1 Poisson

Se tiene que si $\lambda > 0$ y $p = \lambda/n$, entonces para todo entero $r \geq 0$ tenemos

$$\lim_{n\to\infty} \binom{n}{r} p^r (1-p)^{n-r} = \frac{e^{-\lambda}\lambda^r}{r!}$$

Este conjunto de límites demuestra que si $(X_n)_{n\geq 1}$ es una secuencia de variables aleatorias donde X_n tiene distribución Binomial(n,λ/n) entonces $(X_n)_{n\geq 1}$ converge en distribución a una variable X con distribución Poisson(λ).

2 Normal

Fijamos $p \in (0,1)$ y definimos, para todo entero $n \geq 1$, X_n =Binomial(n,p) y

$$Y_n = \frac{X_n - np}{\sqrt{np(1-p)}}.$$

Entonces se tiene que $(Y_n)_{n\geq 1}$ converge en distribución a Y, donde Y es una variable aleatoria con distribución normal estándar.

3 Ejercicios

1. Confirmaremos gráficamente que si $\lambda > 0$, $p = \lambda/n$ y n es *muy grande*, entonces para todo entero $r \geq 0$

$$\binom{n}{r} p^r (1-p)^{n-r} \sim \frac{e^{-np}(np)^r}{r!}$$

   - Para $\lambda = 5, 10$ y para tres n distintos, grafiquen ambas funciones (de r) superpuestas. En este ejercicio puedes usar *dbinom()* y *dpois()*.
     Para estas gráficas se sugiere usar las siguientes funciones.

     ```
     plot(,type = 's',col= 'red')
     par(new=TRUE)
     lines(,type= 's',col = 'blue')
     ```

     Con sus correspondientes parámetros, genera algo como la figura 1.
     Usen colores para distinguir las gráficas.
   - ¿Para λ fijo, qué valores de n hacen una buena la aproximación?
   - Si X_n =Binomial(n,p) y X =Poisson(λ), calcula numéricamente el siguiente error (justifica tus resultados):

$$\max_{k\geq 0}\{|P(X_n = k) - P(X = k)|\}$$

2. Para el caso de que $p \in (0{,}1)$ sea fijo y la convergencia (de las variables reescaladas) sea a una variable normal. Compara las funciones de probabilidad y densidad, usando (a) ó (b):

   [label=()]X_n y $X = N(np, \sqrt{np(1-p)})$ Y_n y $Y = N(0{,}1)$

   (b) Probar con $p = 0.1$, 0.6 y 0.85,
   - ¿Para qué valores de n la dirías que ya tienes una buena aproximación?
   - Propón un criterio numérico alternativo al *máx* para medir la velocidad de la convergencia.

Observación: Para calcular la función de probabidad de Y_n solo se tiene que centrar la gráfica de $P(X_n = k) = P(Y_n = \frac{k-np}{\sqrt{np(1-p)}})$

Aproximacion Binomial-Poisson

Generaremos variables aleatorias con numeros de enseyos variables para ir graficando cada uno de ella con la distribución de binomial, al mismo tiempo, definiremos un λ con valores de 5 y 10. Una vez definido nuestro lambda, podemos conocer la propabilidad de exito para nuestro variables binomiales:

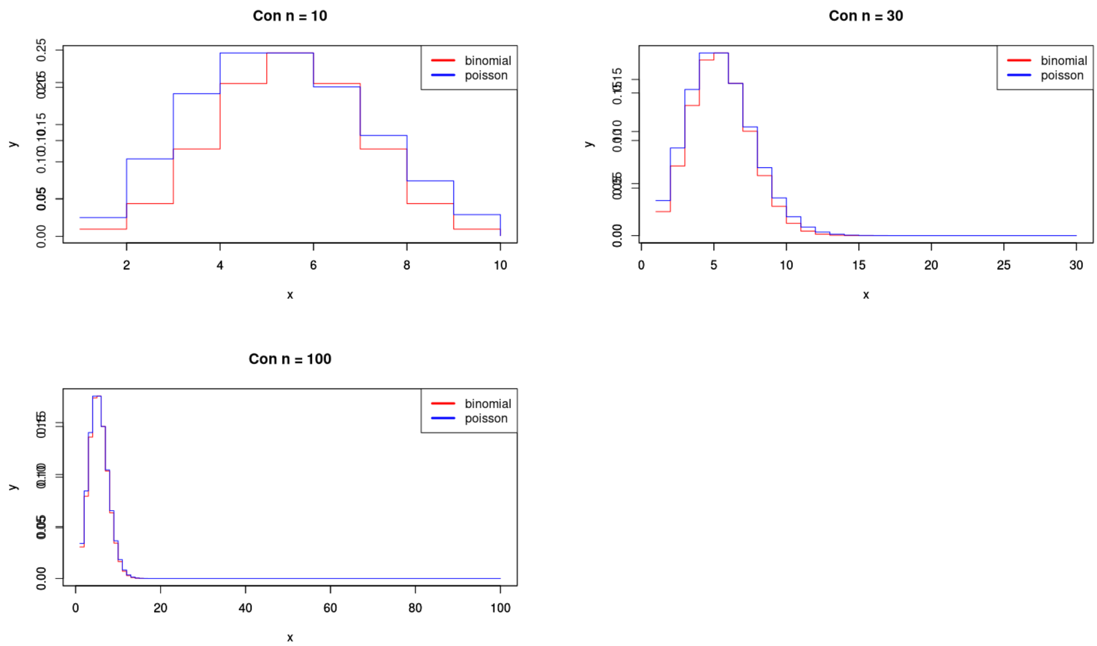
$$\lambda = np$$
$$p = \frac{\lambda}{n}$$

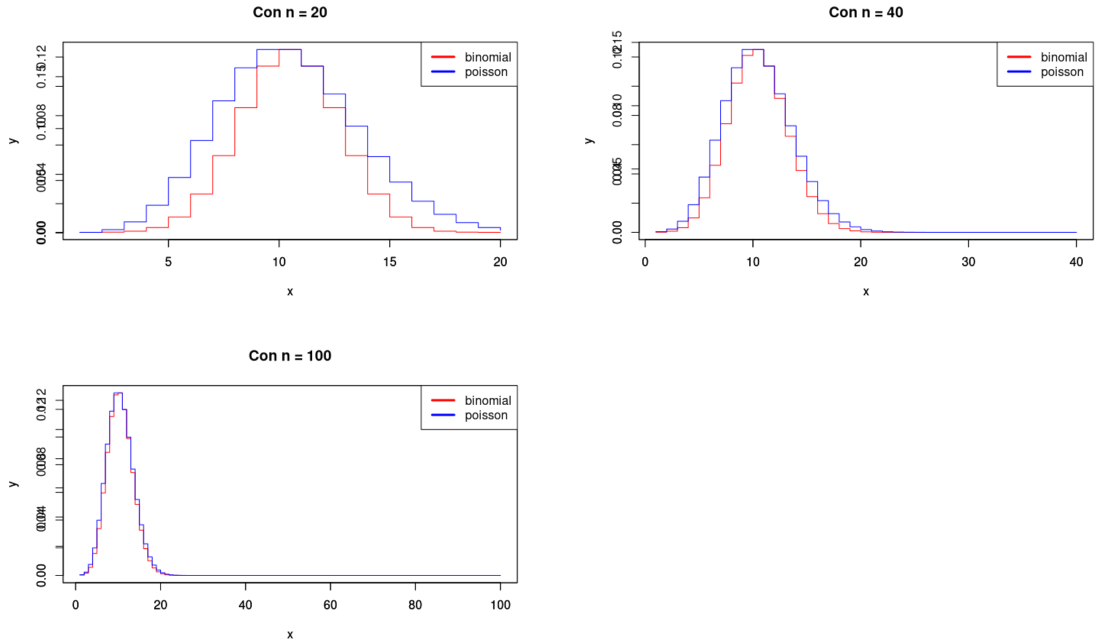
```r
#' Compara las funciones de probabilidad binomial y poisson
#'
#' @param x # Variables aleatorias
#' @param n # Numero de ensayos
#' @param lambda  # Promedio de ocurrencias
#'
#' @return None
comparar_funciones <- function(x, n, lambda) {

  # lambda = np (promedio de ocurrencias)
  # p = lambda / n
  p = lambda / n # Probabilidad de xito

  # Variable aleatorias binomiales
  # Modela: Numero de xitos en n ensayos.
  y <- dbinom(x, n, p)
  plot(x, y, type = 's', col = 'red')

  par(new = 'True')

  # Variables aleatorias poisson
  # Modela: Numero de ocurrencias de eventos muy pocos problables, pero
  # tienen muchos individuos que pueden activarlo
```

```
  y <- dpois(x, lambda)
  plot(x, y, type = 's', col = 'blue')

  legend("topright", legend = c("binomial", "poisson"),
         lwd = 3, col = c("red", "blue"))
}
```

En nuestro codigo usaremos las funciones incorporadas para la funcion de masa ‘dbinom‘ y ‘dpois‘, solo necesitaresmo para un vector ‘x‘ que representen un conjunto de valores de variables aleatorias, una ‘n‘ para el número de ensayos y por último un λ para la distribución de Poisson. Con esto parametros podremos conocer la probabilidad éxito para la distribución binomial.

```
# Para lambda = 5
lambda = 5
n = 10
x = 1:n
comparar_funciones(x, n, lambda)

n = 30
x = 1:n
comparar_funciones(x, n, lambda)

n = 100
x = 1:n
comparar_funciones(x, n, lambda)

#---------------------------------------------------------
# Para lamda = 10
lambda = 10
n = 20
x = 1:n
comparar_funciones(x, n, lambda)

n = 40
x = 1:n
comparar_funciones(x, n, lambda)

n = 100
x = 1:n
comparar_funciones(x, n, lambda)

#---------------------------------------------------------
```



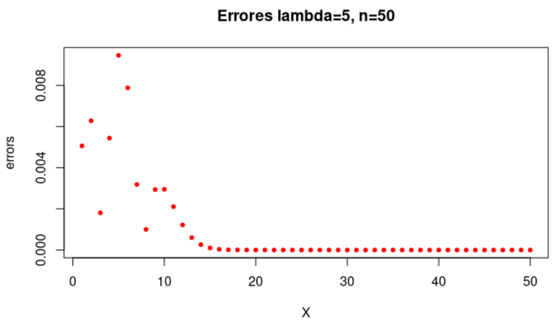


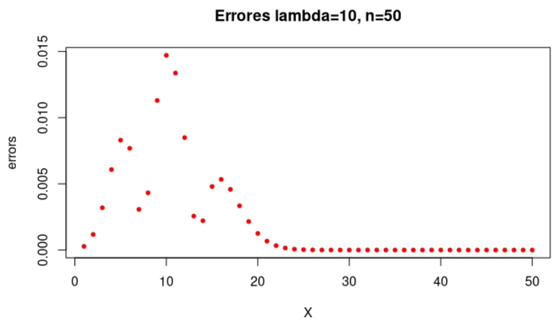
Observamos que con un $n = 10$ las dos distribuciones tiene un desfase, pero conforme vamos aumentando la n ese desfase va disminuyendo de tal modo que cuando tenemos $n = 100$ parece ser casi la misma grafica.

Calculando el error

```
# Encontrando el error
lambda = 5
```

```
n = 50
X = 1:n
p = lambda / n # Probabilidad de xito
errors = abs(dbinom(X, n, p) - dpois(X, lambda))
plot(X, errors, col='red', pch=20, main="Errores lambda=5, n=50")
# Error maximo
print(max(errors))
```

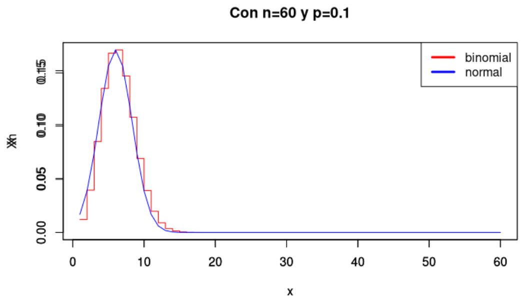


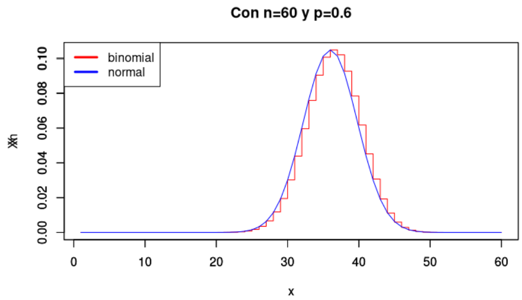


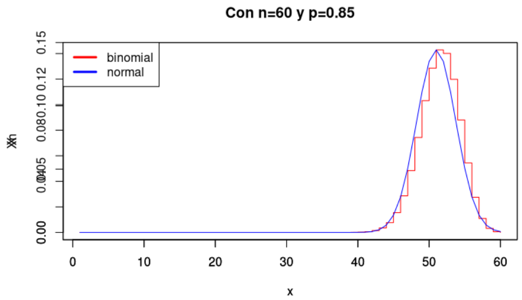
Vemos que al principio el margen de error varia, pero conforme se aumenta el valor de la variable aleatoria converge a un valor con un error significativo. Al analizar las gráficas el error máximo que podemos encontrarnos sera de 0.009457231 y despues converga a un error minimo, podemos también ver al tener una $n = 20$ el error sera mínimo, siendo cada vez mejor. Al igual con $\lambda = 5$ tambien sucedera con $\lambda = 10$ el valor del error convergerá aproximadamente con una $n = 30$.

Ahora aproximaremos con la funcion normal, tomando tres p distintintas:

```
comparar_binom_normal <- function(x, n, p, titulo) {
  Xn = dbinom(x, n, p)
  plot(x, Xn, type = 's', col = 'red', main=titulo)
  par(new = 'True')

  X = dnorm(x, n*p, sqrt(n*p*(1-p)))

  plot(x, X, type = 'l', col = 'blue')
}
n = 60
p = 0.1
x = 1:n
comparar_binom_normal(x, n, p, "Con n=60 y p=0.1")
legend("topright", legend = c("binomial", "normal"),
       lwd = 3, col = c("red", "blue"))
```

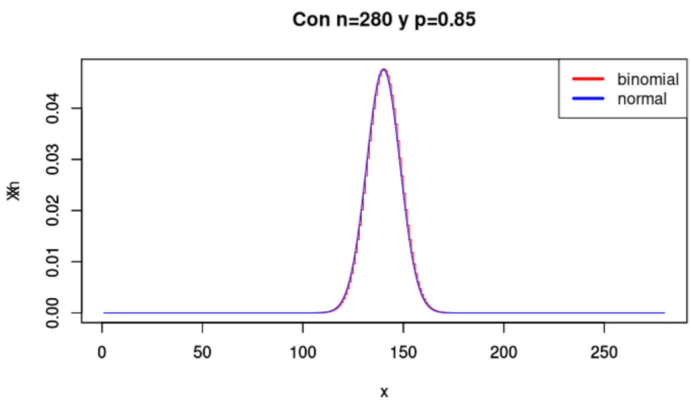






Observamos que con tres probabilidades distintas la graficas van cambiando acercándose al eje y y alejandose de el, vemos que cuando la probabilidad de exito es de 0.1 la forma de la gráfica esta más cerca del eje y y la comparación de la distribucion normal esta adentro de la grafica de la distribución binomial, cuando vamos aumentando la probabilidad la gráfica se va desplazando hacia la derecha.

```
n = 280
p = 0.5
x = 1:n
comparar_binom_normal(x, n, p, "Con n=280 y p=0.85")
legend("topright", legend = c("binomial", "normal"),
        lwd = 3, col = c("red", "blue"))
```



Haciendo varias prueba pudimos encontrar que con una $n = 280$ y $p = 0.85$ la distribución de la normal y binomial obtuvieron una muy buena aproximación. Concluyendo que es verdad que la binomial converge a la normal cuando n se hace muy grande.