrado además puede obtenerse a partir de distribuciones normales estándar. Siendo Z_1, \dots, Z_n variables aleatorias con distribución normal estándar, $Z_i \sim N(0, 1) \forall i = 1, \dots, n$, se define una variable aleatoria J como la suma del cuadrado de variables aleatorias con distribución normal estándar,

$$J = \sum_{i=1}^n Z_i^2$$

Se puede demostrar que J tiene una distribución Ji-Cuadrado con n grados de libertad. La demostración formal de esto se verá posteriormente, aunque ahora podemos utilizar una simulación en R para mostrar esta propiedad.

En primer lugar, podemos mostrar cómo construir realizaciones de una variable aleatoria con distribución normal estándar.

```

set.seed(285)

n <- 10
Z <- rnorm(n, 0, 1)

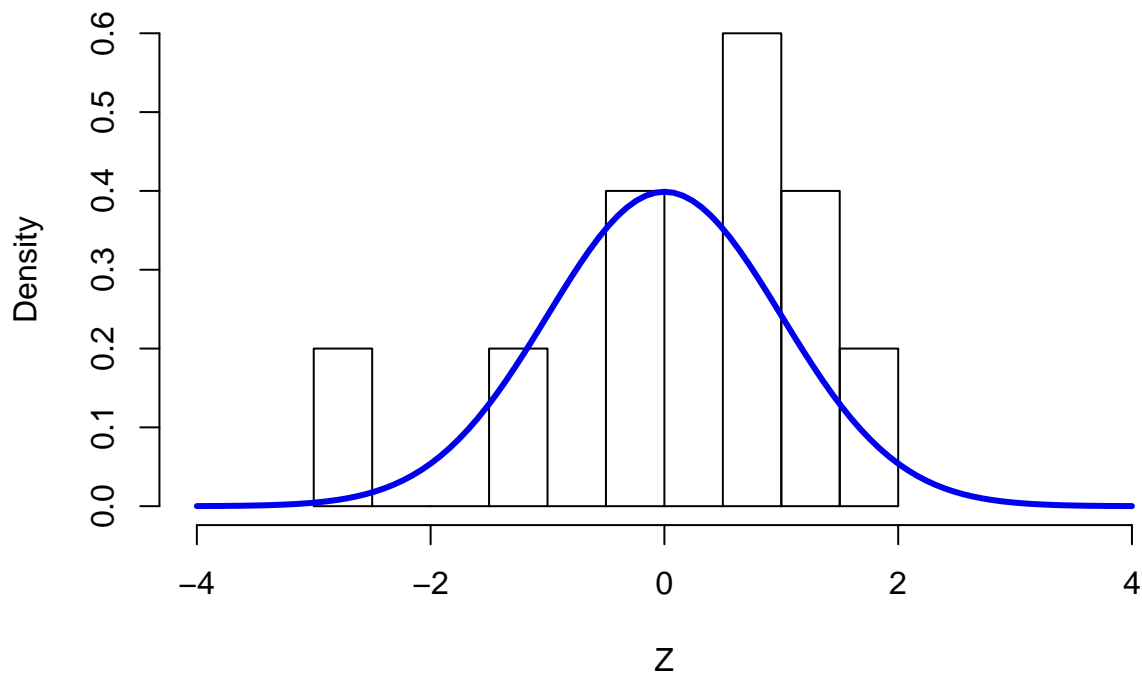
print(Z)

## [1] 0.56819281 -0.03564371 1.19478613 -2.61908879 0.56540354
## [6] 0.72790979 -0.40139468 1.14647640 -1.23468688 1.80025080

x <- seq(-4,4,0.01)
hist(Z, freq = FALSE,
     breaks = 10,
     xlim = c(-4,4),
     main = "Realizaciones de v.a. con distribución normal estándar")
curve(dnorm(x, 0, 1),
      from = -4, to = 4,
      col = "blue2",
      lwd = 3,
      add = TRUE)

```

Realizaciones de v.a. con distribución normal estándar



Ahora bien, una vez que tenemos estos valores, podemos tomar el cuadrado de cada uno de ellos y sumarlos. De acuerdo con lo recién expuesto, esto debería darnos una realización una variable aleatoria que se define de esta forma. Esto sería una sola observación de una variable aleatoria “suma de n variables aleatorias con distribución normal estándar”.

```
print(Z)

## [1] 0.56819281 -0.03564371 1.19478613 -2.61908879 0.56540354
## [6] 0.72790979 -0.40139468 1.14647640 -1.23468688 1.80025080

print(Z^2)

## [1] 0.322843064 0.001270474 1.427513896 6.859626066 0.319681160
## [6] 0.529852656 0.161117693 1.314408143 1.524451698 3.240902928

print(sum(Z^2))

## [1] 15.70167
```

Podríamos repetir este proceso todas las veces que queramos, recolectar los resultados que obtengamos y observar si las distintas realizaciones siguen una distribución Ji-Cuadrado. Para simplificar el trabajo, se puede usar una función que cree un vector en el que cada elemento contenga la suma de los cuadrados de n realizaciones de variables aleatorias con distribución normal estándar. La función `SumZsq` devuelve un vector con m elementos, y en cada una de ellas hay una realización de la variable aleatoria definida como $\sum_{i=1}^n Z_i^2$.

```
SumZsq <- function(n, m){
  # n es la cantidad de N(0,1) que sumo
  # m es el numero de repeticiones que quiero
  v <- vector()
  for(i in 1:m) {
    z <- rnorm(n, 0, 1)
    v[i] <- sum(z^2)
  }
  return(v)
}
```

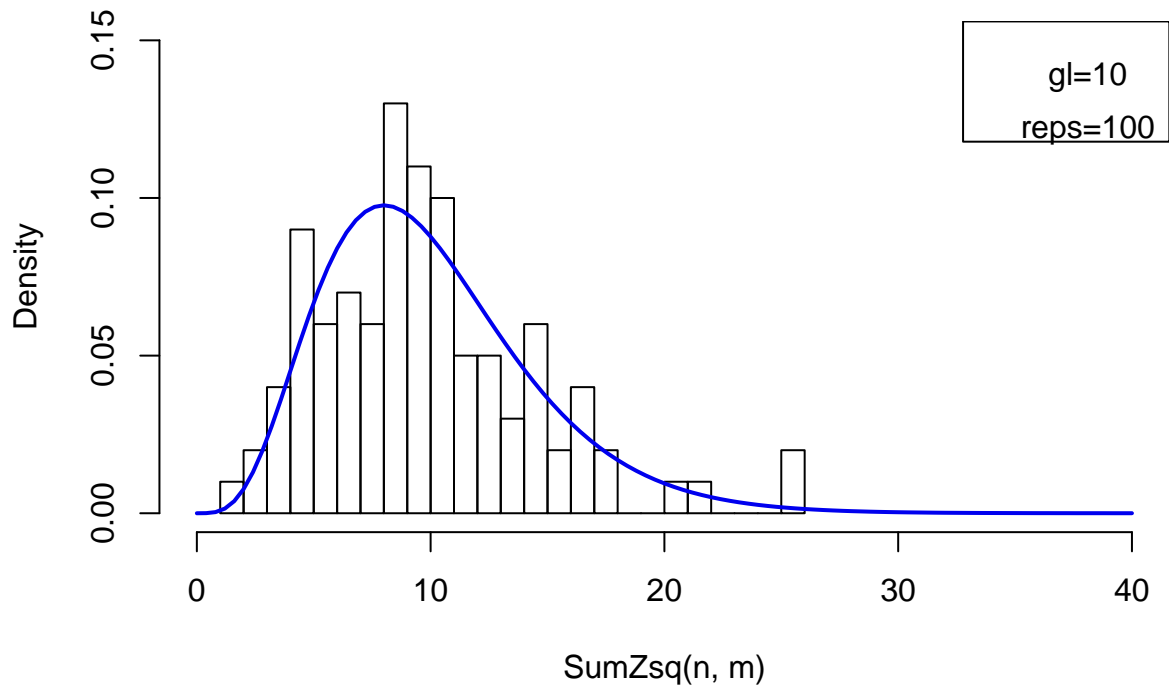
De acuerdo con lo que dice la teoría estadística, esta variable se distribuye Ji-Cuadrado con n grados de libertad ¿cómo podemos ver si esto se cumple? En primer lugar, vamos a utilizar la función que recién creamos para generar un vector que tenga m realizaciones de J , y luego vamos graficar el histograma de los valores de J , junto con una distribución Ji-Cuadrado con n grados de libertad, y luego compararlos.

```
SumZsq(n=10, m=40)

## [1] 15.015513 14.526763 4.979442 16.211373 9.940637 17.793723 6.670770
## [8] 4.816104 7.489471 3.952622 5.361086 4.299929 3.991255 9.590502
## [15] 9.860864 17.042254 12.572875 12.939149 10.432114 7.437532 7.235573
## [22] 6.962744 8.433115 3.148007 15.432692 10.395595 9.406276 5.609586
## [29] 12.185731 9.834225 10.611742 9.431430 7.955094 4.597255 8.615015
## [36] 6.740848 17.981920 10.995885 11.967719 9.488362

n <- 10
m <- 100
hist(SumZsq(n, m), freq = FALSE, breaks=30,
     main = "Suma de cuadrado de realizaciones de N(0,1) \n y distribución Ji-Cuadrado",
     xlim = c(0,40), ylim = c(0,.15))
curve(dchisq(x, df=n),
      from = 0, to = 40,
      col = "blue2",
      lwd = 2,
      add = TRUE)
legend('topright', legend = bquote(atop(paste("gl=", .(n)), paste("reps=", .(m))))))
```

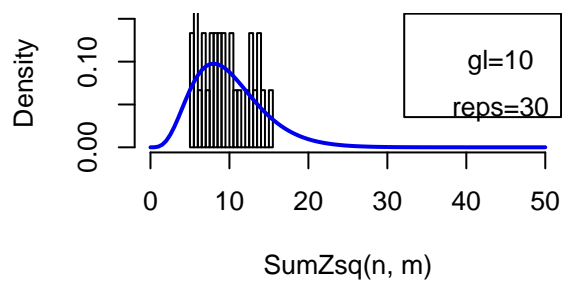
Suma de cuadrado de realizaciones de $N(0,1)$ y distribución Ji-Cuadrado



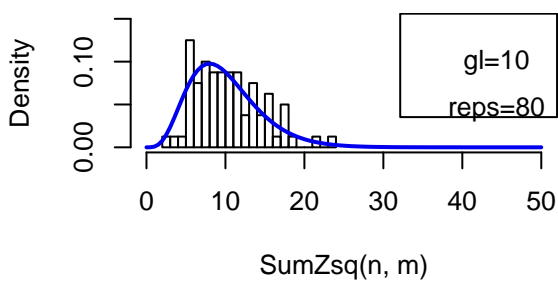
Probando distintos valores de m , es decir, cantidad de repeticiones (con n fijo en 10)

```
par(mfrow=c(2,2))
for (m in c(30, 80, 150, 275, 500)){
  set.seed(10)
  n <- 10
  hist(SumZsq(n, m), freq = FALSE, breaks=20,
       xlim = c(0,50), ylim = c(0,.15))
  curve(dchisq(x, df=n),
        from = 0, to = 50,
        col = "blue2",
        lwd = 2,
        add = TRUE)
  legend('topright', legend = bquote(atop(paste("gl=", .(n)), paste("reps=", .(m))))))
}
```

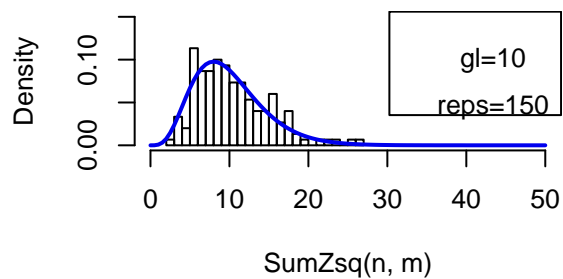

Histogram of SumZsq(n, m)



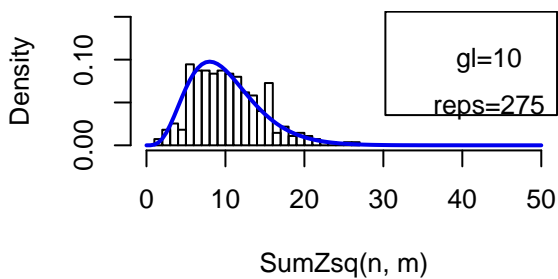
Histogram of SumZsq(n, m)

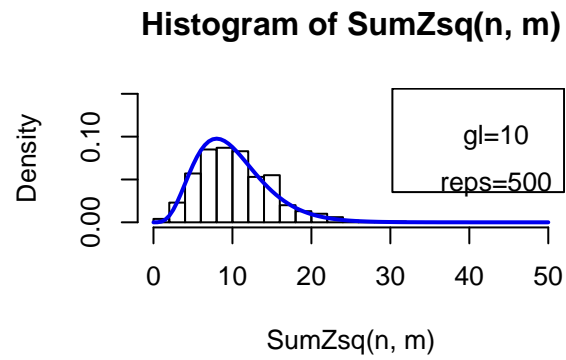


Histogram of SumZsq(n, m)



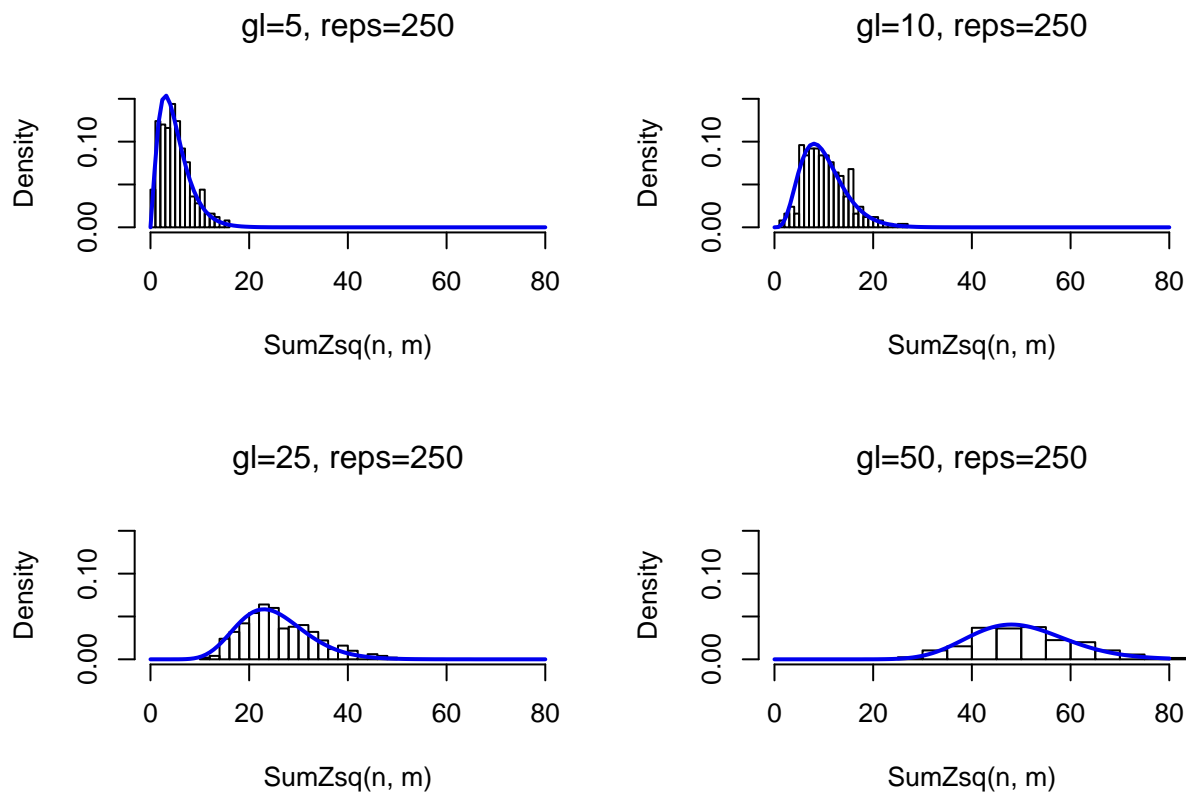
Histogram of SumZsq(n, m)





Probando con distintos valores de n , es decir, cantidad de normales estándar que estamos sumando (con m fijo en 250)

```
par(mfrow=c(2,2))
for (n in c(5, 10, 25, 50)){
  set.seed(10)
  m <- 250
  hist(SumZsq(n, m), freq = FALSE, breaks=20,
       main = bquote(paste("gl=", .(n), ", reps=", .(m))),
       xlim = c(0,80), ylim = c(0,.15))
  curve(dchisq(x, df=n),
        from = 0, to = 80,
        col = "blue2",
        lwd = 2,
        add = TRUE)
}
```



Haciendo simulaciones, parece cumplirse que la suma de los cuadrados variables aleatorias normales estándar se distribuya sea una variable aleatoria que se distribuye Ji-Cuadrado con n grados de libertad.

Distribución t de Student

Siendo $Z \sim N(0, 1)$ y $J \sim \chi_n^2$, una variable aleatoria con distribución t de Student se define como el cociente entre una variable aleatoria normal estándar y la raíz cuadrada de una variable aleatoria Ji-Cuadrado sobre sus grados de libertad

$$T = \frac{Z}{\sqrt{J/n}},$$

Las funciones de densidad, probabilidad acumulada, cuantiles y generación de números aleatorios en R son las siguientes:

```
#df = degrees of freedom = grados de libertad
#dt(x, df, ncp, log = FALSE)
#pt(q, df, ncp, lower.tail = TRUE, log.p = FALSE)
#qt(p, df, ncp, lower.tail = TRUE, log.p = FALSE)
#rt(n, df, ncp)
```

Se puede mostrar a través de simulaciones que se cumple la relación entre la distribución t y las distribuciones normal estándar y Ji-Cuadrado. Para esto, vamos a seguir el mismo procedimiento realizado anteriormente, aunque con menos “detalle” y una mejor forma de implementarlo en R.

```
set.seed(285)

draw.T <- function(n){
```

```

# n es grados de libertad
Z <- rnorm(1, 0, 1)
J <- rchisq(1, n)
T <- Z / sqrt(J/n)
return(T)
}

draw.T(5);draw.T(5);draw.T(5);draw.T(5);draw.T(5);draw.T(5);draw.T(5)

## [1] 0.6433665
## [1] 0.8512115
## [1] 0.9484228
## [1] -0.4790173
## [1] -0.6630038
## [1] 0.4028808
## [1] -1.429738

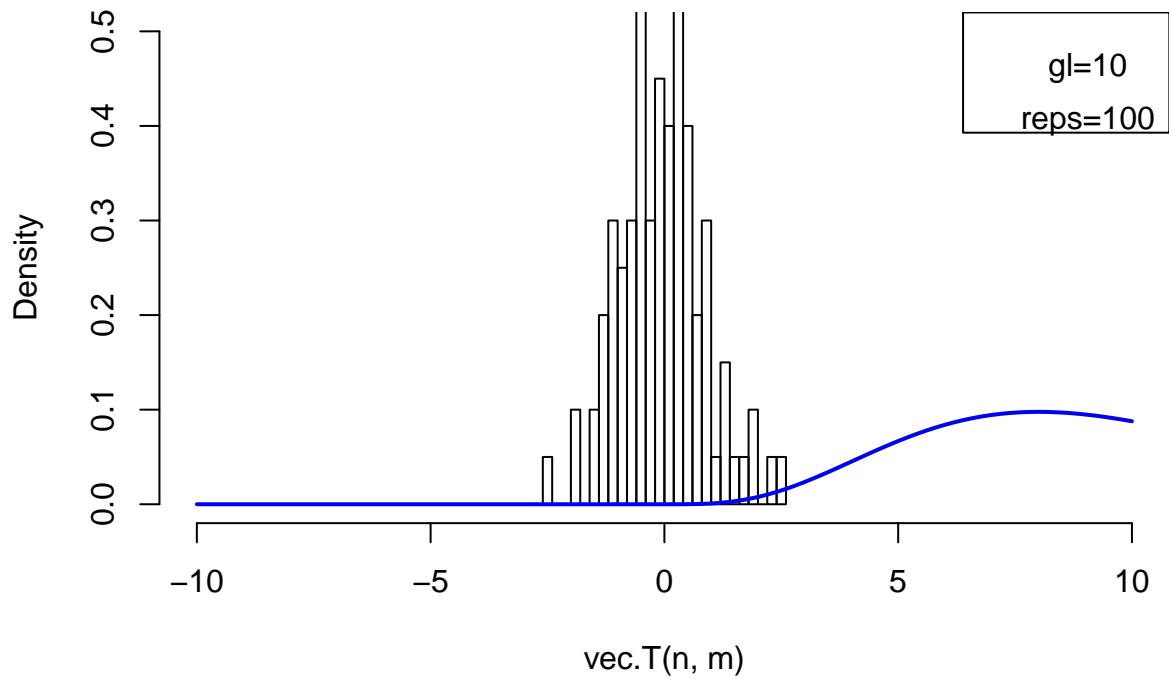
vec.T <- function(n, m){
  # n: grados de libertad
  # m: el numero de repeticiones que quiero
  v <- vector()
  for(i in 1:m){
    v[i] <- draw.T(n)
  }
  return(v)
}

n <- 10
m <- 100

hist(vec.T(n, m), freq = FALSE, breaks=30,
     main = "Suma de cuadrados de realizaciones de N(0,1) \n y distribución Ji-Cuadrado",
     xlim = c(-10,10), ylim = c(0,.5))
curve(dchisq(x, df=n),
      from = -10, to = 10,
      col = "blue2",
      lwd = 2,
      add = TRUE)
legend('topright', legend = bquote(atop(paste("gl=", .(n)), paste("reps=", .(m))))

```

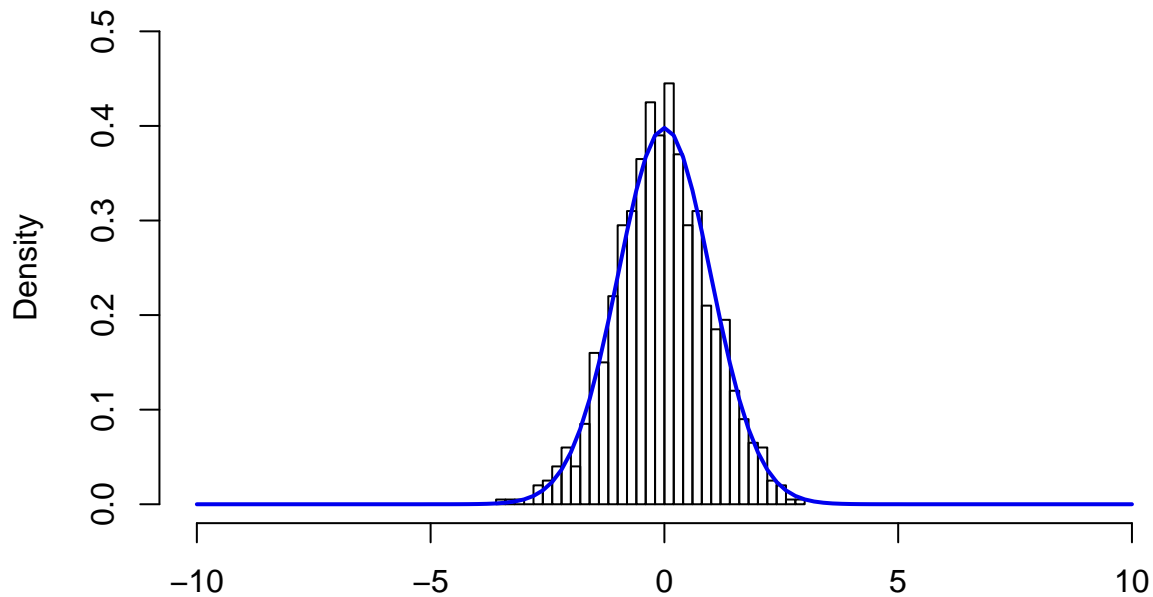
Suma de cuadrados de realizaciones de $N(0,1)$ y distribución Ji-Cuadrado



```
Plot1Simul.T <- function(n,m){
  hist(vec.T(n, m), freq = FALSE, breaks=30,
    main = bquote(paste("gl=", .(n), ", reps=", .(m))),
    xlab = "",
    xlim = c(-10,10), ylim = c(0,.5))
  curve(dt(x, df=n),
    from = -10, to = 10,
    col = "blue2",
    lwd = 2,
    add = TRUE)
}
```

```
Plot1Simul.T(n=100,m=1000)
```

gl=100, reps=1000



```
# cambiando el número de repeticiones

PlotSimulReps.T <- function(n, m.opt, p.rows, p.cols){
  # n: grados de libertad
  # m.opt: vector de numero de distintas repeticiones
  # p.row: numero de filas en el plot dividido
  # p.col: numero de columnas en el plot dividido
  # restriccion: length(m.opt) = p.row * p.col
  if(length(m.opt)<9){p.cols.def<-2} else {p.cols.def<-3}
  if(missing(p.rows) | missing(p.cols)){
    p.cols = p.cols.def
    p.rows = ceiling(length(m.opt)/p.cols.def)}
  par(mfrow=c(p.rows,p.cols), mar=c(2,2,2,2))
  if(missing(p.rows) | missing(p.cols)){
    p.cols = 2
    p.rows = ceiling(length(m.opt)/2)}
  par(mfrow=c(p.rows,p.cols))
  for (m in m.opt){
    Plot1Simul.T(n,m)
  }
}

PlotSimulReps.T(n=8, m.opt = c(20,50,100,500,1000))

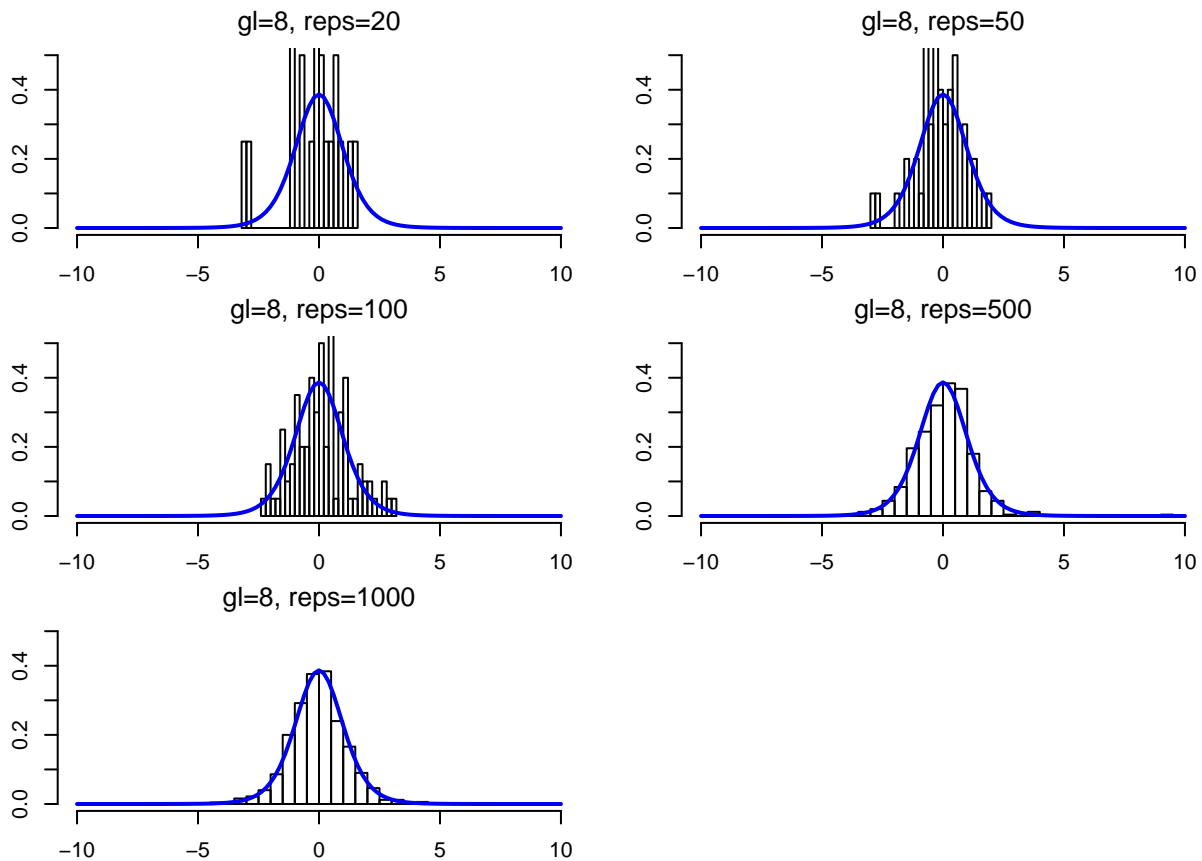
# cambiando los grados de libertad
```

```

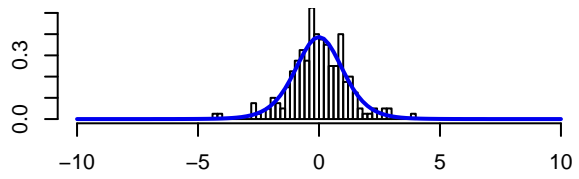
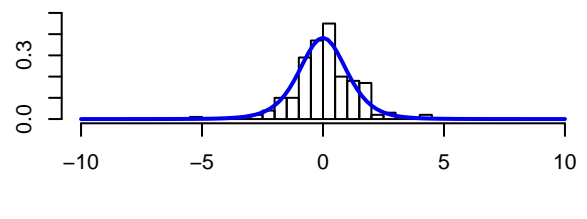
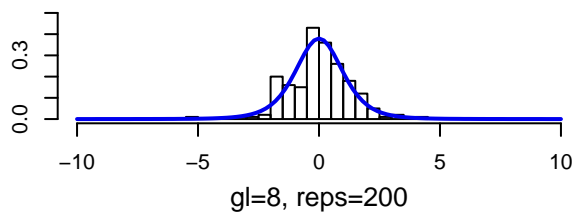
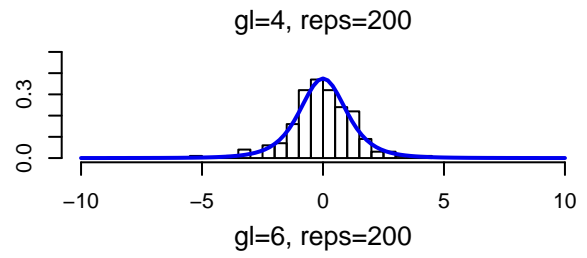
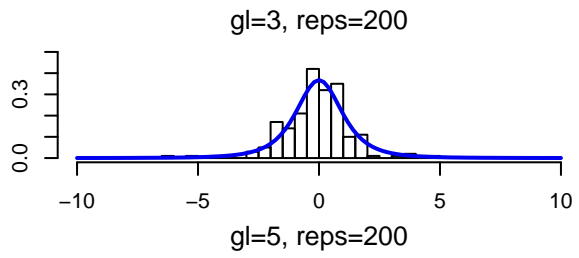
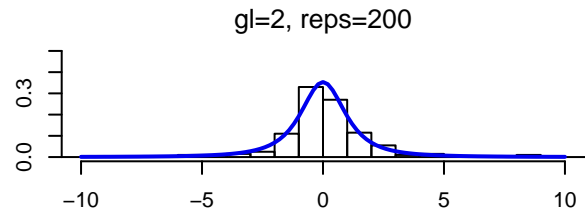
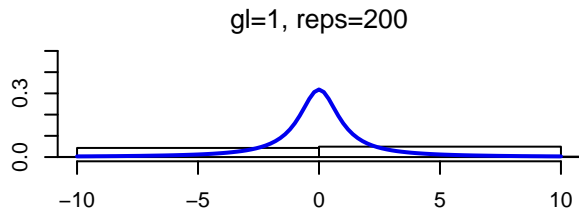
PlotSimulDF.T <- function(m, n.opt, p.rows, p.cols){
  # n: grados de libertad
  # m.opt: vector de numero de distintas repeticiones
  # p.row: numero de filas en el plot dividido
  # p.col: numero de columnas en el plot dividido
  # restriccion: length(m.opt) = p.row * p.col
  if(length(n.opt)<9){p.cols.def<-2} else {p.cols.def<-3}
  if(missing(p.rows) | missing(p.cols)){
    p.cols <- p.cols.def
    p.rows <- ceiling(length(n.opt)/p.cols.def)}
  par(mfrow = c(p.rows,p.cols), mar = c(2,2,2,2))
  for (n in n.opt){
    Plot1Simul.T(n,m)
  }
}

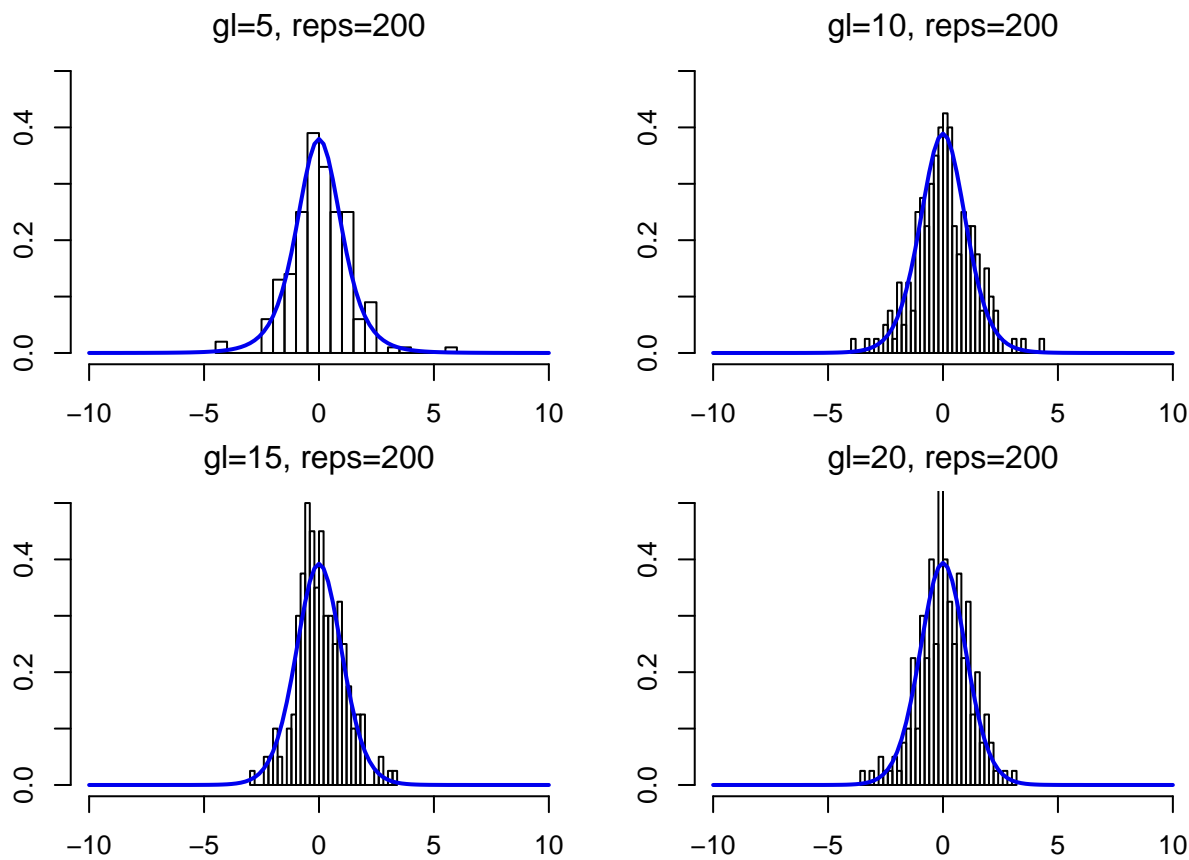
```

```
PlotSimulDF.T(m=200, n.opt = c(1,2,3,4,5,6,8))
```



```
PlotSimulDF.T(m=200, n.opt = c(5,10,15,20))
```





Distribución F de Fisher o de Snedecor

Siendo $J_1 \sim \chi^2(n_1)$ y $J_2 \sim \chi^2(n_2)$ dos distribuciones Ji-Cuadrado con respectivamente n_1 y n_2 grados de libertad, se define la distribución F como el cociente de dos distribuciones Ji-Cuadrado escaladas por sus grados de libertad.

$$F = \frac{J_1/n_1}{J_2/n_2},$$

Las funciones en R para la función de densidad, función de probabilidad acumulada, función de cuantiles y generación de números aleatorios son:

```
#df = degrees of freedom = grados de libertad
#dt(x, df_num, df_denom, ncp, log = FALSE)
#pt(q, df_num, df_denom, ncp, lower.tail = TRUE, log.p = FALSE)
#qt(p, df_num, df_denom, ncp, lower.tail = TRUE, log.p = FALSE)
#rt(n, df_num, df_denom, ncp)
```

A través de simulaciones, se puede verificar que una distribución F se define de esta forma

Momentos

Propiedades de FGM:

Sean X , Y dos variables aleatorias independientes, y a , b dos números reales.

- $\varphi_{aX}(t) = \varphi_X(at)$

- $\varphi_{X+Y}(t) = \varphi_X(t) \cdot \varphi_Y(t)$
- $\varphi_{aX+b}(t) = e^{bt} \varphi_{aX}(t)$

Pensar qué se puede hacer con simulaciones

FGM de distintas distribuciones

Distribución Binomial, o Bernoulli con $m = 1$

$$\varphi_x(t) = (pe^t + q)^m$$

Distribución de Poisson

$$\varphi_x(t) = e^{-\lambda(1-e^t)} = e^{\lambda(e^t-1)}$$

Distribución geométrica

$$\varphi_x(t) = \frac{pe^t}{1 - qe^t}$$

Distribución normal estándar

$$\varphi_x(t) = e^{-\frac{1}{2}t^2}$$

Distribución normal

$$\varphi_x(t) = e^{\mu t + \frac{1}{2}\sigma^2 t^2}$$

Distribución gamma

$$\varphi_x(t) = (1 - \beta t)^{-\alpha}$$

Casos especiales:

- Ji-Cuadrado

$$\varphi_x(t) = (1 - 2t)^{-n/2}$$

- Exponencial

$$\varphi_x(t) = (1 - \beta t)^{-1} = \frac{1}{1 - \beta t}$$

Ley de los Grandes Números y Teorema Central del Límite

Estimadores

Propiedades

Insesgamiento

Sesgo

Insesgamiento Asintótico

Consistencia

Eficiencia relativa

Varianza

Error cuadrático medio

Eficiencia en sentido absoluto

Cota de Rao-Cramer

Métodos de Estimación

Método de momentos

Estimación de Máxima Verosimilitud

Esto estoy segura que se hace. al menos en py

Intervalos de Confianza

Una población

Método de la cantidad pivotal

Media μ de una variable aleatoria x con distribución normal con σ conocido

Media μ de una variable aleatoria x con distribución normal con σ desconocido

Varianza σ^2 de una variable aleatoria x con distribución normal con σ desconocido

IC asintótico para proporción p de una variable aleatoria con distribución Bernoulli

Dos poblaciones

IC para diferencia de medias $\mu_1 - \mu_2$ de dos poblaciones x, y con distribuciones normales independientes con desvíos estándar σ_1 y σ_2 conocidos

IC para diferencia de medias $\mu_1 - \mu_2$ de dos poblaciones x, y con distribuciones normales independientes con desvíos estándar σ_1 y σ_2 desconocidos pero supuestamente iguales, $\sigma_1 = \sigma_2 = \sigma$

Test de Hipótesis

Regresión Lineal

Mínimos Cuadrados Ordinarios

Inferencia estadística en modelos de regresión

Acá basarme un poco en lo que tengo de Ezequiel, ver que puedo condimentar con las cosas de Econometría (creo que poco)

Ver si puedo armar funciones que computen todo lo que hace MCO. Dejarlo como ejercicio y al final ponerlo como solución

ANOVA

Bondad de Ajuste e Independencia de Atributos

En la literatura de econ vi que usan estos tests para probar la normalidad de distribuciones. Por ejemplo, Reinhart y Rogoff.

Test Ji-Cuadrado

Ajuste con distintas variables

Test Kolmogorov-Smirnov

ks.test

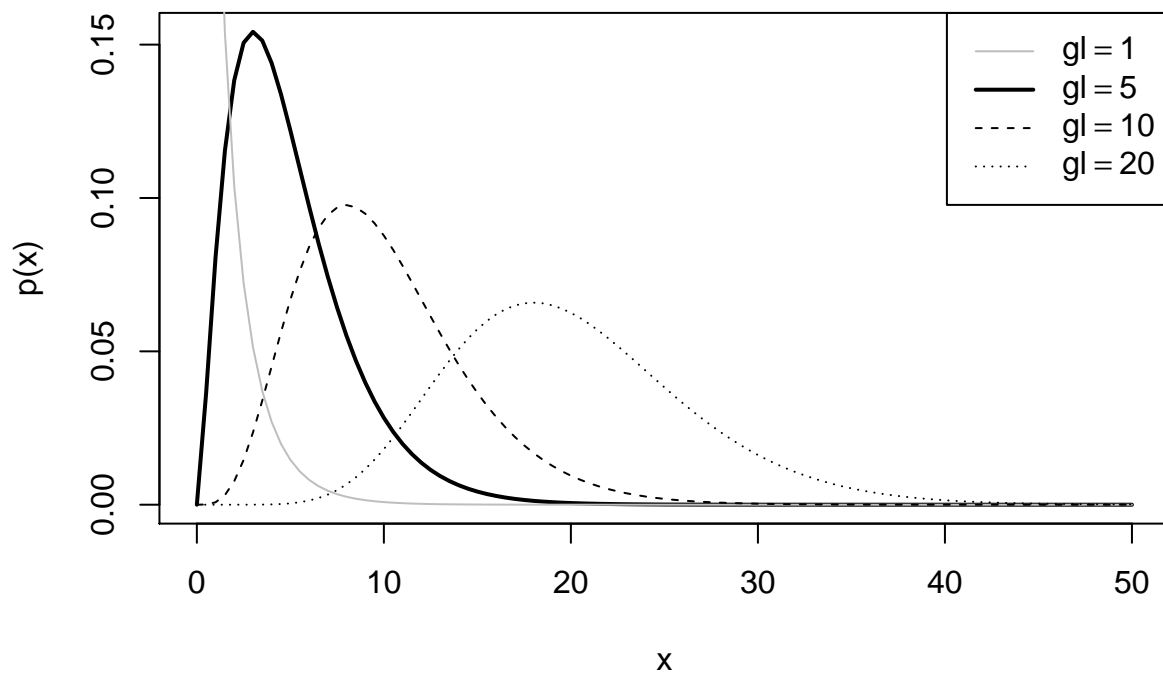
Otras medidas de bondad del ajuste

```
x <- seq(1:50)

curve( dchisq(x, df=5), col='black', lwd = 2,
       from = 0, to = 50,
       ylab = 'p(x)',
       main = 'Distribución Ji-Cuadrado')
curve( dchisq(x, df=1), col='gray', add=TRUE)
curve( dchisq(x, df=10), col='black', add=TRUE, lty = 2)
curve( dchisq(x, df=20), col='black', add=TRUE, lty = 3 )

legend('topright',
       expression(gl==1, gl==5, gl==10, gl==20),
       lwd = c(1, 2, 1, 1),
       lty = c(1,1, 2, 3),
       col = c('grey', 'black', 'black', 'black'))
```

Distribución Ji-Cuadrado



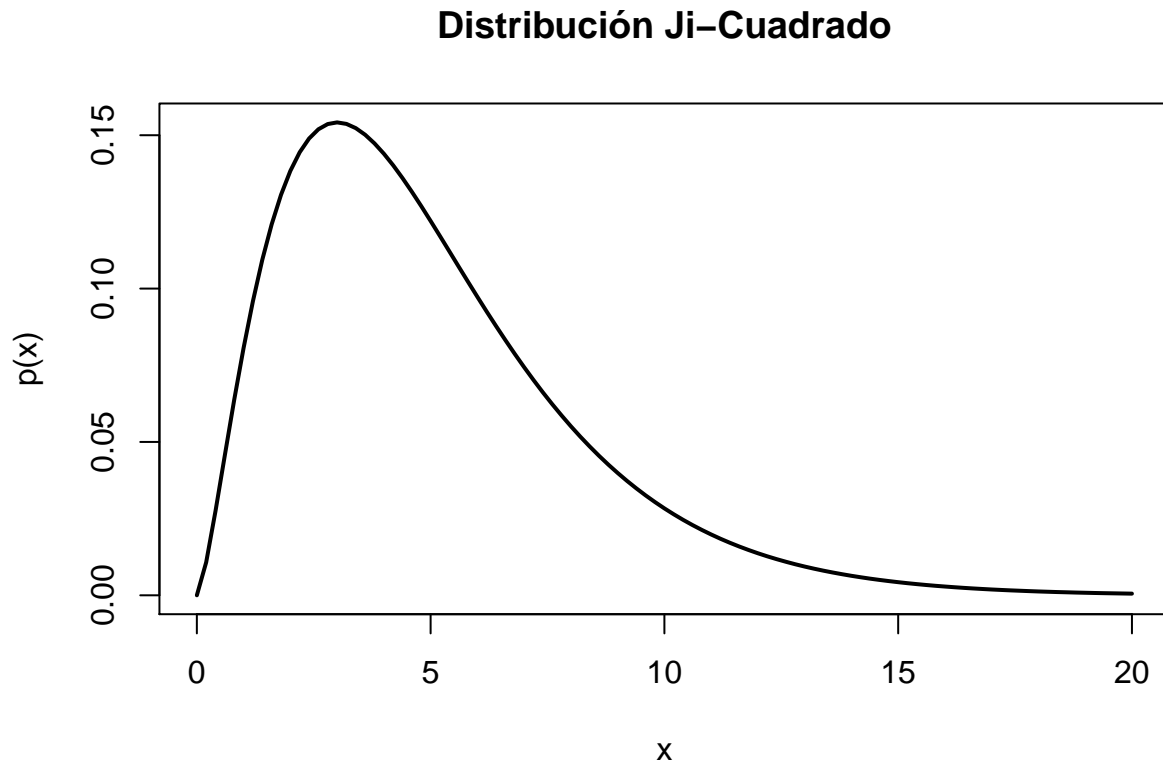
```
# Armar tabla
# Data wrangling para calcular el estadístico
# Test

curve( dchisq(x, df=5), col='black', lwd = 2,
```

```

from = 0, to = 20,
ylab = 'p(x)',
main = 'Distribución Ji-Cuadrado')

```



```

#Graficando la región crítica

#Selección de grados de libertad
df <- 6
#Selección de error tipo I
alpha <- 0.10
#Defino valor del eje de las abcisas (será un valor del estadístico) a partir del cual rechazo H0
rc <- qchisq(1 - alpha, df)

#Lo que haremos para graficar la región crítica será construir un polígono con la forma de la distribución

#Defino el eje x (técnicamente debería seguir hasta infinito, pero a fines del gráfico se corta antes)
x <- seq(0,30,0.01)
#Defino la región crítica en el eje
z <- seq(rc,30,0.01)
#Defino el "lado" superior del polígono, que será una curva con la forma de la distribución en el segundo eje
p <- dchisq(z, df)
#redefino los vectores de lados para la función polígono
z <- c(z,30,rc)
p <- c(p,min(p),min(p))

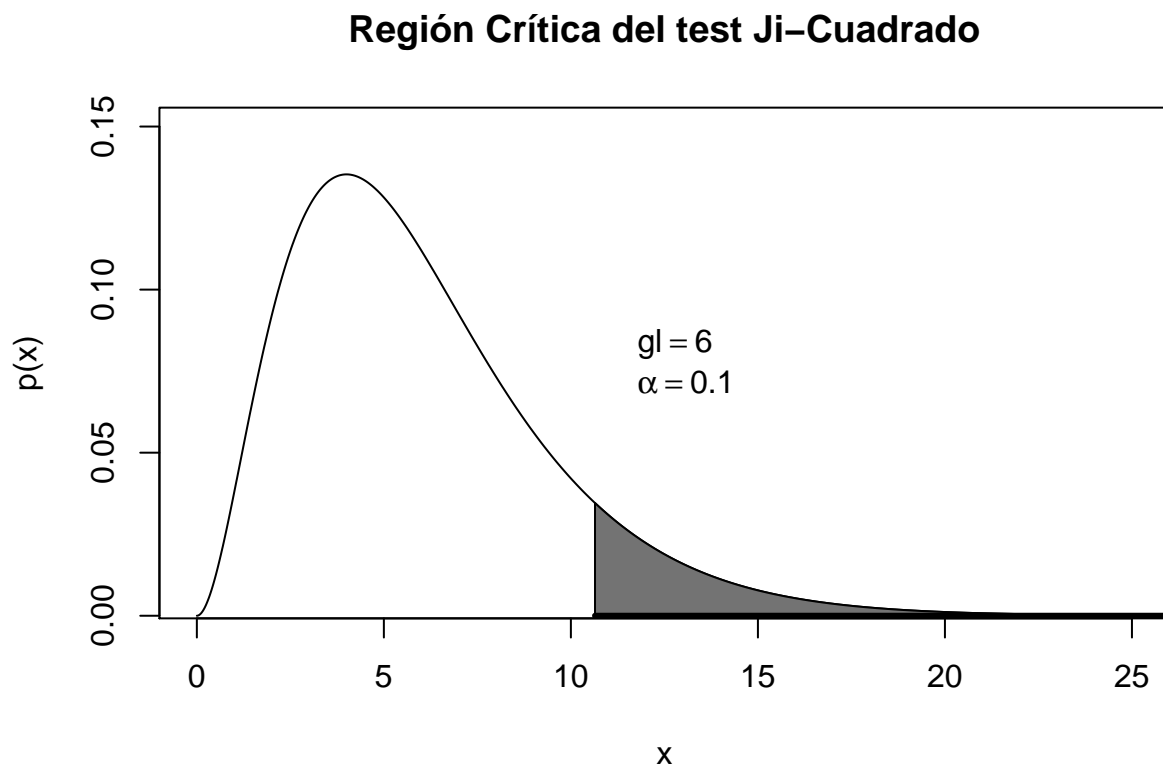
#Grafico distribución Ji-Cuadrado

```

```

plot(x,dchisq(x, df),type="l",
     ylab="p(x)",xlab="x",
     ylim=c(0.005,0.15), xlim = c(0,25),
     main = "Región Crítica del test Ji-Cuadrado")
#ylim sólo está para que quede el eje x bien cruzado con el cero
#Agrego al gráfico el polígono
polygon(z,p,col="gray45")
#Agrego una línea que define la región crítica
segments(x0=rc, y0=0, x1=30, y1=0, lwd=3)
legend('center',
      expression(gl==6, alpha==0.10),
      bty = "n")

```



Hablar del R^2 ? QQ plots? Jarque Bera?

Inferencia Bayesiana