semicircledistr: An R Package to Simulate the Semicircle Distribution

Dylan Perrenoud, Marcus Quincy, & Matthew White

Utah State University

April 18, 2025

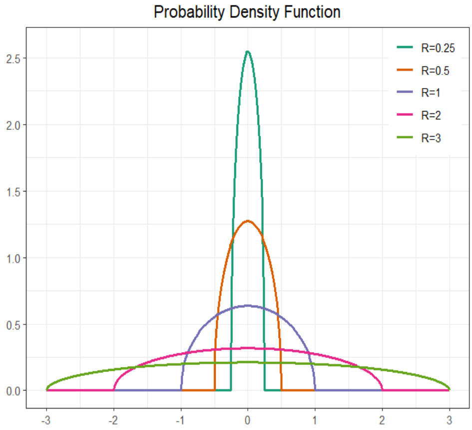
Outline

Project Motivation

- There is currently no existing R package to simulate the semicircle distribution

Wigner Semicircle Distribution

- Proposed by physicist Eugene Wigner
- Called the “semicircle" distribution because its probability density function forms a semicircle shape
- Its main two parameters are a radius value and offset value



Wigner Semicircle Distribution

The probability density function is given as:

$$f(x) = \frac{2}{\pi R^2}\sqrt{R^2 - (x-a)^2}$$

for $-R \leq x - a \leq R$, and

$$f(x) = 0$$

if $|x - a| > R$.

Wigner Semicircle Distribution - Motivation

The distribution of the eigenvalues of an N x N i.i.d. random matrix will converge to the semicircle distribution as N goes to infinity.

Random Matrices

- A random matrix is a matrix in which some or all of its entries are sampled randomly from a probability distribution
- An independent and identically distributed (i.i.d.) random matrix is a matrix in which each of its entries is drawn from the same probability distribution
- It is important that the matrix is square (N x N)

Random Matrices - 5 x 5 Example

```
random_matrix_example <- matrix(data = rnorm(25), 5, 5)
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.54735940	1.82725439	-1.5285173	-0.3118123	-1.6712724
[2,]	-1.43739012	1.44368867	0.4260014	-1.2588068	-0.4185352
[3,]	0.53677831	0.08067672	0.7296175	-0.5003984	0.6294235
[4,]	0.56573756	0.32864457	-2.2356619	0.6693546	0.9341677
[5,]	0.06313499	2.11090828	0.1252048	-1.2673569	-0.5067101

Eigenvalues

- Given:
  - An n x n matrix A
  - A non-zero vector v of length n
- If Av scales v by a factor of λ, where λ is a scalar, then v is an eigenvector of A and λ is the corresponding eigenvalue
- The matrix A has at most n eigenvalues

The R Package: semicircledistr

README License MIT license

semicircledistr

R-CMD-check.yaml passing

The goal of semicircledistr is to simulate the Wigner semicircle distribution. This package includes the four main distribution simulation functions that other distributions in R have.

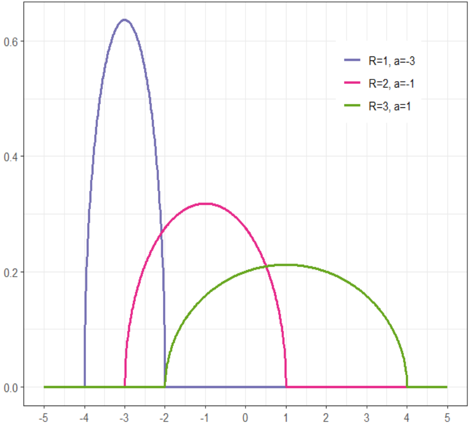
Installation

You can install the development version of semicircledistr like so:

```
devtools::install_github("https://github.com/matthewwhite1/semicircledistr")
```

Semicircle Probability Density Function (PDF)

- A PDF is a function whose value is the relative probability that the value of a random variable is equal to that sample
- Implemented as the dsemicircle function
- Allows for an offset parameter, while being centered at 0 is the default



Semicircle PDF Implementation

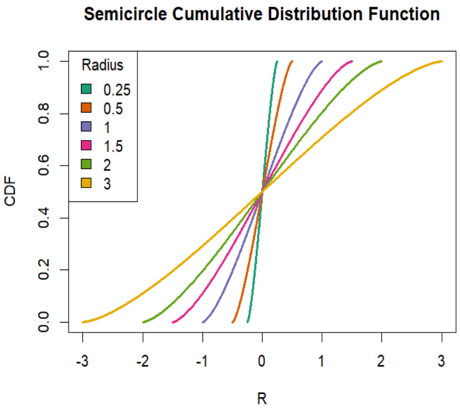
- Function signature: dsemicirle(x, R, a = 0)
  - **x** is the value to get the PDF of
  - **R** is the radius
  - **a** is an optional shift parameter
  - Accepts vectorized input
- Performs validation on inputs
  - Such as making sure values are numeric
- Uses the closed form of the PDF to produce results
  - In the range of $[-R+a, R+a]$:
    $$f(x) = \frac{2}{\pi R^2}\sqrt{R^2-(x-a)^2}$$
  - Otherwise, the probability is 0
  - Closed form obtained from Wikipedia

dsemicircle Code

```r
dsemicircle <- function(x, R, a = 0) {
  if (!is.numeric(x)) {
    stop("x must be numeric.")
  }
  if (R <= 0) {
    stop("R must be positive.")
  }
  if (!is.numeric(a)) {
    stop("a must be numeric.")
  }
  ifelse(abs(x - a) > R, 0, (2 / (pi * R^2)) * sqrt(R^2 - (x - a)^2))
}
```

Semicircle Cumulative Distribution Function (CDF)

- The CDF describes the probability that a value is less than or equal to x in the semicircle distribution
- Implemented in the psemicircle function



Semicircle CDF Implementation (psemicircle)

- Function signature: psemicirle(x, R, a = 0)
  - **x** is the value to get the CDF of
  - **R** is the radius
  - **a** is an optional shift parameter
  - Accepts vectorized input
- Performs validation on inputs
  - Such as making sure values are numeric and x is within the bounds of the radius
- Uses the closed form of the CDF to produce results
  - In the range of $[-R+a, R+a]$:

$$\frac{1}{2} + \frac{(x-a)\sqrt{R^2-(x-a)^2}}{\pi R^2} + \frac{\sin^{-1}(\frac{x-a}{R})}{\pi}$$

  - Closed form obtained from Wikipedia

psemicircle Code

```
psemicircle <- function(x, R, a = 0) {
  if (!is.numeric(x)) {
    stop("x must be numeric.")
  }
  if (R <= 0) {
    stop("R must be positive.")
  }
  if (!is.numeric(a)) {
    stop("a must be numeric.")
  }
  result <- numeric(length(x))
  for (i in 1:length(x)) {
    val <- x[i] - a
    if (abs(val) > R) {
      stop("x must be within radius.")
    }
    result[i] <- 0.5 + (val * sqrt(R^2 - val^2)) / (pi * R^2) + asin(val / R) / pi
  }
  return(result)
}
```

Semicircle Quantile Function Implementation

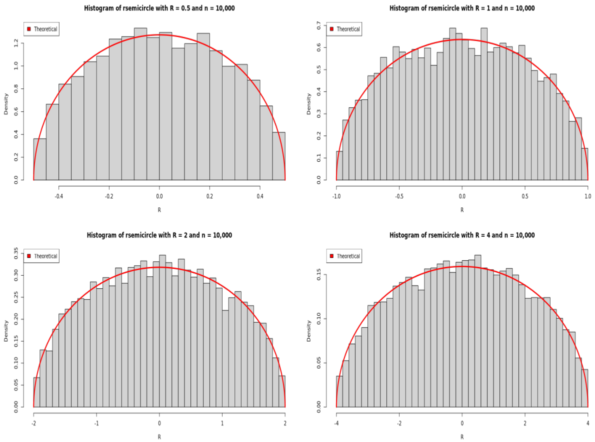
- The quantile function is the inverse CDF
- There is no closed-form solution
- To solve this issue, we used uniroot(), which is a built-in R function solving f(x) = 0
  - Finds root (value of x) within an interval such that f(x) is as close to 0 as possible
  - We solve psemicircle(x, R, a) - p = 0
  - Finds the quantile **x** so the cumulative probability up to **x** equals given **p**

qsemicircle Code

```
qsemicircle <- function(p, R, a = 0) {
  if (!is.numeric(p) || any(p < 0) || any(p > 1)) {
    stop("p must be between 0 and 1.")
  }
  if (R <= 0) {
    stop("R must be positive.")
  }
  if (!is.numeric(a)) {
    stop("a must be numeric.")
  }
  quantile_fn <- function(prob) {
    sapply(prob, function(pi) {
      if (pi == 0) return(a - R)
      if (pi == 1) return(a + R)
      uniroot(function(x) psemicircle(x, R, a) - pi,
              lower = a - R, upper = a + R)$root
    })
  }
  quantile_fn(p)
}
```

Random Generation for Value in Distribution (rsemicircle)

- Histograms show values generated for different values of R



Semicircle Random Generation Implementation (rsemicircle)

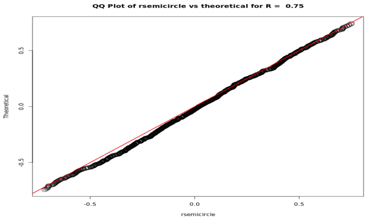
- Function signature: rsemicirle(n, R, a = 0)
  - **n** is the number of random values to generate
  - **R** is the radius
  - **a** is an optional shift parameter
- Performs validation on inputs
  - Such as making sure values are numeric and n is an integer
- Leverages the inverse CDF to generate a random value
  - The function generates random uniform values from 0 to 1, then passes these values into the inverse CDF function
  - Uses qsemicircle as the inverse CDF

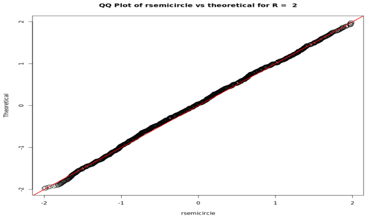
rsemicircle Code

```
rsemicircle <- function(n, R, a = 0) {
  if (!is.numeric(n) || n != round(n)) {
    stop("n must be an integer.")
  }
  if (R <= 0) {
    stop("R must be positive.")
  }
  if (!is.numeric(a)) {
    stop("a must be numeric.")
  }
  u <- stats::runif(n)  # generate n uniform values between 0 and 1
  qsemicircle(u, R, a)  # apply the inverse CDF (quantile function)
}
```

rsemicircle: Comparison with qsemicircle

- If we compare the quantiles of random values generated with rsemicircle with the theoretical quantiles from the distribution generated with qsemicircle we expect to see a straight line
- As shown on the right, the values fall approximately on a straight line





Testing

- Unit-level testing is performed using the R testthat library
- Testing exists for all 4 functions verifying the following:
  - The given parameters are of expected type (e.g. the radius is numeric)
  - The given parameters are within valid range (e.g. x must be within radius for psemicircle)
  - The functions support vectorization
  - The functions support the offset parameter (a)
  - The functions give the expected results for the given inputs, based on theoretical results
- Tests are run automatically in GitHub

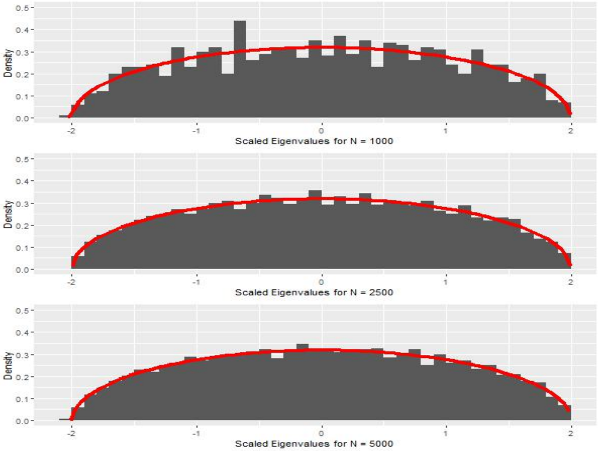
Testing

```
i Testing semicircledistr

Attaching package: 'testthat'

The following object is masked from 'package:devtools':

    test_file

✔ | F W  S  OK | Context
✔ |         24 | semicircle

══ Results ═══════════════════════════════════════════════
[ FAIL 0 | WARN 0 | SKIP 0 | PASS 24 ]
```

Verification of Wigner's Semicircle Law - Normal Distribution

- Recall: The distribution of the eigenvalues of an N x N i.i.d. random matrix will converge to the semicircle distribution as N goes to infinity
- Steps to verify this law using R simulation:
  - Simulate three Gaussian random matrices with different sizes (1000 x 1000, 2500 x 2500, 5000 by 5000)
  - Compute the eigenvalues for each matrix
  - Scale the eigenvalues for easier comparison across histograms - subtract each value in each eigenvalue vector by the mean of the vector and divide by the standard deviation of the vector
- After doing this, we found that the range of scaled eigenvalues for each matrix went from -2 to 2
- Applying a curve with R = 2 from our dsemicircle function:

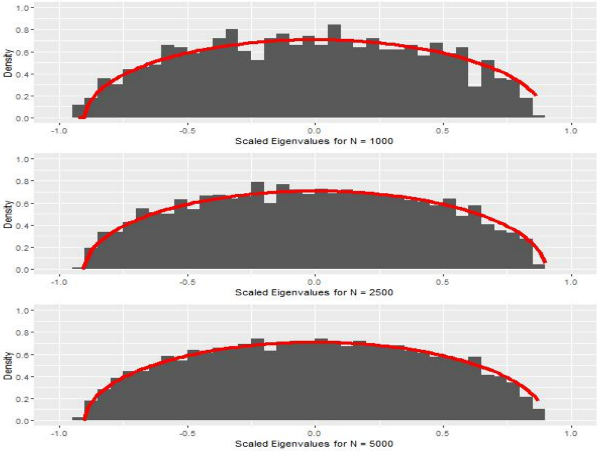
Verification of Wigner's Semicircle Law - Normal Distribution



Verification of Wigner's Semicircle Law - Exponential Distribution

- As far as we can tell from our online research, Wigner's semicircle law applies to any probability distribution - not just the normal distribution
- We followed the same steps on three exponential random matrices with $\lambda = 1$
- After scaling the eigenvalues, we found that their range was from about -1 to 1, so we chose to apply a curve with R = 0.9:

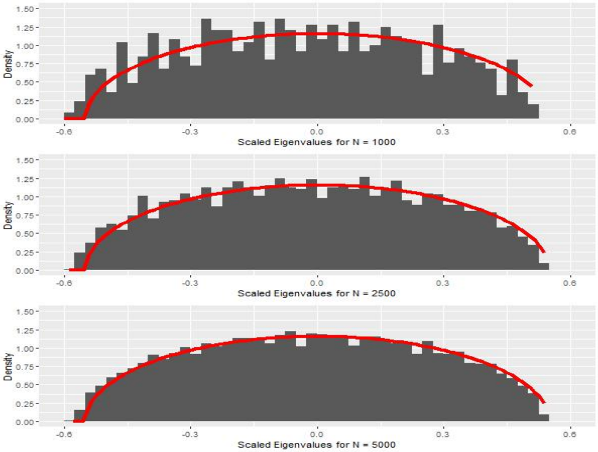
Verification of Wigner's Semicircle Law - Exponential Distribution



Verification of Wigner's Semicircle Law - Uniform Distribution

- We followed the same steps on three uniform random matrices with a min of 0 and a max of 1
- After scaling the eigenvalues, we found that their range was from about -0.6 to 0.6, so we chose to apply a curve with R = 0.55:

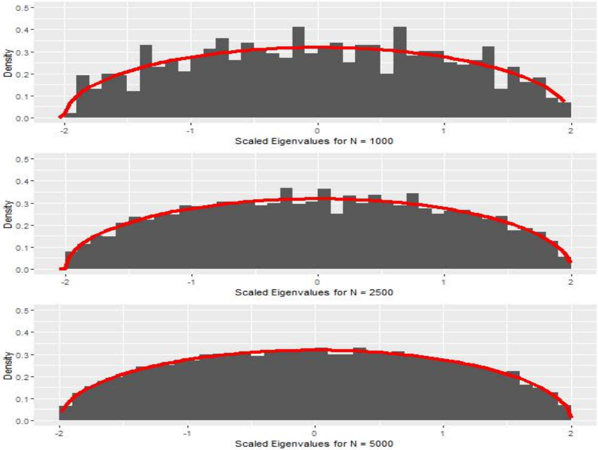
Verification of Wigner's Semicircle Law - Uniform Distribution



Verification of Wigner's Semicircle Law - Semicircle Distribution

- Finally, we followed the same steps on three semicircle random matrices with a radius of 2
- After scaling the eigenvalues, we found that their range was from -2 to 2, so we chose to apply a curve with R = 2:

Verification of Wigner's Semicircle Law - Semicircle Distribution



Verifying a Beta Transformation Theory

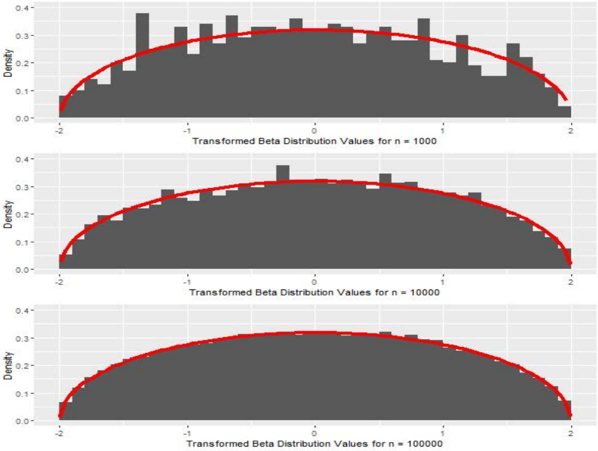
From Wikipedia:

"If Y is a beta-distributed random variable with parameters $\alpha = \beta = \frac{3}{2}$, then the random variable $2RY - R$ exhibits a Wigner semicircle distribution with radius R."

Verifying a Beta Transformation Theory

- Steps to verify this theory using R simulation:
  - Simulate three random vectors from a Beta distribution with parameters $\alpha = \beta = \frac{3}{2}$ and different sizes (1000, 10000, 100000)
  - Calculate a transformation of these vectors given $2RY - R$
  - Examine their distributions
- We chose R = 2 to test this theory
- Following the above steps output this plot:

Verifying a Beta Transformation Theory



Summary

- We created an R package to simulate the semicircle distribution, which does not have any other current R packages that simulate it
- This R package includes the four simulation functions, documentation for these functions, an overall README, and tests
- We accomplished everything we said we would in our proposal
- If we had more time to work on this project, we would have liked to do some more advanced math with the distribution simulation (calculating moments and conducting some more transforms, some of which are mentioned on Wikipedia)

Future Work

- This summer, we will submit this package to CRAN
- Before we do this, we still need to:
  - Edit the DESCRIPTION file to make it more informative
  - Select an official package maintainer
- Fortunately, we have already completed most of the requirements for having a CRAN-ready R package
- Dr. Bean may have some more insight on what we still need to do as the professor currently teaching a course all about R package development

References

- Jiang, T. (2021). *Wigner's semicircle law for Gaussian random matrices.* University of Chicago Mathematics REU. https://math.uchicago.edu/~may/REU2021/REUPapers/Jiang,Tianchong.pdf
- Li, L. H. (2020, January 29). *Wigner's semicircle law.* MIT Physics Directed Reading Program. https://phys-drp.mit.edu/sites/default/files/lhli_0.pdf
- Wikimedia Foundation. *Wigner semicircle distribution.* Wikipedia. https://en.wikipedia.org/wiki/Wigner_semicircle_distribution
- Wolfram. *Wigner's Semicircle Law.* Wolfram MathWorld. https://mathworld.wolfram.com/WignersSemicircleLaw.html