Tarea 1

```r
library (tidyr)
library(ggplot2)
library(ggthemes)
require(gridExtra)
```

```
## Loading required package: gridExtra
```

```r
source("MM1_queue.R")
```

1. Sea X el número de 'unos' obtenido en 12 lanzamientos de un dado honesto. Entonces X tiene una distribución binomial. Calcular una tabla con los valores de la función de distribución para $x = 0, 1, ..., 12$ por dos métodos: usando la función cumsum y usando la función *pbinom*. También determinar cuánto vale $P(X > 7)$.

   - En este caso tenemos que $X \sim Bin\left(12, \frac{1}{6}\right)$, se quiere obtener la distribución de dicha variable aleatoria, comencemos por obtener la función de densidad y a partir de ella obtengamos la distribución:

     ```r
     bin_density = dbinom(0:12, 12, 1/6)
     bin_distribution_1 = cumsum(bin_density)
     distribution <- data.frame(bin_distribution_1, row.names = 0:12)
     knitr::kable(distribution)
     ```

	bin_distribution_1
0	0.1121567
1	0.3813326
2	0.6774262
3	0.8748219
4	0.9636500
5	0.9920750
6	0.9987075
7	0.9998445
8	0.9999866
9	0.9999992
10	1.0000000
11	1.0000000
12	1.0000000

   - Ahora, utilizando el comando de R para obtener la distribución

     ```r
     bin_distribution_2 = pbinom(0:12,12,1/6)
     distribution <- data.frame(bin_distribution_2, row.names = 0:12)
     knitr::kable(distribution)
     ```

	bin_distribution_2
0	0.1121567
1	0.3813326
2	0.6774262
3	0.8748219
4	0.9636500

	bin_distribution_2
5	0.9920750
6	0.9987075
7	0.9998445
8	0.9999866
9	0.9999992
10	1.0000000
11	1.0000000
12	1.0000000

- Finalmente, determinamos $P(X > 7)$:

```
1 - pbinom(7, 12, 1/6)
```

```
## [1] 0.0001555443
```

```
sum(dbinom(8:12, 12, 1/6))
```

```
## [1] 0.0001555443
```

2. (Estaturas de presidentes gringos). En un artículo de Wikipedia, se reportan las estaturas de los Presidentes de los Estados Unidos y los de sus oponentes en elecciones. Se ha notado que mientras más alto sea el presidente típicamente gana la elección.

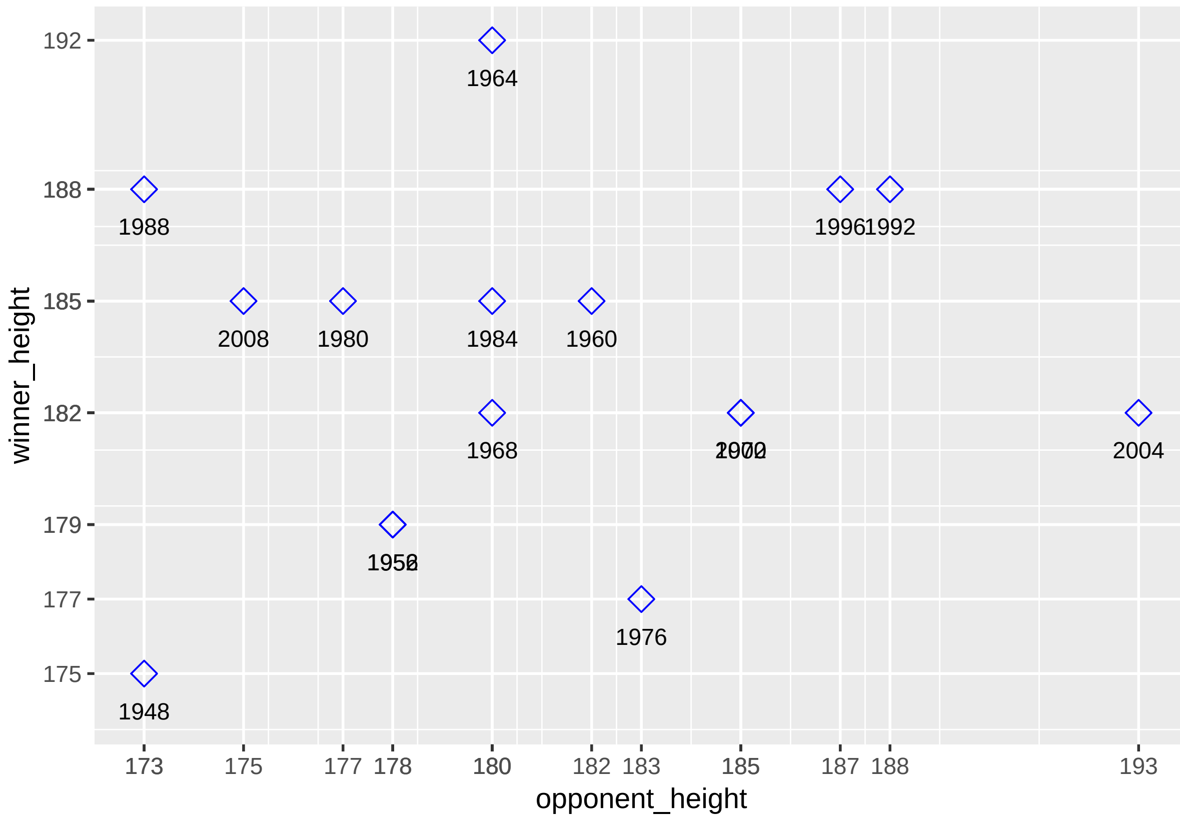
- Comencemos por leer el Dataframe de el archivo csv obtenido de Wikipedia:

```
presidents_heights <- read.csv('presidents_heights.csv')
knitr::kable(presidents_heights)
```

year	winner	winner_height	opponent	opponent_height
2008	Barack Obama	185	John McCain	175
2004	George W. Bush	182	John Kerry	193
2000	George W. Bush	182	Al Gore	185
1996	Bill Clinton	188	Bob Dole	187
1992	Bill Clinton	188	George H.W. Bush	188
1988	George H.W. Bush	188	Michael Dukakis	173
1984	Ronald Reagan	185	Walter Mondale	180
1980	Ronald Reagan	185	Jimmy Carter	177
1976	Jimmy Carter	177	Gerald Ford	183
1972	Richard Nixon	182	George McGovern	185
1968	Richard Nixon	182	Hubert Humphrey	180
1964	Lyndon B. Johnson	192	Barry Goldwater	180
1960	John F. Kennedy	185	Richard Nixon	182
1956	Dwight D. Eisenhower	179	Adlai Stevenson II	178
1952	Dwight D. Eisenhower	179	Adlai Stevenson II	178
1948	Harry S. Truman	175	Thomas Dewey	173

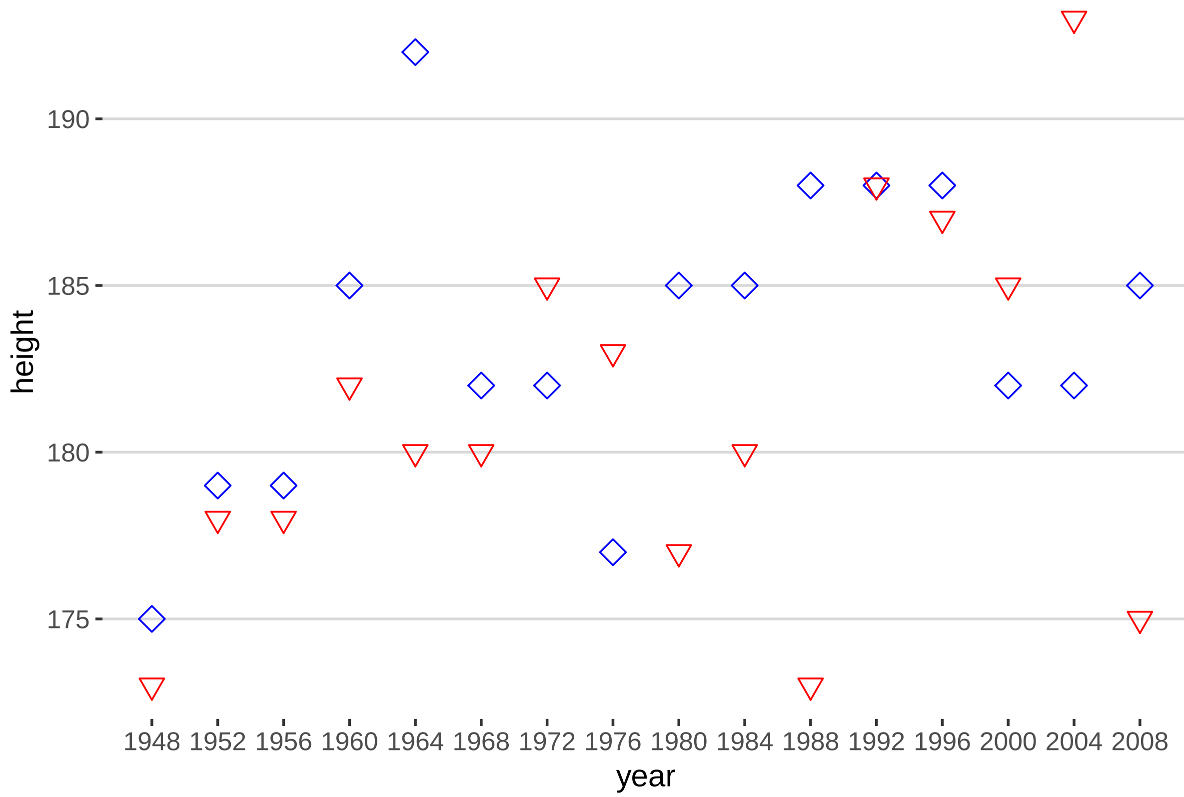
- Y realizamos la gráfica:

```
ggplot(presidents_heights, aes(opponent_height, winner_height)) +
    scale_x_continuous(breaks = presidents_heights$opponent_height) +
    scale_y_continuous(breaks = presidents_heights$winner_height) +
    geom_point(aes(opponent_height, winner_height) , color = "blue", shape = 5, size=3) +
    geom_text(aes(label=year),nudge_y = -1, size=3)
```



+ Y otro tipo de gráfica:

```
ggplot(presidents_heights, aes(year, height)) +
    scale_x_continuous(breaks = presidents_heights$year) +
    geom_point(aes(year, winner_height) , color = "blue", shape = 5, size=3) +
    geom_point(aes(year, opponent_height), color = "red", shape = 6, size=3) +
  theme_hc()
```



3. La función *rpois* genera observaciones aleatorias de una distribución Poisson. Usen la función *rpois* para simular un número grande ($n = 1000$, $n = 10000$) de muestras Poisson con parámetro $\lambda = 0.61$. Encuentren la función de masa de probabilidad, media, y varianza para las muestras. Comparen con los valores teóricos de la densidad Poisson.

- Comencemos por obtener la muestra con $n = 1000$

```
lambda = .61
writeLines(paste('Valores teóricos:\nMedia   : ', lambda, ' \nVarianza :', lambda))
```

```
## Valores teóricos:
## Media   :  0.61
## Varianza : 0.61
```

```
pois = rpois(1000, lambda)
writeLines(paste('Con n = 1000:\nMedia muestral    : ', mean(pois), ' \nVarianza muestral :', v
```

```
## Con n = 1000:
## Media muestral    :  0.603
## Varianza muestral : 0.624015015015015
```

Donde la tabla de densidad de la muestra contra la densidad teórica de la distribución es:

```
count <- data.frame(table(pois))
count$Freq = count$Freq/1000
m = max(pois)
count$teorica = dpois(0:m, .61)
count$error_absoluto = abs(count$teorica - count$Freq)

knitr::kable(count)
```

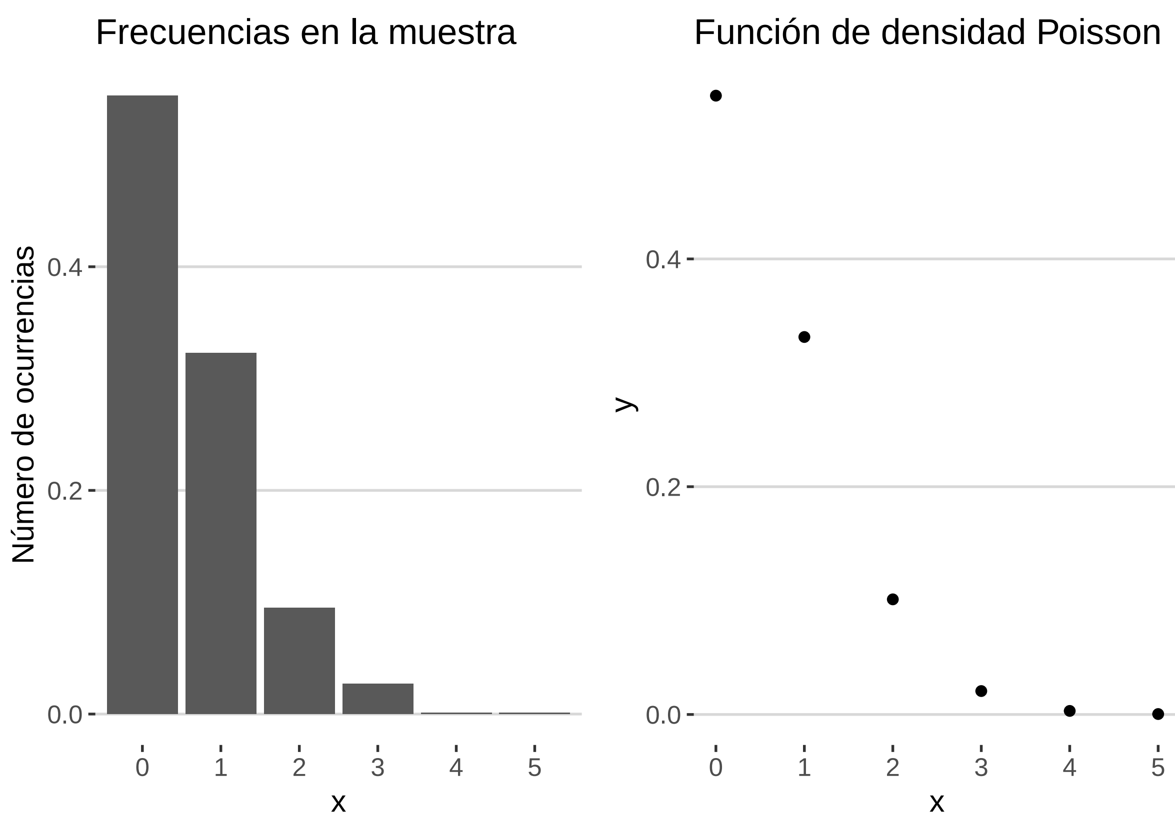
pois	Freq	teorica	error_absoluto
0	0.553	0.5433509	0.0096491
1	0.323	0.3314440	0.0084440
2	0.095	0.1010904	0.0060904
3	0.027	0.0205551	0.0064449
4	0.001	0.0031346	0.0021346
5	0.001	0.0003824	0.0006176

Finalmente, graficamos para tener una comparación visual

```
expetimental = ggplot(count, aes(pois, Freq)) +
  labs(title="Frecuencias en la muestra", y="Número de ocurrencias", x="x") +
  geom_bar(stat="identity") + theme_hc()

teorica = ggplot(data.frame(x = c(0:m) ), aes(x = x)) +
  labs(title="Función de densidad Poisson", y="y", x="x") +
    stat_function( geom="point", n=m+1, fun=dpois, args=list(lambda)) +
    theme_hc()

grid.arrange(expetimental,teorica,ncol=2)
```




```
lambda = .61
writeLines(paste('Valores teóricos:\nMedia   : ', lambda, ' \nVarianza :', lambda))
```

```
## Valores teóricos:
## Media   :  0.61
## Varianza : 0.61
```

```r
pois = rpois(10000, lambda)
writeLines(paste('Con n = 10000:\nMedia muestral    : ', mean(pois), ' \nVarianza muestral :',
```

```
## Con n = 10000:
## Media muestral    :  0.6136
## Varianza muestral : 0.601555195519552
```

Donde la tabla de densidad de la muestra contra la densidad teórica de la distribución es:

```r
count <- data.frame(table(pois))
count$Freq = count$Freq/10000
m = max(pois)
count
```

```
##   pois   Freq
## 1    0 0.5364
## 2    1 0.3411
## 3    2 0.0990
## 4    3 0.0200
## 5    4 0.0032
## 6    5 0.0001
## 7    6 0.0002
```

```r
count$teorica = dpois(0:m, .61)
count$error_absoluto = abs(count$teorica - count$Freq)

knitr::kable(count)
```

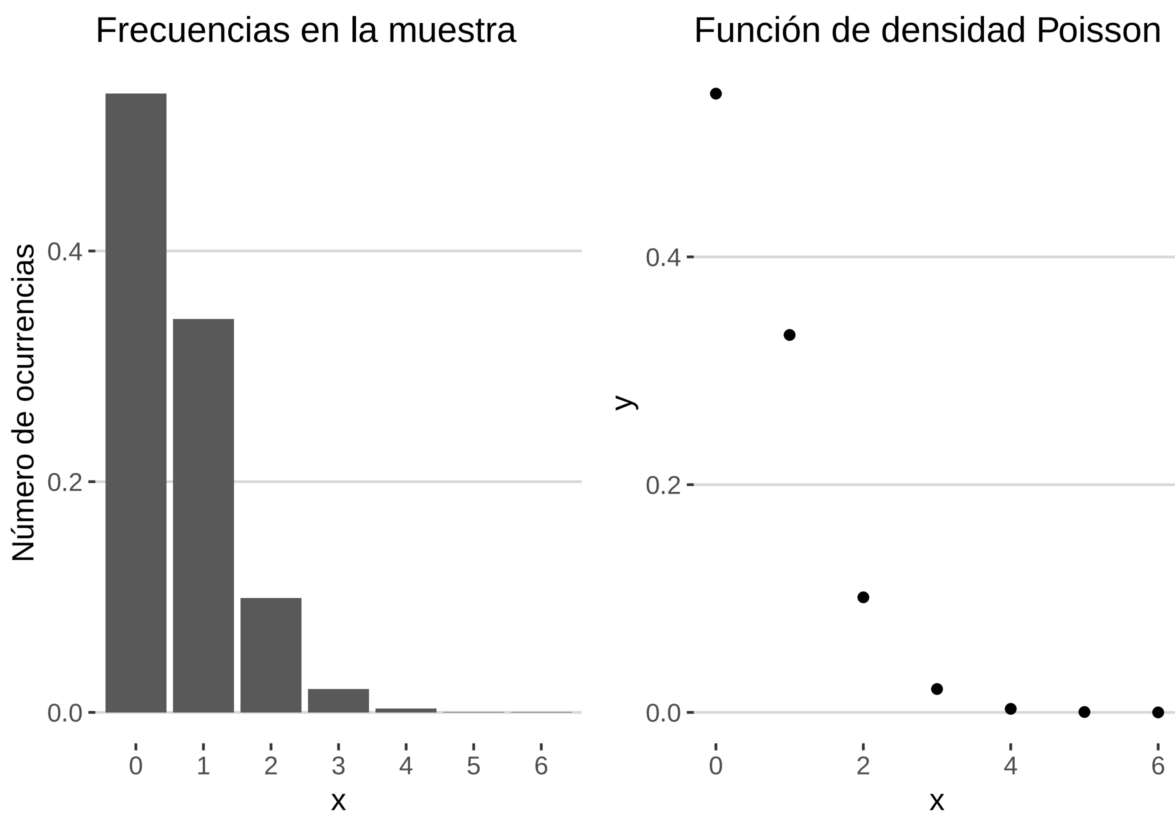
pois	Freq	teorica	error_absoluto
0	0.5364	0.5433509	0.0069509
1	0.3411	0.3314440	0.0096560
2	0.0990	0.1010904	0.0020904
3	0.0200	0.0205551	0.0005551
4	0.0032	0.0031346	0.0000654
5	0.0001	0.0003824	0.0002824
6	0.0002	0.0000389	0.0001611

Finalmente, graficamos para tener una comparación visual

```r
expetimental = ggplot(count, aes(pois, Freq)) +
  labs(title="Frecuencias en la muestra", y="Número de ocurrencias", x="x") +
  geom_bar(stat="identity") + theme_hc()

teorica = ggplot(data.frame(x = c(0:m) ), aes(x = x)) +
  labs(title="Función de densidad Poisson", y="y", x="x") +
    stat_function( geom="point", n=m+1, fun=dpois, args=list(lambda)) +
    theme_hc()

grid.arrange(expetimental,teorica,ncol=2)
```



4. Escriban una función en R llamada *sd.n* que regrese el valor estimado de $\hat{\sigma}$ de una muestra de tamaño n, utilizando la fórmula del estimado máximo verosímil de la varianza.

   - Dado que

$$\hat{\sigma} = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

```
sd.n <- function(X, insesgado = FALSE){

  media = mean(X)
    if (insesgado == TRUE){
      var = sum( (X - media )^2 )/ (length(X)-1)
    }
    else{
      var =  sum( (X - media )^2 )/ (length(X))
    }

    return (var)
}
```

Y comparando dicha función con la de R:

```
var(pois)
```

```
## [1] 0.6015552
```

```
sd.n(pois, insesgado=TRUE)
```

```
## [1] 0.6015552
```

```r
sd.n(pois)
```

```
## [1] 0.601495
```

5. Escriban una función norma que calcule la norma Euclideana de un vector numerico de longitud n. Evaluar la norma de los vectores $(0, 0, 0, 1)$, $(2, 5, 2, 4)$ y $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$

```r
  norm_2 <- function(x){
      sqrt(sum(x^2))
  }

v1 = c(0,0,0,1)
v2 = c(2, 5, 2, 4)
v3 = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

norm_2( v1 )
```

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

```r
norm_2( v2 )
```

```
## [1] 7
```

```r
norm_2( v3 )
```

```
## [1] 19.62142
```

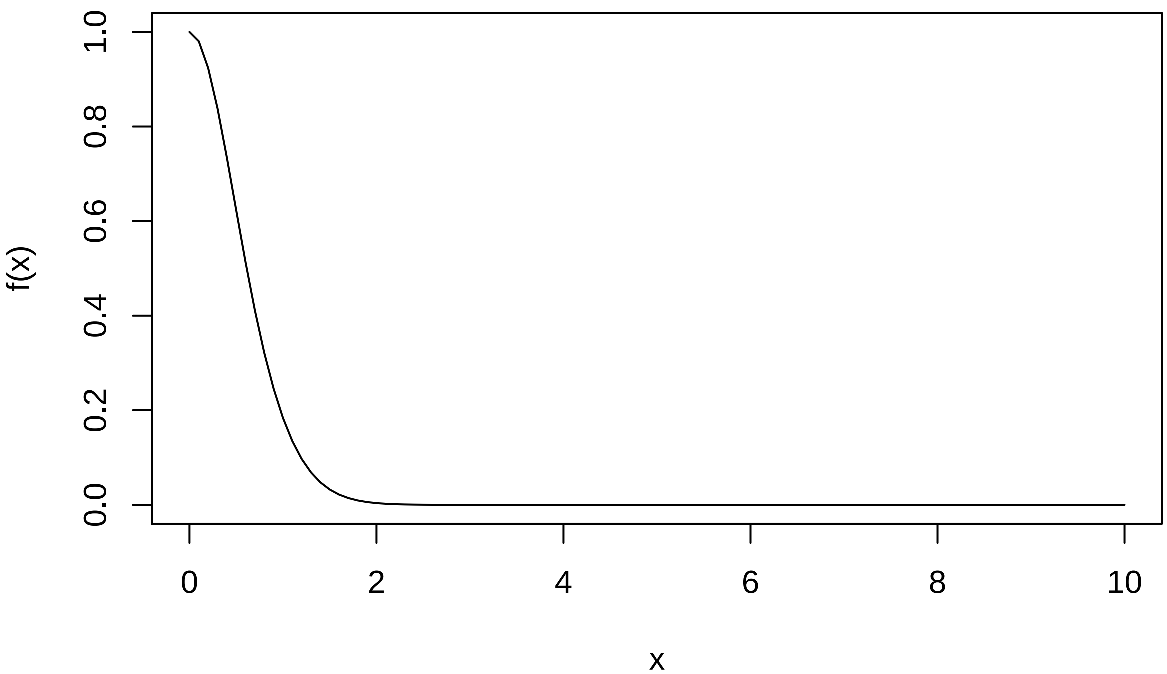
6. Usar la función curve para graficar la función $f(x) = \frac{e^{-x^2}}{1+x^2}$ en el intervalo $0 \leq x \leq 10$. Luego usar la función integrate para calcular el valor de la integral:

$$\int_0^\infty \frac{e^{-x^2}}{1+x^2}$$

El límite superior se especifica usando el argumento upper=Inf en la función integrate.

```r
f <- function(x){
    exp(-(x^2))/(1 + x^2)
}

curve(f, from = 0, to = 10)
```




```
integrate(f, lower = 0, upper = Inf)
```

```
## 0.6716467 with absolute error < 8.3e-05
```

7. Construir una matriz con 10 renglones y 2 columnas que contienen datos provenientes de una normal estándar: $x < -$ matrix(rnorm(20),10,2) Esta es una muestra de 10 observaciones de una distribución normal bivariada. Usen la función apply y la función norma que crearon en un ejercicio anterior para calcular las normas euclideanas para cada una de las 10 observaciones.

```
x <- matrix(rnorm(20),10,2)
x
```

```
##              [,1]        [,2]
##  [1,] -0.09913813 -0.44375200
##  [2,]  1.34742503 -0.95506147
##  [3,]  0.53841952 -0.96241010
##  [4,]  0.71762816 -0.09480885
##  [5,]  1.95564575 -1.36114360
##  [6,]  1.25814274 -0.72308197
##  [7,]  0.07777546 -0.78866919
##  [8,] -1.02006111  2.96638149
##  [9,] -1.61064496 -2.17435747
## [10,] -1.26323484  0.31610020
```

```
apply(x, 1, norm_2)
```

```
##  [1] 0.4546913 1.6515740 1.1027823 0.7238639 2.3827006 1.4511274 0.7924949
##  [8] 3.1368685 2.7059208 1.3021834
```

8. Los siguientes datos describen el factor de desgaste de papel manufacturado bajo diferentes presiones durante el prensado. Cuatro hojas de papel fueron seleccionadas y probadas para cada uno de los cinco lotes manufacturados, Hacer un boxplot para comparar los diferentes factores de resistencia para cada presión.

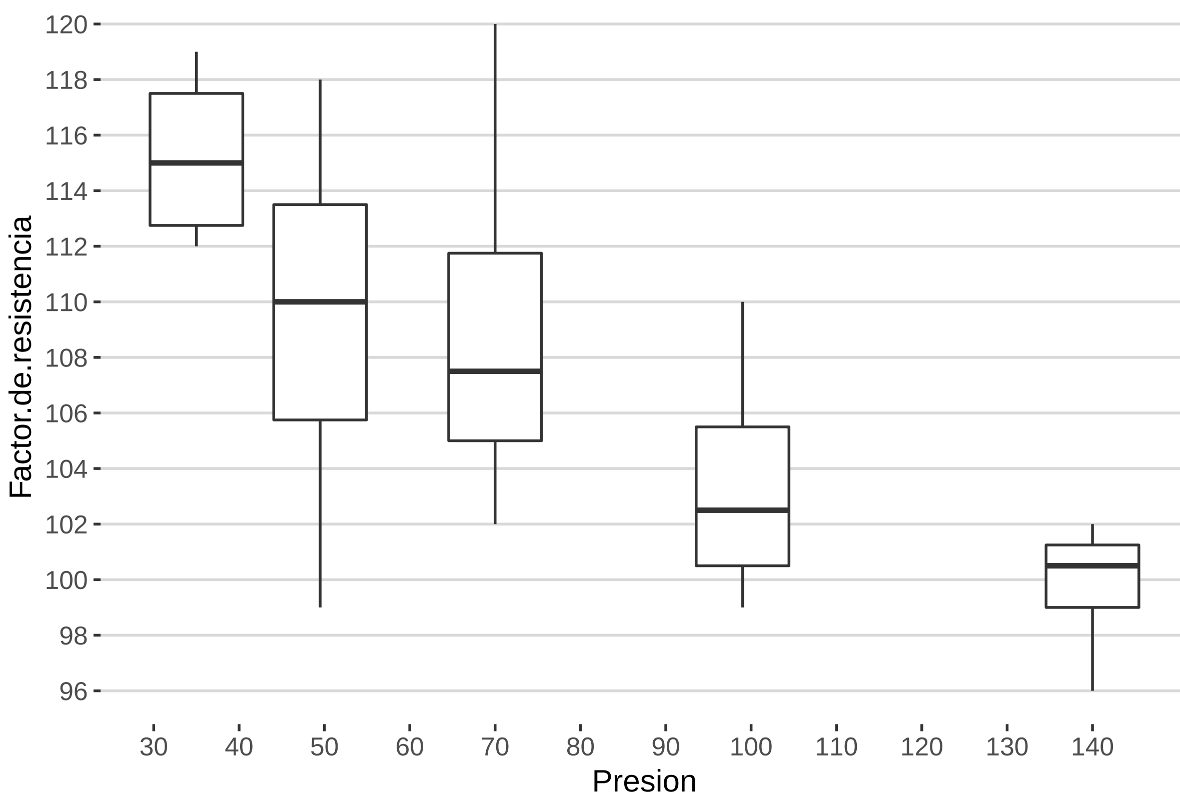
```
  a = c(35.0,35.0,35.0,35.0,49.5,49.5,49.5,49.5,70.0,70.0,70.0,
        70.0,99.0,99.0,99.0,99.0,140.0,140.0,140.0,140.0)
```

```r
b = c(112,119,117,113,108,99,112,118,120,106,102,109,110,101,99,104,100,102,96,101)
desgaste <- data.frame("Presion" = a, "Factor de resistencia" = b)
knitr::kable(head(desgaste))
```

Presion	Factor.de.resistencia
35.0	112
35.0	119
35.0	117
35.0	113
49.5	108
49.5	99

Y creamos la boxplot:

```r
ggplot(desgaste, aes(group = Presion, x = Presion, y = Factor.de.resistencia)) +
      geom_boxplot() +
      scale_x_continuous(breaks = pretty(desgaste$Presion, n = 10)) +
      scale_y_continuous(breaks = pretty(desgaste$Factor.de.resistencia, n = 10)) +
      theme_hc()
```



9. Relacionado con el modelo de línea de espera. Incorporar las variables:
   - El tiempo total promedio de los n clientes en el sistema ('promedio_tiempo_en_sistema')
   - La longitud máxima de la cola ('long_max_cola')
   - La máxima espera en cola ('max_espera')

   Y ejecutar el modelo 100 veces con $\lambda A = 5$, $\lambda S = 4$ y $n = 1000$ y hacer un histograma para cada una de las estadísticas de desempeño, y calcular estadísticas descriptivas (media, varianza y coeficiente de variación, min, max, etc) para cada una de ellas.

Obteniendo los resultados de las medidas descriptivas para las 100 simulaciones:

```r
df <- data.frame(0,0,0,0,0,0)
names(df)<-c("Prom_espera","Utilizacion","Prom_clientes",
             "Promedio_en_sistema", 'Long_max', 'Max_espera')

for (i in 1:1000){
  ans = mm1(lambdaA = 5, lambdaS = 4, n = 1000)
  df[nrow(df) + 1,] = c(ans[1:6])
}
df <- data.frame(df[-1, ])
row.names(df) <- NULL
knitr::kable(head(df))
```

Prom_espera	Utilizacion	Prom_clientes	Promedio_en_sistema	Long_max	Max_espera
13.49569	79.20069	2.620515	17.57453	18	70.35655
28.57668	83.04183	5.832776	32.64517	30	120.99195
13.26427	77.59852	2.705189	17.06913	20	74.75628
20.03419	83.49445	4.162040	24.05323	24	92.22748
13.50559	78.62030	2.688913	17.45444	15	65.57620
11.14837	78.52725	2.169740	15.18320	15	83.64582

Mostrando estadísticas descriptivas

```r
summary(df)
```

```
##   Prom_espera      Utilizacion    Prom_clientes    Promedio_en_sistema
##  Min.   : 7.114   Min.   :70.31   Min.   : 1.341   Min.   :10.75
##  1st Qu.:11.977   1st Qu.:77.22   1st Qu.: 2.355   1st Qu.:15.92
##  Median :14.224   Median :79.56   Median : 2.842   Median :18.23
##  Mean   :15.431   Mean   :79.65   Mean   : 3.090   Mean   :19.43
##  3rd Qu.:17.818   3rd Qu.:82.01   3rd Qu.: 3.603   3rd Qu.:21.91
##  Max.   :63.412   Max.   :90.82   Max.   :13.232   Max.   :67.63
##     Long_max       Max_espera
##  Min.   : 9.00   Min.   : 37.22
##  1st Qu.:15.00   1st Qu.: 65.99
##  Median :18.00   Median : 77.88
##  Mean   :18.68   Mean   : 82.20
##  3rd Qu.:22.00   3rd Qu.: 95.29
##  Max.   :45.00   Max.   :213.20
```

Y finalmente, graficando:

```r
Promedio_espera_hist    = ggplot(df, aes(Prom_espera)) +
                        geom_histogram()
Utilizacion_hist        = ggplot(df, aes(Utilizacion)) +
                          geom_histogram()
Promedio_clientes_hist  = ggplot(df, aes(Prom_clientes)) +
                          geom_histogram()
tiempo_en_sistema_hist  = ggplot(df, aes(Promedio_en_sistema)) +
                          geom_histogram()
Long_max_cola_hist      = ggplot(df, aes(Long_max)) +
```

```
                            geom_histogram()
Max_espera_en_cola_hist = ggplot(df, aes(Max_espera)) +
                            geom_histogram()

grid.arrange(Promedio_espera_hist, Utilizacion_hist,
             Promedio_clientes_hist, tiempo_en_sistema_hist,
             Long_max_cola_hist, Max_espera_en_cola_hist,
             ncol=3)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```