Matrices aleatorias (Tarea 2)

J. Antonio García Ramirez

20 de semtiembre 2018

1. Generar $M = 10000$ matrices para las dimensiones $N \in \{10, 100\}$ y producir los histogramas normalizados de la muestra completa de $M \times N$ valores propios. Para obtener los histogramas normalizados, se deben escalar por el factor $1/\sqrt{\beta N}$, donde $\beta = 1, 2, 4$, para el caso GOE, GUE, y GSE; respectivamente. Compare con el resultado teórico $\rho(x) = \frac{1}{\pi}\sqrt{2 - x^2}$, *lamada ley del semicirculo de Wigner.*

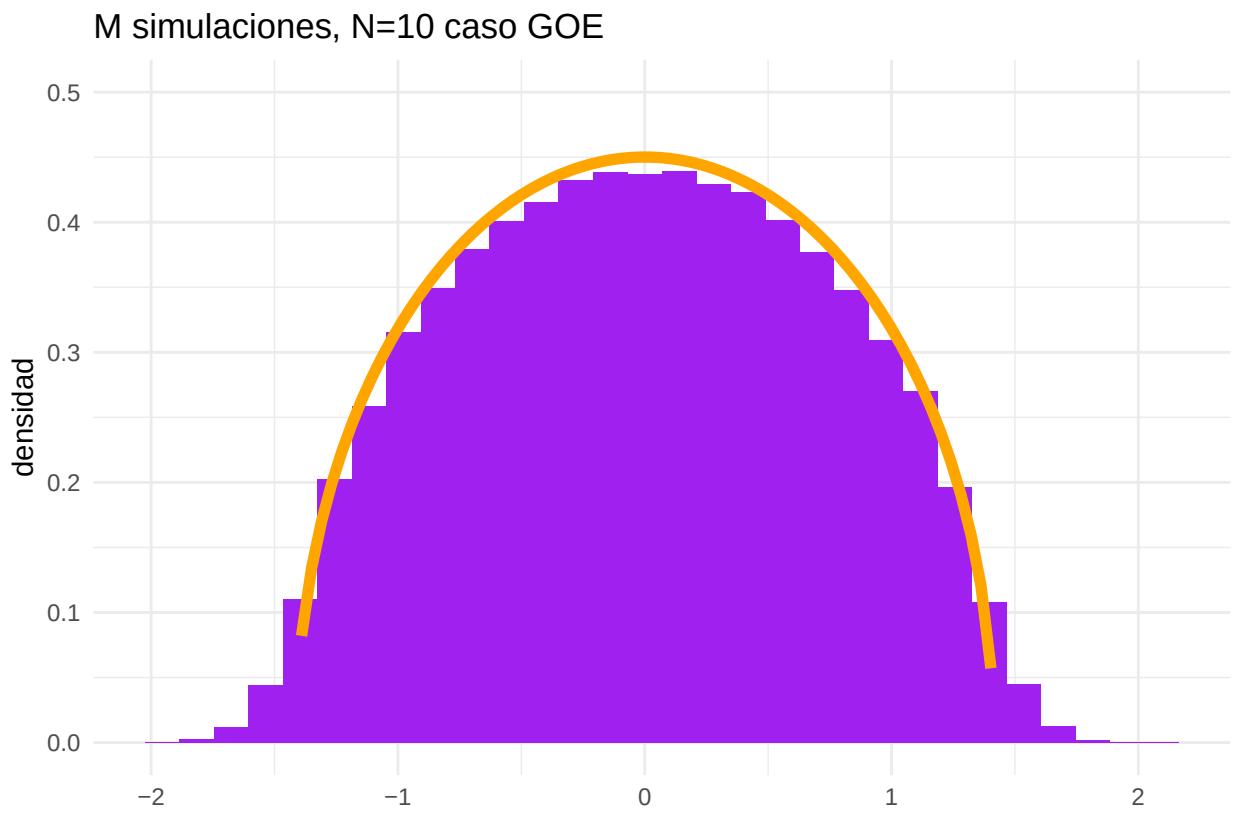
Con la siguiente Closure implemente todas las simulaciones:

```
Wigner.semi.circulo.init <- function(N, caso)
{
  # Entradas
  ## N (int) dimension de la matriz
  #Regreza uuna funcion para simular obetener los vect. propis de una matriz
  caso <- caso
  function(N)
  {
    if(caso=='GOE')
    {
      matriz <- matrix(rnorm(N**2), ncol = N)
      matriz <- matriz * (1/((1*N)**(.5))) #correcion
      matriz <- (matriz + t(matriz)) / 2
      val.pro <- eigen(matriz)$values
    }
    if(caso=='GUE')
    {
      entradas <- complex(real=rnorm(N**2),
                          imaginary = rnorm(N**2))
      matriz <- matrix(entradas, ncol = N) * (1/((2*N)**(.5))) #correcion
      matriz <- (matriz + t(Conj(matriz)) )/2
      val.pro <- eigen(matriz)$values
    }
    if( caso=='GSE')
    {
      A <- complex(real=rnorm(N**2), imaginary = rnorm(N**2))
      A <- matrix(A, ncol = N) * (1/((2*N)**(.5))) #correcion
      B <- complex(real=rnorm(N**2), imaginary = rnorm(N**2))
      B <- matrix(B, ncol = N) * (1/((2*N)**(.5))) #correcion
      M1 <- cbind(A,B)
      M2 <- cbind(-t(Conj(B)), t(Conj(A)))
      matriz <- rbind(M1, M2)
      matriz <- (matriz + t(Conj(matriz)))/2
      val.pro <- eigen(matriz)$values
    }
    return(val.pro)
  }
}
```

Primero vamos por el caso GOE. En las siguientes gráficas se muestran los resultados de las simulaciones y se contrasta con la función que define el semicírculo (en color naranja)

```
ggplot(s, aes(x = GOE.x)) +
  geom_histogram(aes(y=..density..,fill=I('purple')),
                 bins = 30)+
  theme_minimal()  + ylab('densidad') + xlab('') +
  ggtitle('M simulaciones, N=10 caso GOE') +
  stat_function(fun=function(x) {(1/pi)*((2-x**2)**.5)},
                colour='orange', size =2) + ylim(c(0,.5))
```

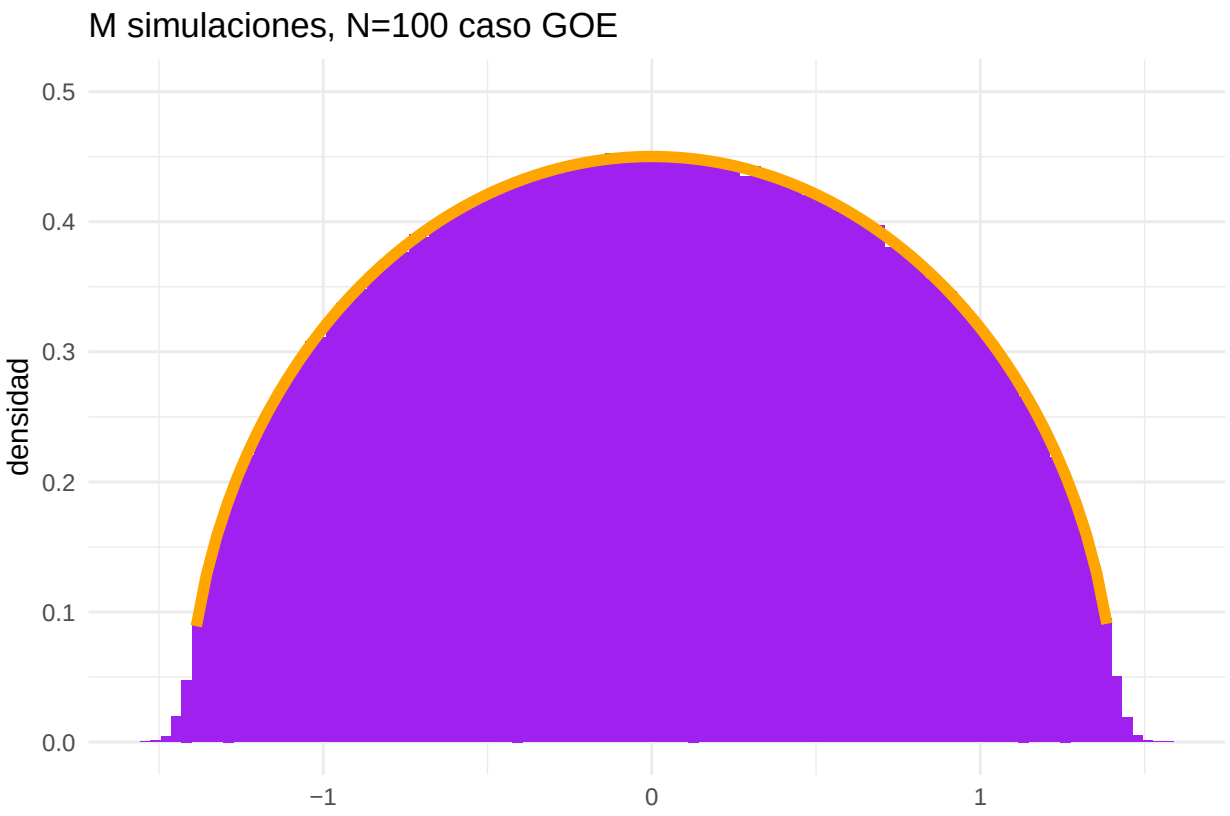
```
## Warning: Removed 31 rows containing missing values (geom_path).
```




```
#caso de dimension 100
```

```
ggplot(s, aes(x = GOE.x)) +
  geom_histogram(aes(y=..density..,fill=I('purple')),
                 bins = 100)+
  theme_minimal()  + ylab('densidad') + xlab('') +
  ggtitle('M simulaciones, N=100 caso GOE') +
  stat_function(fun=function(x) (1/pi)*((2-x**2)**.5), colour='orange',size =2)+
  ylim(0,.5)
```

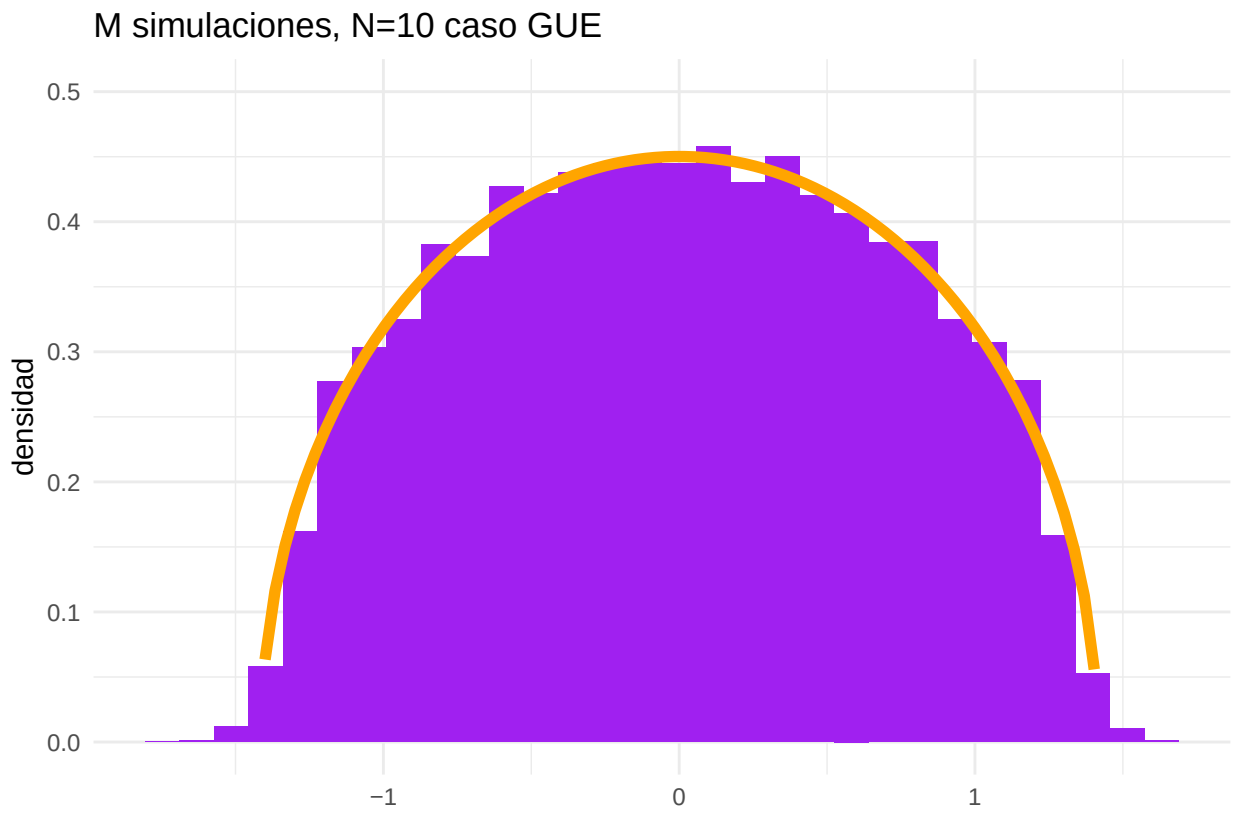
```
## Warning: Removed 11 rows containing missing values (geom_path).
```



Continuamos con el caso GUE. En las siguientes gráficas se muestran los resultados de las simulaciones y se contrasta con la función que define el semicírculo (en color naranja)

```
ggplot(s, aes(x = GUE.x)) +
  geom_histogram(aes(y=..density..,fill=I('purple')),
                 bins = 30)+
  theme_minimal()  + ylab('densidad') + xlab('') +
  ggtitle('M simulaciones, N=10 caso GUE') +
  stat_function(fun=function(x) (1/pi)*((2-x**2)**.5), colour='orange', size = 2)+
  ylim(c(0,.5))
```

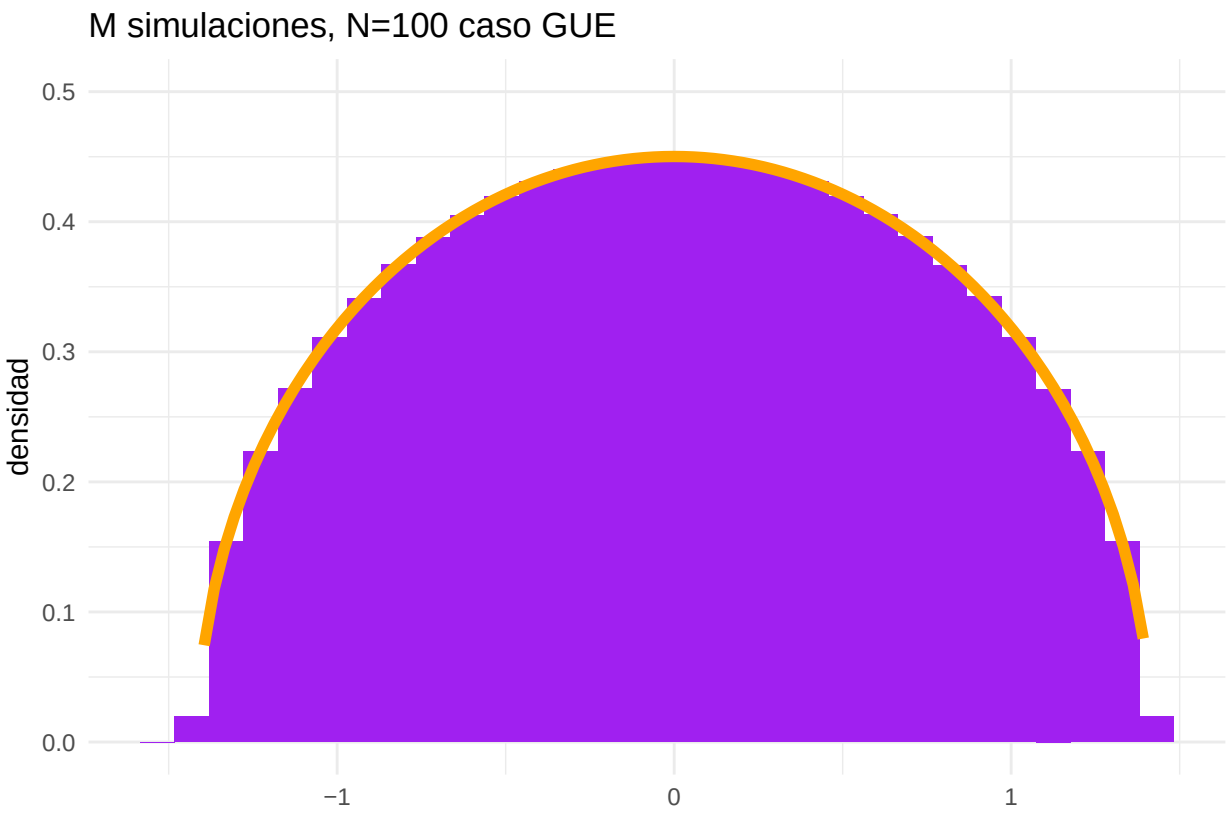
```
## Warning: Removed 17 rows containing missing values (geom_path).
```




```
#caso de dimension 100
```

```
ggplot(s, aes(x = GUE.x)) +
  geom_histogram(aes(y=..density..,fill=I('purple')),
                 bins = 30)+
  theme_minimal()  + ylab('densidad') + xlab('') +
  ggtitle('M simulaciones, N=100 caso GUE') +
  stat_function(fun=function(x) (1/pi)*((2-x**2)**.5), colour='orange', size =2)+
  ylim(c(0,.5))
```

```
## Warning: Removed 6 rows containing missing values (geom_path).
```

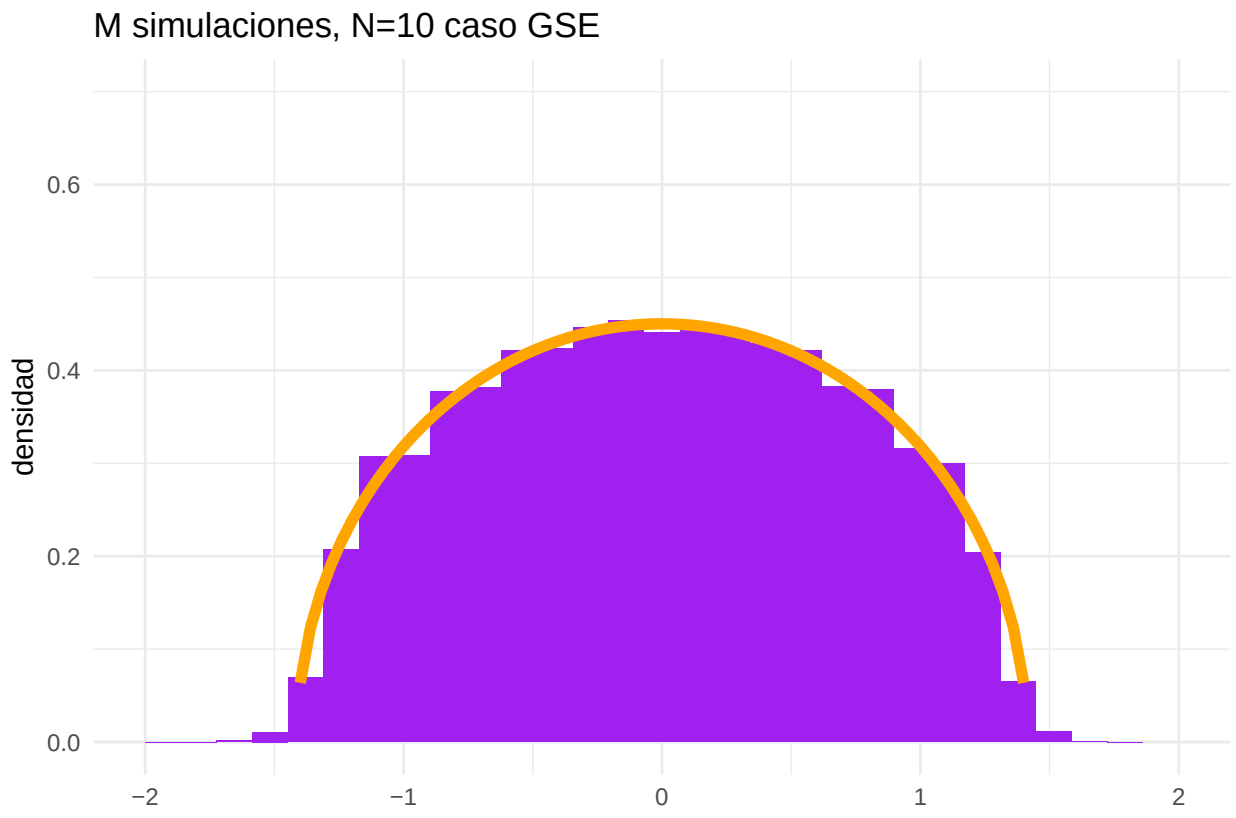


Terminamos con el caso GSE. En las siguientes gráficas se muestran los resultados de las simulaciones y se contrasta con la función que define el semicírculo (en color naranja):

```
ggplot(s, aes(x = GSE.x)) +
  geom_histogram(aes(y=..density..,fill=I('purple')), bins = 30)+
  theme_minimal()  + ylab('densidad') + xlab('') +
  ggtitle('M simulaciones, N=10 caso GSE') +
  stat_function(fun=function(x) (1/pi)*((2-x**2)**.5), colour='orange', size = 2)+ ylim(c(0,.7)) +xlim(
```

```
## Warning: Removed 1 rows containing missing values (geom_bar).
```

```
## Warning: Removed 30 rows containing missing values (geom_path).
```

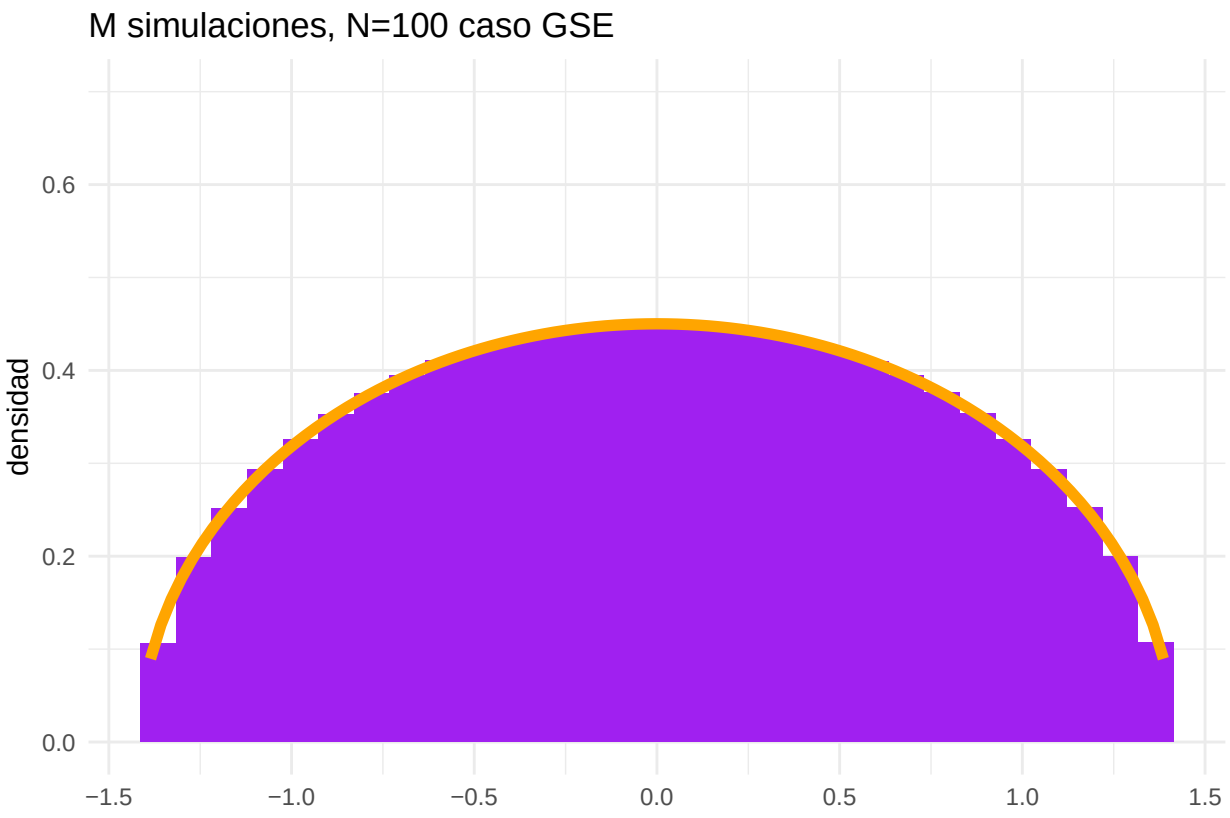



```
#caso de dimension 100
```

```
ggplot(s, aes(x = GSE.x)) +
  geom_histogram(aes(y=..density..,fill=I('purple')),
                 bins = 30)+
  theme_minimal()  + ylab('densidad') + xlab('') +
  ggtitle('M simulaciones, N=100 caso GSE') +
  stat_function(fun=function(x) (1/pi)*((2-x**2)**.5), colour='orange', size=2) +ylim(c(0,0.7)) + xlim(
```

```
## Warning: Removed 1202 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```

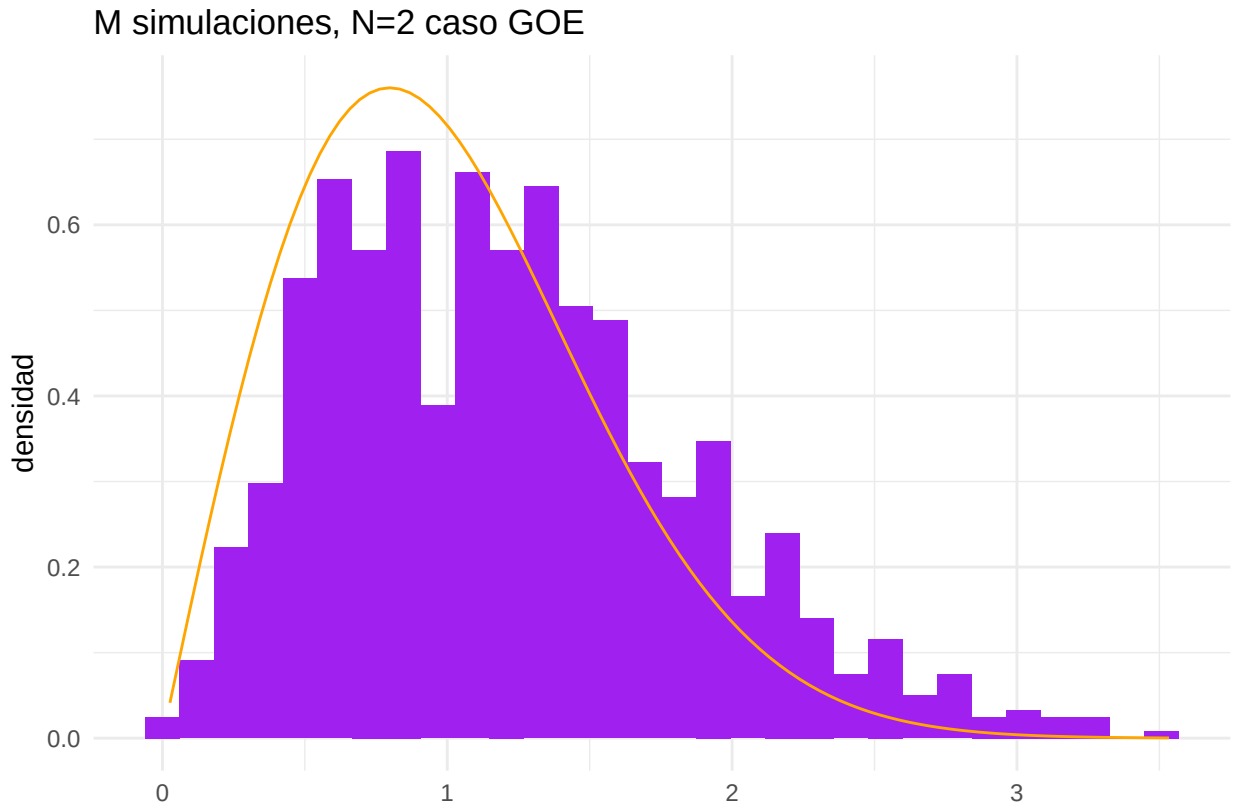


2. Graficar la distribución de espaciamiento contiguos (ordenando los eigenvalores) para un ensemble de $M = 1000$ matrices simétricas de dimensión 100×100 ($H_{s100\times100}$). Sobreponer a la simulación el resultado de la conjetura de Wigner para una matriz del GOE de 2×2 (obtenido en clase): $\bar{P}(s) = \frac{\pi s}{2} e^{-\pi s^2/4}$ ¿Existe buena concordancia ómo la medira? Mi implementación fue la siguiente, primero la probamos para el caso visto en clase $N = 2$:

```
Wigner.semi.circulo.init2 <- function(N, caso)
{
  # Entradas
  ## N (int) dimension de la matriz
  #Regreza uuna funcion para simular obetener los vect. propis de una matriz
  caso <- caso
  function(N)
  {
    if(caso=='GOE')
    {
      matriz <- matrix(rnorm(N**2), ncol = N)*(1/((1*N)**(.5)))
      matriz <- (matriz + t(matriz)) / 2
      val.pro <- eigen(matriz)$values
    }
    return(abs(diff(val.pro)))
  }
}
set.seed(0)
N <- 2
M <- 1000
```

```
GOE <- Wigner.semi.circulo.init2(caso = 'GOE')
muestra.GOE <- mapply(FUN = GOE, rep(N, M) )
dim(muestra.GOE) <- c(1, 1000)
s <- data.frame(GOE.x = t(muestra.GOE))
ggplot(s, aes(x = GOE.x)) +
  geom_histogram(aes(y=..density..,fill=I('purple')))+
  stat_function(fun=function(x) {(pi*x/2)*exp(-pi*x**2/4)}, colour=I('orange'))+
  theme_minimal()  + ylab('densidad') + xlab('') +
  ggtitle('M simulaciones, N=2 caso GOE')
```

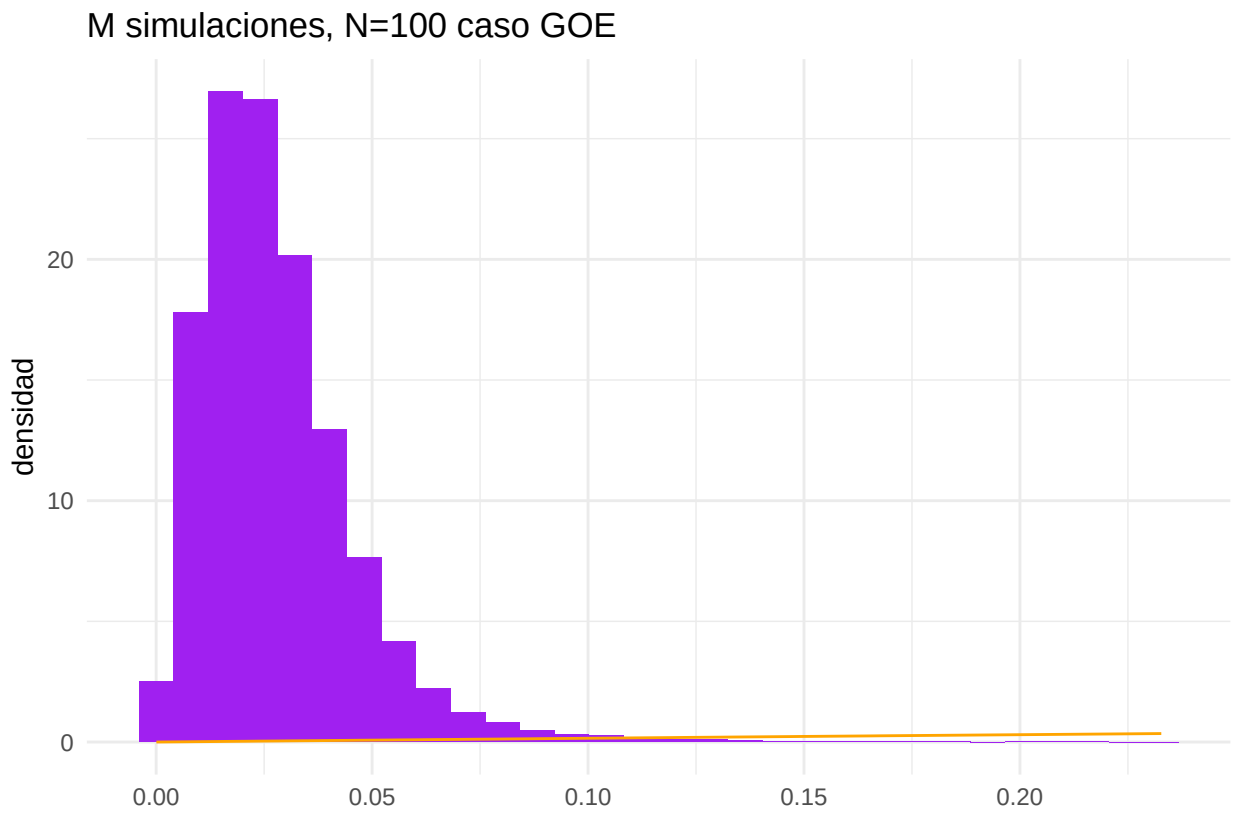
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Y finalmente observamos el caso requerido $N = 100$:

```
set.seed(0)
N <- 100
M <- 1000
muestra.GOE <- mapply(FUN = GOE, rep(N, M) )
dim(muestra.GOE) <- c(1, 99*1000)
s <- data.frame(GOE.x = t(muestra.GOE))
ggplot(s, aes(x = GOE.x)) +
  geom_histogram(aes(y=..density..,fill=I('purple')))+
  stat_function(fun=function(x) {(pi*x/2)*exp(-pi*x**2/4)}, colour=I('orange'))+
  theme_minimal()  + ylab('densidad') + xlab('') +
  ggtitle('M simulaciones, N=100 caso GOE')
```

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



En resumen, podemos apreciar que el ajuste es apropiado (en el primer caso), lo cual respalda la conjetura, a diferencia de lo que comente en clase la distribución conjunta se aleja de una gamma, y más aun de la exponencial que facilitaría los cálculos en distribuciones y en casos de dimesión mayor, el detalle es que esta distribución se refiere a los valores extremos.

Yo mediría el desempeño con la distribución muestral de la variable y el resultado teórico o lo que es lo mismo discretizar la medida L_2 entre la diferencia de las dos distribuciones o de manera análoga utilizar el test de Kolmogórov-Smirnov. Por cuestiones de tiempo no incluyo la implementación.

3. **Estadística en dimensiones altas a través de Matrices Aleatorias**

El articulo *High dimensional statistical inference and random matrices. Johnstone (2006)* comienza con un ejemplo que me es familiar el método de PCA.

Si bien en la práctica un problema de interés al usar PCA es saber el número de componentes adecuados, o digamos el número de valores propios de la matriz de covarianza que son lo suficientemente grandes para ser considerados. En particular este tema es de mi interés pues el semestre anterior en la clase de multivariado hicimos uso constante de las descomposiciones espectrales de diversas matrices, ese es un tema que llama mucho mi interés y hasta el día de hoy para determinar el número de factores de PCA que se deben retener prefiero el método conocido como *Parallel Analysis* donde lo importante es simular apropiadamente a las variables (y por ende a sus matrices de covarianza) para determinar dicho número.

El semestre pasado al realizar un *Parallel Analysis*, como parte de una tarea, y al leer el artículo donde presentan dicha técnica en encontré explícitamente con el problema de encontrar la distribución de los valores propios de una matriz.

Un tema delicado en PCA es el número de observaciones de las que se dispone pues aunque el artículo habla de ejemplos de $p = 10$ variables y $n = 10$ observaciones (en teoría esto no es suficiente pues se requiere estimar $p(p-1)$ parámetros por lo que se requieren de más observaciones. En el articulo tambien hacen constante referencia al supuesto de que las entradas de la matriz tienen una distribución Gaussiana. Tiempo atrás esta suposición me parecía sumamente restrictiva y de hecho el artículo lo toma como supuesto para hacer PCA (siendo que no se requiere de este supuesto y en la práctica es menos usado) y lo cita en la pág. 5. También en la quinta página hablando de este supuesto citan la famosas frase de Cox "All models are wrong, some are useful", hecho curioso pues como vimos en la clase pasada aún con el supuesto de normalidad la distribución de los valores propios para una matriz cuadrada de dimensiones 2×2 los cálculos se complican y más aún si extendemos el número de dimensiones.

Otro dato de bastante interés se da en la pág. 6 donde se menciona que la distribución conjunta de los valores propios en los casos de PCA y análisis de correlación canónica se conoce desde 1939 y que fue descubierta por cinco estadísticos diferentes. Es hasta la pág. 7 donde se habla de Wigner y su trabajo, quien estando interesado en los niveles de energía noto que la versión discretizada de un operador Hermitiano se puede ver como una matriz cuadrada. Aunque el artículo no habla explícitamente de la conjetura de Wigner sí hace mención de la ley del semicírculo con el cual trabajamos en el primer ejercicio de esta tarea.

En la sección 2.4 se habla acerca del idealizado concepto de la estadística clásica el cual consiste en dejar fijo el número de variables p y hacer que n tienda a infinito, y hablan sobre el problema actual del análisis de datos cuando el número de variables se incrementa. La idea anterior es el enlace entre la estadística y las matrices aleatorias.

A finales de la sección 2 y toda la tercera se dan resultados acerca de los valores extremos de los valores propios lo cual me parece natural pues en varias aplicaciones no se requiere de conocer todos los valores propios de una matriz sino solo los más pequeños o los más grandes (como en los algoritmos de particionamiento de gráficas o el algoritmo de Lanczos dos temas que son de importancia en la segmentación de imágenes o incluso en encontrar cluster de individuos u observaciones).Justo en la pág. 11 el artículo tiene como nota al pie una url donde prometen código en S para comprobar los valores de una distribución pero de momento ese link no sirve.

En la tercera sección se dan resultados teóricos acerca de la distribución del valor propio más grande de una matriz para diversos casos, en ellos podemos ver que tanto el número de variables como el número de observaciones juegan un papel importante en la construcción de estas distribuciones límite.

Para finalizar me gustaría recalcar el ejemplo al final de la cuarta sección donde se habla de que Stephen J. Brown utilizó un enfoque de análisis de factores para una collecion de precios en stock, aparte de su descubrimiento esto se entrelaza directamente con lo último que vimos en la clase de series de tiempo y el concepto de cointegración.

El articulo me parecio una lectura agradable y muy ilustrativa sobre PCA y como las matrices aleatorias pueden y están ayudando a reformularlo. Esto seguramente me será de utilidad en mi proyecto final de

cómputo estadístico.