Spectral Statistics of Random Matrices

A Thesis
Presented to
The Division of Mathematics and Natural Sciences
Reed College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts

Ali Taqi

May 2021

Approved for the Division
(Mathematics)

_____

Jonathan M. Wells

List of Abbreviations

r.v	Random Variable
i.i.d	Independent and Identically Distributed
h.e.d	Homogenously and Explicitly Distributed
pdf	Probability Distribution Function
cdf	Cumulative Distribution Function

Table of Contents

List of Tables

Table of Random Matrix Distributions			
Distribution	Notation ($\mathcal{D}$)	Parameters	Class
Normal	$\mathcal{N}(\mu, \sigma)$	$\mu \in \mathbb{R}, \sigma \in \mathbb{R}^+$	Explicit (H)
Uniform	Unif(a, b)	$a, b \in \mathbb{R}$	Explicit (H)
Hermite-β	$\mathcal{H}(\beta)$	$\beta \in \mathbb{N}$	Explicit (NH)
Stochastic	Stoch	-	Implicit
Erdos-p	ER(p)	$p \in [0, 1]$	Implicit

Table of Spectrum Schema			
Scheme	Matrix	Notation	Ordering
Sign-Ordered	P	$\sigma_S(P)$	$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$
Norm-Ordered	P	$\sigma_N(P)$	$\lvert\lambda_1\rvert \geq \lvert\lambda_2\rvert \geq ... \geq \lvert\lambda_N\rvert$
Singular	$P \cdot P^T$	$\sigma_+(P)$	$\sqrt{\lambda_1} \geq \sqrt{\lambda_2} \geq ... \geq \sqrt{\lambda_N}$

Table of Dispersion Metrics				
Metric*	Notation	Formula	Symmetric	Parameters
Standard Norm	δ	$\lvert z' - z\rvert$	True	-
β-Norm	δ_β	$\lvert z' - z\rvert^\beta$	True	$\beta \in \mathbb{N}$
Difference of Absolutes	δ_{abs}	$\lvert z'\rvert - \lvert z\rvert$	False	-
Identity Difference	δ_{id}	$z' - z$	False	-

Table of Pairing Schema		
Scheme	Notation	Formula
Lower	$\Pi_<$	$\{(i, j) \mid i < j \text{ for } i, j \in \mathbb{N}_N\}$
Upper	$\Pi_>$	$\{(i, j) \mid i > j \text{ for } i, j \in \mathbb{N}_N\}$
Consecutive	Π_C	$\{(i, j) \mid i = j + 1 \text{ for } i, j \in \mathbb{N}_N\}$
All	Π_0	$\{(i, j) \mid i, j \in \mathbb{N}_N\}$

Abstract

On their own, random variables exude deterministic properties regarding their uncertainty. The same generalization can be made for random matrices, which are matrices whose entries are random variables. One particular statistic worth investigating is the distribution of a matrix ensemble's eigenvalues, or its spectrum. In this thesis, there will be an exploration of various classes of random matrices and relevant spectral statistics like their spectra and mixing times.

Dedication

For my mother.

Introduction

You begin to read the title of this thesis. The words random and matrix are familiar, and odds are you correctly surmise what random matrices are. If you cannot wait to know — they are matrices with entries which are random variables.

But, what are *spectral statistics*? Do they have to do with rainbows? Sceptres? Well, no they don't, but they're almost as colorful and regal. The word "spectral" is borrowed from the spectral-like patterns observed in statistical physics — whether it may be atomic spectra or other quantum mechanical phenomena. In that sense, the borrowing is loose and not literal, but still somewhat well-founded.

Our field of study, aptly called Random Matrix Theory, is a field at the intersection of Linear Algebra (matrices) and Probability Theory (random variables). For a primer or review of these subjects, **Appendix A** may be a useful resource. In any case, the field of Random Matrix Theory was extensively developed in the 1930s by the nuclear physicist Eugene Wigner. In fact, Wigner was interested in finding a model of atoms with heavy nuclei. In turn, he found connections between the deterministic properties of atomic nuclei and their random and stochastic behaviors. The link? Random matrices. (Wigner (1955))

So to the answer the question: in the context of this thesis, *spectral statistics* will be an umbrella term for random matrix statistics that somehow involve that matrix's eigenvalues and eigenvectors. Namely, we will consider two spectral statistics of random matrices:

1. Their eigenvalues, which we call their **spectra**.

2. The spacings between those eigenvalues, which we called their **dispersion**.

With all the technical jargon out of the way, we can state the intention of this paper. The intention is that a reader with a decent background in the prerequisites mentioned above can come out this paper with the toolkit necessary to study more advanced results in Random Matrix Theory. The field is rich and there are so many connections to other fields. Out of this paper, you will hopefully come out knowing many essential results, theorems, and ideas to build on top of. In a way, your journey reading through this is akin to my journey of learning the material. We achieve this by performing simulations and surveying important results in the field. Additionally, we report some new findings and provide an numerous examples of how random matrices can impact their spectral statistics by their parameterization/distribution.

On the way to achieve this goal, this thesis sets out to provide a standardized language and notation to talk about several objects in random matrix theory. This

will be discussed again soon.

So before dwelving into that, it is paramount to highlight that there is a large simulation component of this thesis. To be able to study any random objects, we need to be able to simulate it first. To simulate random matrices and explore their spectral statistics, this thesis will utilize the **RMAT** package.

The RMAT Package

> Tell me and I forget; teach me and I may remember; involve me and I learn. (Confucius)

As the quote above implies, interactivity is a critical part of learning. For this reason, all the necessary source code for this thesis will be made available. You are strongly encouraged to try out these simulations yourself! There will be instructions below for how to obtain the **RMAT** package. The Code Appendix (**Appendix C**) contains a minimalist version of the code needed to start these simulations.

As mentioned before, this package was developed alongside this thesis in order to facilitate the simulation of these random matrices and spectral statistics. To showcase the methodology of the simulations, code snippets will be sprinkled about the thesis. All code examples in this thesis are reproducible by setting the seed using $set.seed(23)$.

In any case, with the package explained, the honest truth is that the formalizations and definitions provided in this thesis were written after the code was. The definitions and formalizations were thought and developed as a result of the programming maneuvers that were needed to obtain the results we find in this thesis. A large and very important part of the thesis is developing intuitive definitions of random matrix objects that are consistent with the way the code **RMAT** package is implemented (i.e. the thesis in many ways formalizes some of the programming maneuvers used in **RMAT**).

Installing RMAT

There are three ways to get the RMAT package. In order of convenience:

1. CRAN: simply run $install.packages(\text{“}RMAT\text{”})$ in R.
2. Github: either run devtools::install_github(repo = “ataqi23/RMAT”) or clone the repository found here [1]
3. Source code: reproduce the code available in **Appendix C**

[1] github.com/ataqi23/RMAT

Chapter 1

Random Matrices

> Unfortunately, no one can be told what The Matrix is. You'll have to see it for yourself.
>
> *Morpheus*
> *The Matrix*

As discussed in the introduction, this thesis will be an exploration of spectral statistics of random matrices. This means that we must first be able to understand what random matrices are. At a fundamental level, random matrices are simply matrices which have *some or all* entries as random variables that are distributed in accordance to either some explicit distribution or algorithm. So, to define a random matrix, our approach will be to do so by formalizing and defining what it mean for one to be $\mathcal{D}$-distributed. This way, the notation encapsulates and completely characterizes the random matrix.

Prior to beginning the discussion on $\mathcal{D}$-distributions, the reader should be familiar or at least accquainted with the notion of random variables and what they are. For a review, see **Appendix A.2.1**.

1.1 $\mathcal{D}$-Distributions

As a general rule, when it comes to random simulation, there is usually a rule or constrain to which our randomness must conform. For example, sampling a vector from a distribution is a rudimentary example of this. For random matrices, there are various techniques for generating their entries that are not just sampling from theoretical distributions. As such, we motivate the $\mathcal{D}$-distribution: a generalized matrix entry distribution framework.

Definition 1.1.1 ($\mathcal{D}$-distribution)**.** *Suppose P is a $\mathcal{D}$-distributed random matrix. Then, we notate this $P \sim \mathcal{D}$. In the simplest of terms, $\mathcal{D}$ is essentially the algorithm that generates the entries of P. We define two primary methods of distribution:* ***explicit*** *distribution, and* ***implicit*** *distribution. If $\mathcal{D}$ is an explicit distribution, then*

some or all the entries of P are independent random variables with a given distribution. Otherwise, if $\mathcal{D}$ is implicit, then the matrix has dependent entries imposed by the algorithm that generates it.

1.1.1 Explicit Distributions

Homogenous Explicit $\mathcal{D}$-distributions

The simplest type of $\mathcal{D}$-distribution is one that is explicitly and homogenously distributed. From thereonafter, this will be shortened as e.h.d. Suppose $\mathcal{D}$ is a probability distribution for random variables in the classical sense. The simplest way to think of e.h.d distributions is to use the concept of notational overload. For example, it is unambiguous to say that an r.v. $X \sim \mathcal{N}(0,1)$. However, the same cannot be said if we said a matrix $P \sim \mathcal{N}(0,1)$.

That being said, we can define e.h $\mathcal{D}$-distributions as an notational extension that means **every** entry of the matrix is an i.i.d random variable with that same distribution! In otherwords, we simply perform entry-wise sampling from the corresponding r.v. distribution.

Note that this means by construction, $\mathcal{D}$ can only be e.h.d. if it has a corresponding probability distribution for random variables.

Definition 1.1.2 (Homogenous Explicit $\mathcal{D}$-distribution)**.** *Suppose $P \sim \mathcal{D}$ where $\mathcal{D}$ is a homogenous and explicit distribution. Additionally, let $\mathcal{D}^*$ denote the corresponding random variable analogue of $\mathcal{D}$. Then, every single entry of P is an i.i.d random variable with the corresponding distribution. That is,*

$$P \sim \mathcal{D} \iff \forall i, j \mid p_{ij} \sim \mathcal{D}^*$$

Example. *Suppose $P \sim \mathcal{N}(0,1)$ and that P is a 2×2 matrix. Then, $p_{11}, p_{12}, p_{21}, p_{22}$ are independent, identically distributed random variables with the standard normal distribution.*

Algorithm 1.1.1 (Homogenous Explicit $\mathcal{D}$-Matrix)**.**

1. *To simulate a $\mathcal{D}$-distributed square matrix P of size N, fix $N \in \mathbb{N}$.*
2. *Sample a vector $\vec{X}$ with N i.i.d entries from $\mathcal{D}$. So, generate $\vec{X} = X_1, \ldots, X_N$ where $X_i$is i.i.d $\mathcal{D}$.*
3. *Assign the vector $\vec{X}$ as a row of the matrix P. Repeat for every other row.*
4. *Return the $\mathcal{D}$-distributed matrix P.*

Formalization. *Explicit homogenous distributions can be formalized as overloading the standard notation of random variable distribution as seen in probability theory. This way, our random matrix has a representation as a random vector, which we commonly encounter in probability theory as a (i.i.d) sequence of random variables! Suppose P is an $N \times N$ random matrix that is explicity and homogenously $\mathcal{D}$-distributed. Then, this would mean that P is an array of N^2 i.i.d random variables sampled from $\mathcal{D}$.*

Non-Homogenous Explicit $\mathcal{D}$-distributions

There are a few instances where we encounter the need to only initialize only a subset of a matrix's entries as random variables. In this case, we say that a matrix has a non-homogenous explicit $\mathcal{D}$-distribution. This type is distribution is similar to e.h. $\mathcal{D}$-distributions, but we no longer have the ability to overload random variable notation and indicate that only some entries have such distribution. This would be confusing because not every has the same distribution anymore. As such, we have to define new notation to do so.

Simply put, a non-homogenous explicit $\mathcal{D}$-distribution can be completely characterized by listing the distributions of every entry. In this thesis, there are only one application of this type of $\mathcal{D}$-distribution; however, it describes the entry disributions by its diagonals — this is the Hermite β-ensemble matrices. Since its definition is characterized by distributions on the diagonals, we will avoid full entry-wise generality to be more concise. As such, we will only describe non-homogenous explicit $\mathcal{D}$-distributions as schemes where we assign diagonal bands a specific vector of i.i.d random variables.

Definition 1.1.3 (Diagonal Bands)**.** *Suppose $P = (p_{ij})$ is an $N \times N$ matrix. Then, P may be partitioned into $2n-1$ rows called diagonal bands. Each band is denoted $[\rho]$ where $\rho = \{p_{ij} \mid \rho = i - j\}$. We have $\rho \in \{-(N-1), \ldots, -1, 0, 1, \ldots, N-1\}$.*

Heres an example using the diagonal bands constructor notation.

Example. *Suppose P is an $N \times N$ matrix. Then, $[0]_P$ is the main diagonal of P since $p_{ii} \Rightarrow i = j \Rightarrow i - j = 0 \Rightarrow p_{ii} \in [0]_P$. Similarly, the main off-diagonal in the upper triangle is $[-1]_P$ since $p_{12} \in [-1]_P$. Likewise, the main off-digonal in the lower triangle is $[1]_P$. The entry in the top-right corner of the matrix, p_{1N} comprises $[1-N]_P$.*

Code Example. *Here is an example utilizing diagonal bands constructors in R.*

```
# Set the dimension of the matrix
N <- 5
# Generate an example matrix of zeros
P <- matrix(rep(0, N^2), nrow = N)
# Assign the upper main off-diagonal band as a vector
rho <- -1
P[row(P) - col(P) == rho] <- rnorm(n = 4, mean = 0, sd = 1)
# Assign the main diagonal as a vector
rho <- 0
P[row(P) - col(P) == rho] <- rep(10, N)
# Returns the following
P
     [,1]        [,2]       [,3]       [,4]      [,5]
[1,]   10   0.1932123  0.0000000  0.0000000  0.000000
[2,]    0  10.0000000 -0.4346821  0.0000000  0.000000
```

```
[3,]    0  0.0000000 10.0000000  0.9132671  0.000000
[4,]    0  0.0000000  0.0000000 10.0000000  1.793388
[5,]    0  0.0000000  0.0000000  0.0000000 10.000000
```

For the distribution using diagonal bands, consider the definiton of the beta matrix below.

Algorithm 1.1.2 (Dumitriu's Beta Matrix)**.**

1. *To simulate an $N \times N$ beta matrix, fix $N \in \mathbb{N}$.*
2. *Start by taking a diagonal of $\mathcal{N}(0, 2)$ variables.*
3. *Set both of the nearest off-diagonals to the row that samples from a $\chi(df = c_j)$ where $c_j = \beta \cdot j$ for columns spanning $j = 1, \ldots, n-1$.*

Definition 1.1.4 (β-Matrix)**.** *Suppose $P \sim \mathcal{H}(\beta)$. Then, $[0]_P \sim \mathcal{N}(0, 1)$. Additionally, $[-1]_P = \vec{X}$ where $X_k = \chi(df = \beta k)$ for $k = 1, \ldots, N-1$. Then, set $[1]_P = [-1]_P$.*

Warning (β-Notation)**.** *Please note that the Hermite β-ensemble matrices are **not** related to the Beta distribution. They are completely independent. Instead, $\beta \in \mathbb{N}$ is a natural number that determines the nature of the matrix ensemble (more later). The Beta(a, b) distribution is a r.v. distribution that takes in two parameters. For this reason, it is possible to have a homogenous explicit distribution $P \sim$ Beta(a, b), but they do not mean the same thing.*

1.1.2 Implicit Distributions

In the latter case, we are concerned less about the distribution of the matrix entries and moreso about its holisitic properties.

Consider for example, the following implicit distributions.

1. If $\mathcal{D}$ = Stochastic, then the matrix is a row of random stochastic rows. (See Algorithm B.x)
2. If $\mathcal{D}$ = ER(p), then the matrix is a row of Erdos-Renyi p-stochastic rows. (See Algorithm B.x)

Stochastic Matrices

Stochastic matrices will serve as our canonical implicitly distributed $\mathcal{D}$-distributed matrices. One might ask, what are stochastic matrices? How are they distributions?

Stochastic matrices, in short, are matrices that represent random walks on Markov Chains (see **Appendix A.3**). Conventionally, stochastic matrices represent random walks on fixed graphs - so what does it mean to sample a random stochastic matrix? Well, the algorithm tells us that sampling a random stochastic matrix represents a random walk on a fixed random graph with **randomized weights**. In other words, we sample a fully connected graph, and randomize the weights of each

edge. In **Section 1.2.1**, we explore walks on Erdos-Renyi graphs, which are graphs with sparsity as opposed to fully connected graphs.

Consider the following generating algorithms below. We first start by generating a stochastic row - a row that sums to 1.

Algorithm 1.1.3 (Stochastic Row)**.**

1. *To sample a row r of size N, fix $N \in \mathbb{N}$.*
2. *Sample a vector $\vec{X}$ with N i.i.d entries between $[0, 1]$. So, sample $\vec{X} \sim Unif(0, 1)$.*
3. *Assign $r \leftarrow \vec{X}$, and then normalize the row by diving each entry by the row sum; so assign $r \leftarrow r \cdot \frac{1}{\sum_{i=1}^{N} r_i}$*
4. *Return the stochastic row r.*

Algorithm 1.1.4 (Stochastic Matrix)**.**

1. *To generate a stochastic square matix P of size $N \times N$, fix $N \in \mathbb{N}$.*
2. *Then, for every row of P, randomly sample a stochastic row of size N and assign it.*
3. *Return the stochastic matrix P.*

A stochastic row is a vector whose entries are all dependent. This is because we normalize by the row sum, making things very difficult. Since it is not the perview of this thesis, we won't try to find the exact distribution. As such, to avoid the process of finding the distribution of these dependent entries, we will encapsulate the distribution and abstract it away by calling these implicit distributions!

1.1.3 Random Matrices

With the various types of $\mathcal{D}$-distributions defined, the definition of a random matrix is quite simple.

Definition 1.1.5 (Random Matrix)**.** *Let $P \sim \mathcal{D}$ be an $N \times N$ matrix over $\mathbb{F}$. Then, the entries of P are elements in $\mathbb{F}$ completely determined by the $\mathcal{D}$-distribution, regardless of what type it is. Also, if $\mathcal{D}$ is an explicit distribution, $\mathcal{D}^{\dagger}$ represents the symmetric/hermitian version $\mathcal{D}$.*

Here is a clarification on symmetric/hermitian versions of distributions.

Remark (Symmetric/Hermitian Matrices)**.** *As mentioned in the definition, we automatically have a class of derivative $\mathcal{D}$-distributions given that they are explicitly distributed denoted by $\mathcal{D}^{\dagger}$. To make a matrix symmetric (or Hermitian if $\mathbb{F} = \mathbb{C}$), then we simply set the elements in the upper triangle to be equal to those in the lower triangle. There is also an alternative algorithm provided in the appendix. (See B.x)*

Sometimes, our distribtions may depend on the dimension of the matrix N.

Remark (Dependence on N)**.** *Every implicit $\mathcal{D}$-distribution will have some sort of dependence on the dimension of the matrix N. Non-homogenous explicit distributions may have a dependence, in the case of Hermite-β matrices, it happens that they do. It is only e.h. $\mathcal{D}$-distributions that never have a dependence on N, matrices with such distributions have entries which are i.i.d.*

As mentioned in the definition, a random matrix defined over a field $\mathbb{F}$ has entries from $\mathbb{F}$ that are determined by the $\mathcal{D}$-distribution. Sometimes, specify that a matrix may have complex entires. We notate this by specifying $\mathbb{F} = \mathbb{C}$, and signify it as described below.

Remark (Complex Entries)**.** *To say that a random matrix is explicitly $\mathcal{D}$-distributed over $\mathbb{C}$ would mean that its entries take the form $a + bi$ where $a, b \sim \mathcal{D}$ are random variables. In other words, if we allow the matrix to have complex entries by setting $\mathbb{F} = \mathbb{C}$, then we must sample the real and imaginary component as $\mathcal{D}$-distributed i.i.d. random variables. Note that this means we cannot set the field for implicit $\mathcal{D}$ distributions, as the field is automatically chosen by the generative algorithm.*

Below, we can see code on how to generate a standard normal random matrix using the **RMAT** package.

Code Example (Standard Normal Matrix)**.** *Let $\mathcal{D} = \mathcal{N}(0, 1)$. We can generate $P \sim \mathcal{D}$, a 4×4 standard normal matrix, as such:*

```
library(RMAT)
P <- RM_norm(N = 4, mean = 0, sd = 1)
# Outputs the following
P
            [,1]        [,2]        [,3]         [,4]
[1,]  0.1058257 -1.0835598 -0.7031727  1.01608625
[2,] -0.2170453  1.8206070 -0.4539230  0.06828296
[3,]  1.3002145  0.1254992 -0.5214005 -0.61516174
[4,] -1.0398587  0.1975445 -0.8511950  0.86366082
```

1.2 The Crew: Ensembles

With a random matrix well defined, we may now motivate one of the most important ideas - the random matrix ensemble. One common theme in this thesis will be that random matrices on their own provide little information. When we consider them at the ensemble level, we start to obtain more fruitful results. Without further ado, we motivate the random matrix ensemble.

Definition 1.2.1 (Random Matrix Ensemble)**.** *A $\mathcal{D}$-distributed random matrix ensemble $\mathcal{E}$ over $\mathbb{F}^{N\times N}$ of size K is defined as a set of $\mathcal{D}$-distributed random matrices $\mathcal{E} = \{P_i \sim \mathcal{D} \mid P_i \in \mathbb{F}^{N\times N}\}_{i=1}^{K}$. In simple words, it is simply a collection of K iterations of a specified class of random matrix.*

So, for example, we could compute a simple ensemble of matrices as follows.

Code Example (Standard Normal Hermitian Ensemble)**.** *Let $\mathcal{D} = \mathcal{N}(0,1)^{\dagger}$. We can generate $\mathcal{E} \sim \mathcal{D}$ over $\mathbb{C}$, an ensemble of 4×4 complex Hermitian standard normal matrices of size 10 as such:*

```
library(RMAT)
# By default, mean = 0 and sd = 1.
ensemble <- RME_norm(N = 4, cplx = TRUE, herm = TRUE, size = 10)
# Outputs the following
ensemble
...
[[10]]
                     [,1]                 [,2]                 [,3]
[1,] -0.59931+1.24286i  1.29457+0.66058i  0.83539-0.16662i
[2,]  1.29457-0.66058i  0.78841+0.09818i -1.16592+1.14666i
[3,]  0.83539+0.16662i -1.16592-1.14666i -0.51256+0.17750i
```

With this in mind, we will survey and characterize, and briefly discuss a few special recurring ensembles in this thesis.

1.2.1 Erdos-Renyi p-Ensembles

The Erdos-Renyi transition matrices are a class of stochastic matrices. Any stochastic matrix represents a walk on a fixed graph, the particular class of random graphs that we will consider are the Erdos-Renyi random graphs. Essentially, these are graphs whose vertices are connected with a uniform probability p. We can interpret this as saying an Erdos-Renyi graph is a simple random walk on a graph with parameterized sparsity (given by p). Without further ado, we motivate the Erdos-Renyi graph:

Definition 1.2.2 (Erdos-Renyi Graph)**.** *An Erdos-Renyi graph is a graph $G = (V, E)$ with a set of vertices $V = \{1, \ldots, N\}$ and edges $E = \mathbb{1}_{i,j\in V} \sim Bern(p_{ij})$. It is homogenous if $p_{ij} = p$ is fixed for all i, j.*

Essentially, an Erdos-Renyi graph is a graph whose 'connectedness' is parameterized by a probability p (assuming it's homogenous, which this document will unless otherwise noted). As $p \to 0$, we say that graph becomes more sparse; analogously, as $p \to 1$ the graph becomes more connected.

Recall from probability theory that a sum of i.i.d Bernoulli random variables is a Binomial variable. As such, we may alternatively say that the degree of each vertex v is distributed as $deg(v) \sim Bin(N, p)$ where N is the number of vertices. This makes simulating the graphs much easier. This is demonstrated in the algorithm used to generate the matrices. (See B.x)

Algorithm 1.2.1 (Transition Matrix for an Erdos-Renyi Graph)**.**

1. *Fix $N \in \mathbb{N}$ and $p \in [0, 1]$.*
2. *Generate a matrix Q such that every entry $i, j \in 1, \ldots, N$ is $x_{ij} \sim Unif(0, 1)$.*
3. *For each row r_i in $\{1, \ldots, N\}$, generate $deg(v_i) \sim Bin(N, p)$.*
4. *Randomly chose $N - deg(v_i)$ vertices, set the entries x_{ij} in the j columns to 0 to sever them.*
5. *Renormalize the matrix by dividing each row by its sum; let $(x_i) \leftarrow (x_i)/\sum_j (x_i)$.*

Warning. *Note that we are not considering the adjacency matrix of an Erdos-Renyi graph. Rather, we are simulating a transition matrix that* ***represents a walk on one****.*

Code Example (Erdos-Renyi p = 0.5 Ensemble)**.** *Let $\mathcal{D} = ER(p = 0.5)$. We can generate $\mathcal{E} \sim \mathcal{D}$, an ensemble of 4×4 Erdos-Renyi matrices ($p = 0.5$) of size 10 as such:*

```
library(RMAT)
ensemble <- RME_erdos(N = 4, p = 0.5, size = 10)
# Outputs the following
ensemble
...
[[10]]
          [,1]      [,2]      [,3]     [,4]
[1,] 0.0000000 0.1729581 0.8270419 0.000000
[2,] 0.0000000 0.0000000 1.0000000 0.000000
[3,] 0.2557890 0.3766740 0.0000000 0.367537
[4,] 0.2151029 0.3929580 0.3919391 0.000000
```

1.2.2 Hermite β-Ensembles

The Hermite β-Ensembles will be one of the primary ensembles discussed in this thesis. The Hermite β-ensembles are a normal-like class of random matrices. This ensemble will be characterized, motivated, and defined more thoroughly in **Chapter 4**. However, we will give a brief introduction to the ensemble.

At a practical level, all one would need to know is that the matrices are generated in accordance to an algorithm found in the appendix (Algorithm B.x) and in Chapter 4.

Summary Table of $\mathcal{D}$-Distributions

Table of Random Matrix Distributions			
Distribution	Notation ($\mathcal{D}$)	Parameters	Class
Normal	$\mathcal{N}(\mu, \sigma)$	$\mu \in \mathbb{R}, \sigma \in \mathbb{R}^+$	Explicit (H)
Uniform	Unif(a, b)	$a, b \in \mathbb{R}$	Explicit (H)
Hermite-β	$\mathcal{H}(\beta)$	$\beta \in \mathbb{N}$	Explicit (NH)
Stochastic	Stoch	-	Implicit
Erdos-p	ER(p)	$p \in [0, 1]$	Implicit

Chapter 2

Spectra

Life is like a box of crayons.

Unknown

2.1 Introduction

In the context of this thesis, *spectral statistics* will be an umbrella term for random matrix statistics that somehow involve that matrix's eigenvalues and eigenvectors. That being said, if we fix a *random matrix*, we can study its features by studying its eigenvalues - fundemental numbers that tell us a lot about the matrix. They are quite important for many reasons. For instance in statistical physics, many processes are represented by operators or matrices, and as such, their behaviours could be partially determined by the eigenvalues of their corresponding matrices. The study of eigenvalues and eigenvectors primarily falls in the scope of Linear Algebra, but their utility is far-reaching. So, what are *eigenvalues* exactly?

2.1.1 The Quintessential Spectral Statistic: the Eigenvalue

Given any standard square matrix $P \in \mathbb{F}^{N\times N}$, its *eigenvalues* are simply the roots of the characteristic polynomial $\text{char}_P(\lambda) = \det(P - \lambda I)$. By the Fundamental Theorem of Algebra, we know that there is always have as many complex eigenvalues $\lambda \in \mathbb{C}$ as the dimension of the matrix.

That being said, when our random matrix has a specified distribution (say, standard normal), we can see patterns in the eigenvalue distributions. So, an eigenvalue is a **spectral statistic** of a random matrix! To talk about a matrix's eigenvalues in a more formal and concise manner, we motivate what is the eigenvalue spectrum.

Definition 2.1.1 (Spectrum)**.** *Suppose $P \in \mathbb{F}^{N\times N}$ is a square matrix of size N over $\mathbb{F}$. Then, the (eigenvalue) spectrum of P is defined as the multiset of its eigenvalues and it is denoted*

$$\sigma(P) = \{\lambda_i \in \mathbb{C} \mid char_P(\lambda_i) = 0\}_{i=1}^{N}$$

Note that it is important to specify that a spectrum is a multiset and not just a set; eigenvalues could be repeated due to algebraic multiplicity and we opt to always have N eigenvalues.

For example, consider the following code example from the RMAT package.

Code Example (Spectrum of a Standard Normal Matrix)**.** *Let $P \sim \mathcal{N}(0,1)$ be a 5×5 standard normal random matrix. We can generate the spectrum of P, $\sigma(P)$ as follows:*

```
library(RMAT)
P <- RM_norm(N = 5, mean = 0, sd = 1)
spectrum_P <- spectrum(P)
# Outputs the following
spectrum_P
...
      Real     Imag   Norm Order
 1 -0.5434   1.3539 1.4589     1
 2 -0.5434  -1.3539 1.4589     2
 3  0.2255   1.4250 1.4427     3
 4  0.2255  -1.4250 1.4427     4
 5 -0.8678   0.0000 0.8678     5
```

[Explain what each column represents]

Since this is the first time the output of the spectrum appears, it may be worth briefly specifying what each column represents (i.e. Re denotes real, Im is imaginary, norm is complex norm and order is ranking from largest norm to smallest

While our definition for spectrum is nice and clean, there are a few caveats that we must take care of. First, how are spectra statistics? So we can even call them statistics, so there needs to be some formalization as to why we consider eigenvalues statistics. In this dialogue, we will call back on the same sort of formalization dialogue in the Random Matrices section. Before beginning, the reader is encouraged to review what a statistic is in the review appendix (A.x).

Recall that when we defined and motivated the $\mathcal{D}$-distribution framework of simulating random matrices,we always had one thing - a vector representation of random variables. Using this framework, the formalization is trivial.

Formalization. *Suppose $P \sim \mathcal{D}$ is an $N \times N$ random matrix. Then, P has a representation as a sequence of K random variables (for some $K \in \mathbb{N}$). Denote it as the vector $\vec{X} = X_1, X_2, \ldots, X_K$. Then, the spectrum of the matrix P is simply a function of the vector $\vec{X}$. We can denote this $\sigma(\vec{X})$, where the operator σ is overloaded to mean the spectrum of a matrix* ***with respect to the vector representation****. The actual process for σ is not necessary to explicitly write out, since characterizing it will be sufficient for now. Essentially, σ is a function that would parse the random variables into the array form by index hacking. Then, it must compute the determinant of the matrix $P - \lambda I$, and solve for its roots. To summarize this, consider the flow chart below.*

To summarize, here is how we formalize the spectrum as a statistic. Suppose we sample $P \sim \mathcal{D}$. Then, we take its spectrum formally using σ as such:

$$X_1, X_2, \ldots, X_K \xrightarrow{\text{Make array}} (\vec{X} \rightsquigarrow P^*) \xrightarrow{\text{Compute } \det(P-\lambda I)} char_{P^*}(\lambda) \xrightarrow{\text{Solve for roots}} \sigma(P^*)$$

Remark (Formalization)**.** *Note how in the formalization we said that an $N \times N$ matrix P has a representation as a vector of K random variables rather than N^2 variables. Recall that some non-homogenous explicit $\mathcal{D}$-distributions do not initialize every entry of the matrix. Only our homogenous explicit and implicit $\mathcal{D}$-distributions do.*

2.1.2 Interlude: Ensembles

While the spectrum of a matrix provides a good summary of the matrix, a matrix is only considered a single point/observation in random matrix theory. Additionally, simulating large matrices and computing their eigenvalues becomes harder and more computationally expensive as $N \to \infty$. As such, to obtain more eigenvalue statistics efficiently, another dimension is introduced by motiving the *spectrum of a random matrix ensemble*. If we have an ensemble $\mathcal{E}$, then we can naturally extend the definition of $\sigma(\mathcal{E})$.

Definition 2.1.2 (Ensemble Spectrum)**.** *Let $\mathcal{E} \sim \mathcal{D}$ be an ensemble of matrices $P_i \in \mathbb{F}^{n \times n}$. To take the spectrum of $\mathcal{E}$, simply take the union of the spectra of each of its matrices. In other words, if $\mathcal{E} = \{P_i \sim \mathcal{D}\}_{i=1}^{K}$, then we denote the spectrum of the ensemble*

$$\sigma(\mathcal{E}) = \bigcup_{i=1}^{K} \sigma(P_i)$$

For example, consider the following code example from the RMAT package.

Code Example (Spectrum of a Standard Normal Matrix Ensemble)**.** *Let $\mathcal{E} \sim \mathcal{N}(0,1)$ be an ensemble of 3×3 standard normal random matrices of size 3. We can generate the spectrum of $\mathcal{E}$, $\sigma(\mathcal{E})$ as follows:*

```
library(RMAT)
ens <- RME_norm(N = 3, mean = 0, sd = 1, size = 3)
spectrum_ens <- spectrum(ens)
# Outputs the following
spectrum_ens
...
      Real     Imag   Norm Order
 1  1.7581   0.0000 1.7581     1
 2 -0.2614   1.0012 1.0347     2
 3 -0.2614  -1.0012 1.0347     3
 4  1.2327   0.4227 1.3032     1
 5  1.2327  -0.4227 1.3032     2
```

```
6 -0.8504   0.0000 0.8504      3
7 -0.5296   1.0508 1.1767      1
8 -0.5296  -1.0508 1.1767      2
9  0.7357   0.0000 0.7357      3
```

A common theme in this thesis will be that singleton matrices do not provide insightful information on their own. Rather, it is the collective behavior of a $\mathcal{D}$-distributed ensemble that tells us about how $\mathcal{D}$ impacts our spectral statistics. So in a way, ensemble statistics are the engine of this research.

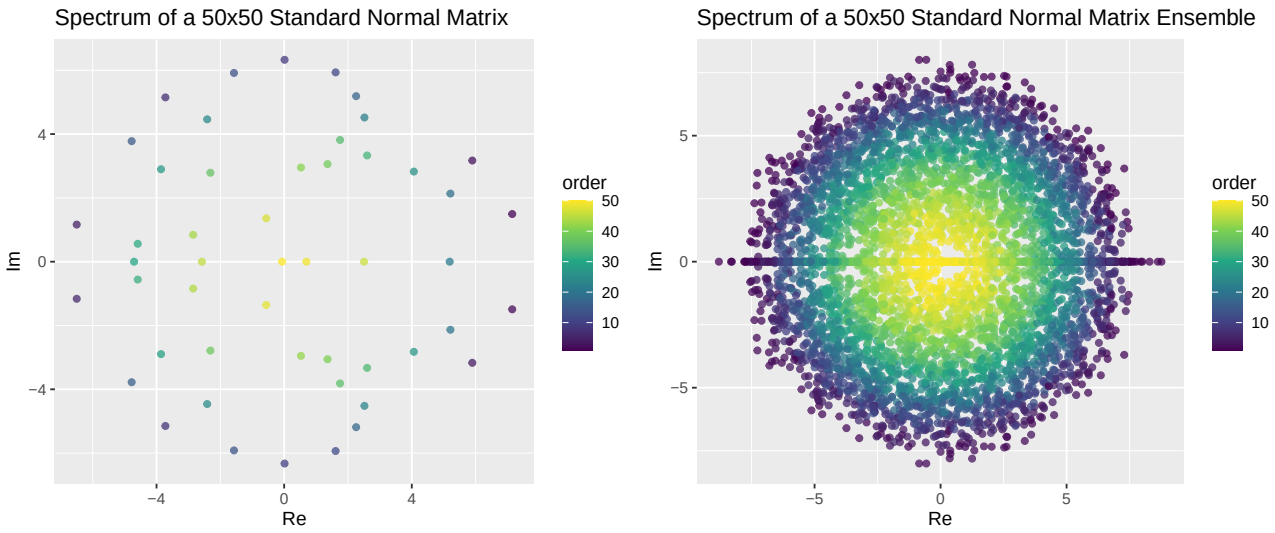


Figure 2.1: Spectrum of a Matrix versus an Ensemble

[Plot takeaways]

Include brief discussion (1 paragraph) about what the reader should take away from this graph in particular

2.2 Spectrum Analysis

2.2.1 Ordered Spectra

When we motivate the idea of matrix dispersion in the next section, we will consider order statistics of that matrix's eigenvalues in tandem with its dispersion. However, to do so presupposes that we have a sense of what *ordered* eigenvalues means. Take a matrix P and its *unordered* spectrum $\sigma(P) = \{\lambda_j\}$. It is paramount to know what ordering scheme $\sigma(P)$ is using, because otherwise, the eigenvalue indices are meaningless! So, to eliminate confusion, we add an index to σ that indicates how the spectrum is ordered. Often, the ordering context will be clear and the indexing will be omitted.

Order Schemes: How to Order Eigenvalues

Consider the two following *ordering schema*:

Standard definitions of an ordered spectrum follows the standard ordering in the reals; denote this as the ordering by the **sign scheme**. Note that because total-ordering is only well-defined on the reals, we can only use this scheme when on a spectrum with real entries. So, we write the *sign-ordered spectrum* as follows:

$$\sigma_S(P) = \{\lambda_j : \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N\}_{j=1}^N$$

Alternatively, we can motivate a different scheme that properly handles complex eigenvalues. We could sort the spectrum by the norm of its entries; denote this as ordering using the *norm scheme*. This way, all the eigenvalues are mapped to a real value, in which we could use the sign-scheme of ordering. Without further ado, we write the *norm-ordered spectrum* as follows:

$$\sigma_N(P) = \{\lambda_j : |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_N|\}_{j=1}^N$$

Note that when we take the norms of the eigenvalues, we essentially ignore "rotational" features of the eigenvalues. Signs of eigenvalues indicate reflection or rotation, so when we take the norm, we essentially become more concerned with scaling.

For example, consider the following plots, showing the difference in using the sign and norm ordering schemes for the same spectrum.

Singular Values

An aternative to using the norm ordering scheme is using the singular values of the matrix. If a matrix is symmetric, the singular values are simply the norm of the eigenvalues. We ignore rotational features and focus solely on scale when we do so.

Suppose P is a random matrix. Then, we can take it singular values as such.

Definition 2.2.1 (Singular Values)**.** *The singular values of a matrix P are given by the square root of the eigenvalues of the corresponding product of that matrix and its transpose. That is, $\sigma_+(P) = \sqrt{\sigma(P \cdot P^T)}$.*

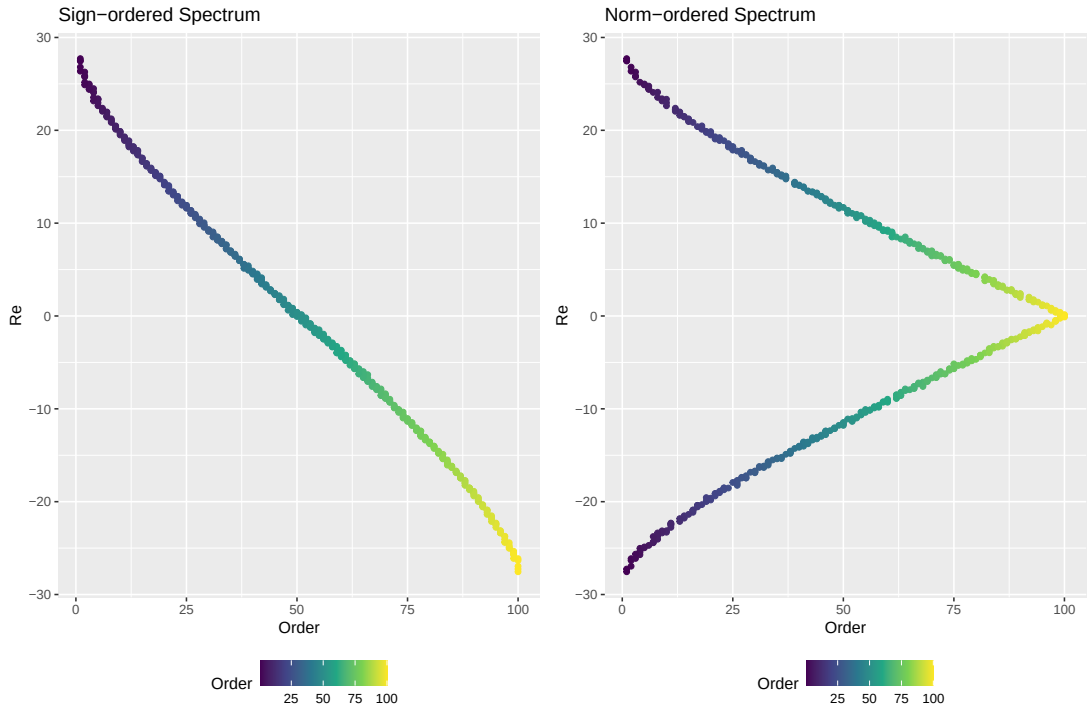


Figure 2.2: Spectrum displaying two different ordering scheme

Summary Table of Spectrum Schema

Table of Spectrum Schema			
Scheme	Matrix	Notation	Ordering
Sign-Ordered	P	$\sigma_S(P)$	$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$
Norm-Ordered	P	$\sigma_N(P)$	$\lvert\lambda_1\rvert \geq \lvert\lambda_2\rvert \geq ... \geq \lvert\lambda_N\rvert$
Singular	$P \cdot P^T$	$\sigma_+(P)$	$\sqrt{\lambda_1} \geq \sqrt{\lambda_2} \geq ... \geq \sqrt{\lambda_N}$

Order Statistics

With eigenvalue ordering unambiguous and well-defined, we may proceed to start talking about their order statistics. In short, given a random sample of fixed size, order statistics are random variables defined as the value of an element conditioning on its rank within the sample. (See A.x)

In general, order statistics are quite useful and tell us a lot about how the eigenvalues distribute given a distribution. They tell us how the eigenvalues space themselves and give us useful upper and lower bounds.

For example, the maximum of a sample is an order statistic concerned with the highest ranked element. In our case, this could correspond the largest eigenvalue of a spectrum. After all, a spectrum is a random sample of fixed size, so this statistic is well-defined.

Example (The Largest Eigenvalue)**.** *Suppose we have seek the largest eigenvalue distribution for a ensemble distribution $\mathcal{D}$, we would simulate an ensemble $\mathcal{E}$ and observe λ_1 for each of its matrices. Then, we can set the distribution of the largest eigenvalue for $\mathcal{D}$ by observing the distribution of λ_1.*

So, we will consider the conditional order statistics $\mathbb{E}(\lambda_i \mid i)$ and $\text{Var}(\lambda_i \mid i)$.

2.2.2 Case Study: Perron-Frobenius Theorem

Definition 2.2.2 (Spectral Radius)**.** *Let be P be any matrix and $\sigma(P)$ be its ordered spectrum. Then, the spectral radius of P is defined as $\rho(P) = ||\sup \sigma(P)||$ is the norm of its largest eigenvalue.*

Detour: The Perron-Frobenius Theorem

Using the toolkit we have now accquired, we can now discuss an elegant, visual representation of the Perron-Frobenius theorem (see **Section A.1.2**). To put it shortly, the Perron-Frobenius theorem, applied to stochastic matrices is a result that guarantees the existence of a stationary distribution to an ergodic Markov Chain (see **Section A.3**). That being said, the following facts give us a heuristic demonstration of the Perron-Frobenius theorem.

Theorem 2.2.1 (Perron-Frobenius Theorem)**.**
Consider the two following facts:

1. *The largest eigenvalue of stochastic matrices is 1.*
2. *When multiplied by P^K, any point v asymptotically enters the eigenspace of the matrix's largest eigenvalue as $K \to \infty$. That is, as K grows, vP^K approaches eigenvector of P of λ_1.*

So, there is an eigenvector of the largest eigenvalue — for a Markov Chain, this is a stationary distribution.

Note. *Note that this is only part of the results of the theorem. The theorem is more general and has a wider scope and applies to matrices in a more general fashion. We only demonstrate this because we have a case with a constant largest eigenvalue and an eigenvector with a unique interpretation (a stationary distribution).*

Again, those two statements were not formally proven in this thesis. However, there are large amounts of computational evidence for both of these results. Consider the first statement.

Below we have a stochastic ensemble spectrum. Notice how the largest eigenvalue is 1, living far from the island of **complex** eigenvalues.

This shows that the spectral radius of stochastic matrices is 1! What is left is the other fact. For the second statement, again, there is plenty of computational evidence to back this up. The reader may refer to **Appendix D** for an empirical

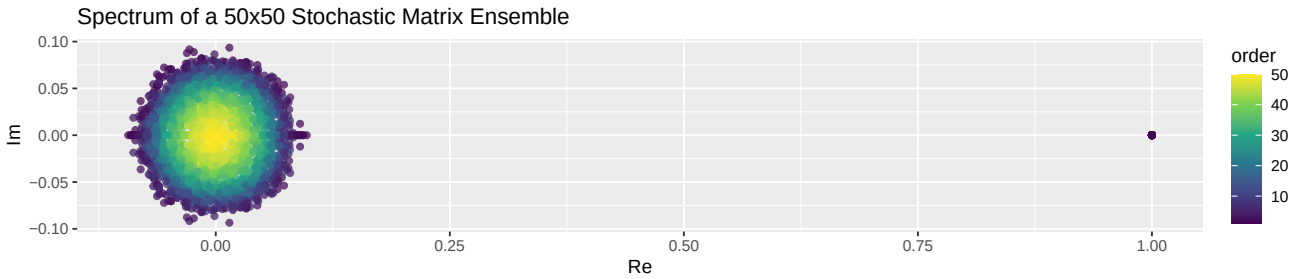


Figure 2.3: Spectrum of a Stochastic Matrix ensemble

demonstation of fact 2. Because the discussion of eigenvectors is too far afield from the current topic, so we will not include it in the current section.

From the simulations, we find that there is overwhelming evidence that most matrices, if not all, satisfy the second statement. The "almost all" part here is an artifact of random matrices, since it is unlikely they will have duplicate eigenvalues, or eigenvalues corresponding to pure rotations (with no scaling). These are the types of eigenvalues that lead to the second statement not holding.

Conclusions. So, to conclude, we can say that the computational evidence for these two facts support and provide an alternative method of empirically demonstrating that the Perron-Frobenius theorem is true.

2.3 Symmetric and Hermitian Matrices

A very important class of matrices in Linear Algebra is that of Symmetric or Hermitian matrices (See A.x). Simply put, those are matrices which are equal to their conjugate transpose.

Remark (Symmetric versus Hermitian)**.** *Since real numbers are their own conjugate transpose, every Symmetric matrix is Hermitian. However, we will still delineate the two terms to avoid confusion and indicate what field* $\mathbb{F}$ *we are working with.*

In any case, one critical result in Linear Algebra that will be extensively wielded in this thesis is the fact that if a matrix is Symmetric or Hermitian, then it has real eigenvalues. In other words:

$$P = \overline{P^T} \implies \sigma(P) = \{\lambda_i \mid \lambda_i \in \mathbb{R}\}$$

Having a complete set of real eigenvalues yields many great properties. For instance, if all eigenvalues are real, we have the option of observing either the sign-ordered spectrum or the norm-ordered spectrum. This way, we can preserve negative signs and we would not lose the rotational aspect of the eigenvalue when we study its statistics. That is just one reason out of many more why having real eigenvalues is quite nice.

Example: Stochastic Matrices

One very pleasing example to look at is stochastic matrices. As seen previously in **Subsection 2.2.2**, stochastic matrices tend to have two components: a complex disk of eigenvalues about the origin and isolated point that is the largest eigenvalue.

Below we have a **symmetric** stochastic ensemble spectrum. Notice how the largest eigenvalue is 1, living far from the island of **real** eigenvalues. This can be compared to the spectrum of non-Symmetric stochastic matrices in the previous figure. The patterns are similar, but now, imposing symmetry on the matrices forces the eigenvalues to "collapse" onto the real line!

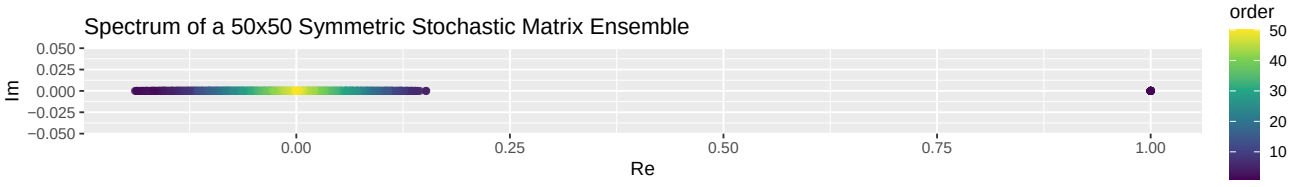


Figure 2.4: Spectrum of a Symmetric Stochastic Matrix ensemble

2.3.1 Wigner's Semicircle Distribution

The eigenvalues of Hermitian matrices obey Wigner's Semicircle distribution. Since Hermitian matrices have real eigenvalues, then we can be more precise and generally say that the real component of the eigenvalues follow the semicircle distribution. The distribution is named after physicist Eugene Wigner.

Definition 2.3.1 (Semicircle Distribtion)**.** *If a random variable X is semicircle distributed with radius $R \in \mathbb{R}^+$, then we say $X \sim SC(R)$. X has the following probability density function:*

$$\boldsymbol{P}(X = x) = \frac{2}{\pi R^2}\sqrt{R^2 - x^2} \text{ for } x \in [-R, R]$$

Remark (Radius and Matrix Dimension)**.** *The dimension of the matrix determines the radius of the eigenvalues. Namely, if a Hermitian matrix P is $N \times N$, then its eigenvalues are approximately semicircle distributed with radius $R = 2\sqrt{N}$; the approximation improves as N gets larger. That is, $P^\dagger$ has a spectrum $\sigma(P) \sim SC(R = 2\sqrt{N})$ as $N \to \infty$.*

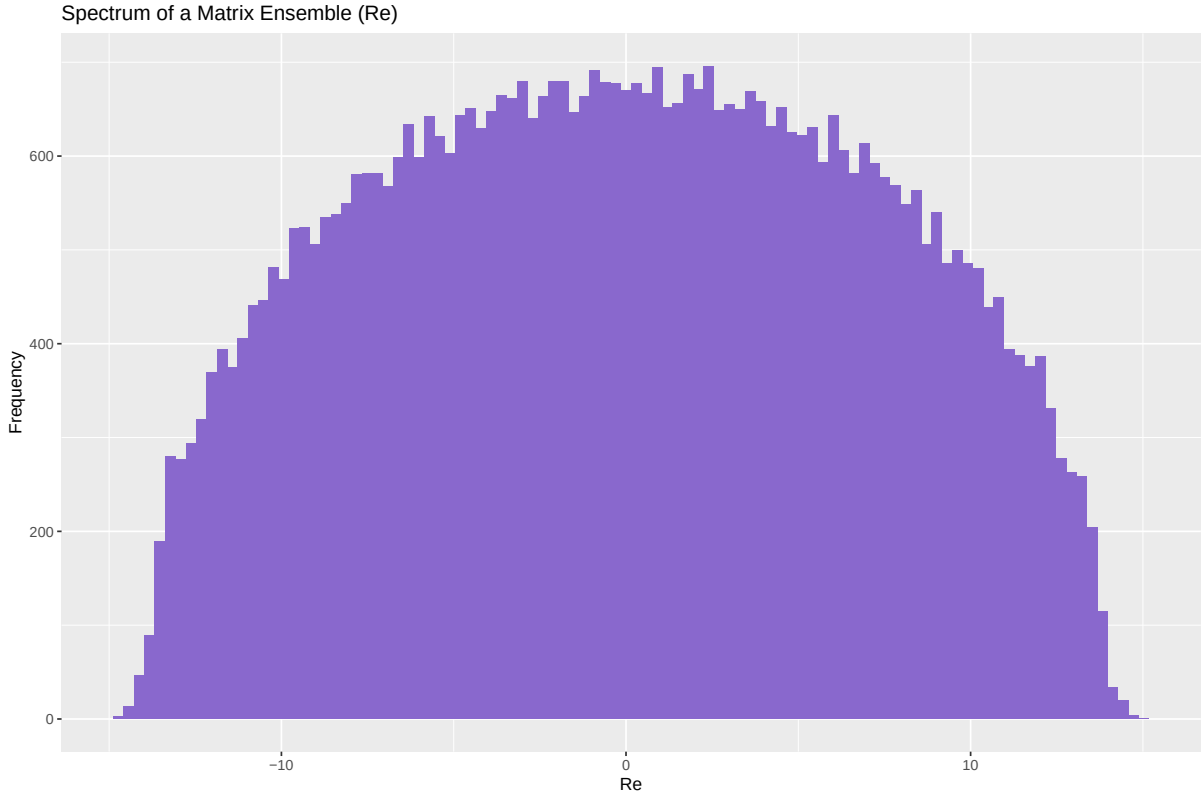


Figure 2.5: Eigenvalues of a Symmetric Matrix displaying the Semicircle Distribution

2.4 Findings

2.4.1 Uniform Ensembles

Here we have an ensemble of $\mathcal{E} \sim \text{Unif}(0, 1)$ matrices. We see a resemblance to stochastic matrices.

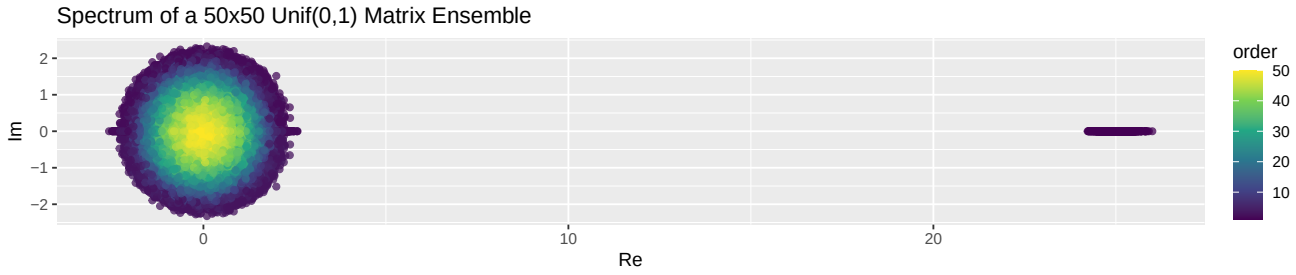


Figure 2.6: Spectrum of a Unif(0,1) ensemble

Plot takeaways

Largest eigenvalue is varied; compared with stochastic!

2.4.2 Erdos p-Ensembles

Here we have the second largest eigenvalue of the Erdos-Renyi ensemble. We parameterize the eigenvalue by p and we observe a trend below.

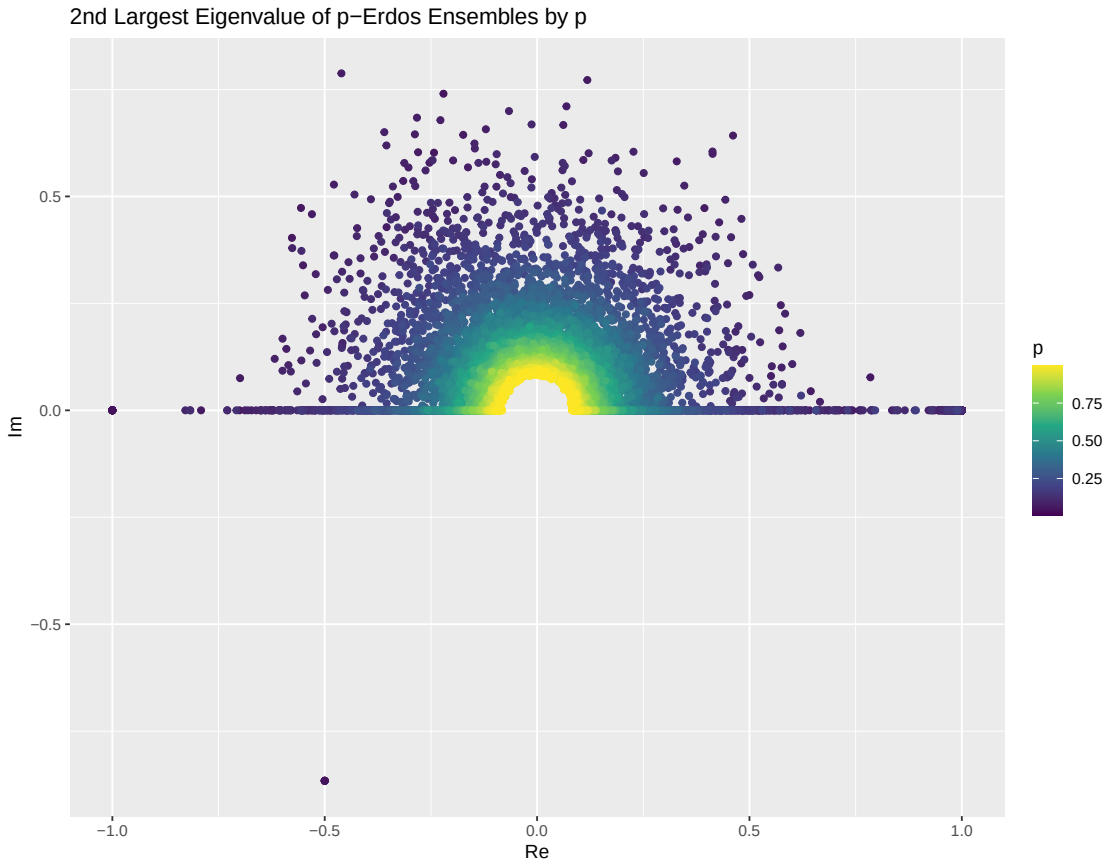


Figure 2.7: 2nd Largest Eigenvalue of an Erdos-p ensemble

Plot takeaways

Why upper half plane only?

2.4.3 Normal Matrices

Normal matrices tend to have eigenvalues distributed about the complex disk.

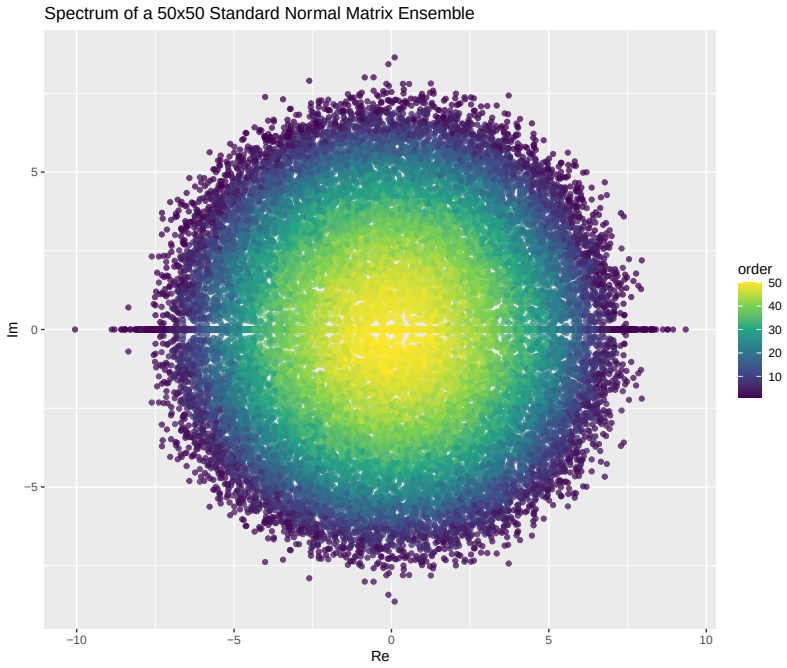


Figure 2.8: Spectrum of a Standard Normal Matrix ensemble

Hermitian Complex Normal matrices tend to have eigenvalues distributed about the complex disk.

Remark (Floating Point Errors)**.** *Because the model involves finding coefficients of polynomials with rounded entries (by computing the determinant), floating point errors and algorithmic systemic error will yield small, but negligible imaginary components.*

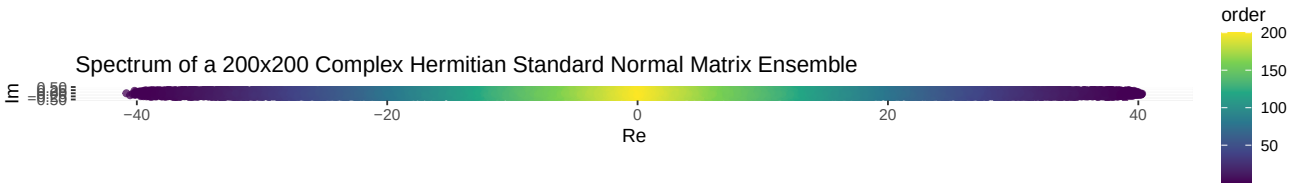


Figure 2.9: Spectrum of a Complex Hermitian Standard Normal Matrix ensemble

Chapter 3

Dispersions

> Distance doesn't exist, in fact, and neither does time. Vibrations from love or music can be felt everywhere, at all times.
>
> *Yoko Ono*

3.1 Introduction

In this section, we define the final spectral statistic studied in this chapter: eigenvalue dispersions. As the name suggests, these statistics are concerned with the distirbution of the spacings between the eigenvalues. Interestingly, this is almost as literal as it gets when we use the word "spectral". In physics and chemistry, atomic spectra are essentially differences between energy levels or quanta, so the translation is close.

In any case, we will begin this chapter by first motivating a few definitions and formalisms in this section. Then, once our setup is ready, we will motivate the definition of a matrix's eigenvalue dispersion and formalize it as a statistic. To outline the section, we will first define two things: the dispersion metric and the pairing scheme. In simple terms, we formalize **what** "eigenvalue spacings" are and **which** eigenvalue pairs' spacings to consider. So, to begin, we first formally define an eigenvalue pair.

Definition 3.1.1 (Eigenvalue Pair)**.** *Suppose P is a matrix and $\sigma(P)$ is its ordered spectrum. Then, an eigenvalue pair with respect to this ordered spectrum is denoted π_{ij}. It is defined as the ordered pair $\pi_{ij} = (\lambda_i, \lambda_j)$.*

Consecutive Pairs

In the future section where we define pairing schemes, we will introduce new notation of eigenvalue pair with one index is introduced when defining the consecutive pairing scheme Π_C, and it takes the form $\tilde{\pi}_j$. This notation is used to denote the consecutive eigenvalue pair for a given matrix. The consecutive eigenvalue pairs are

so special that we denote them with a unique notation for convenience. It also makes the discussion regarding order statistics more intrinisic.

Definition 3.1.2 (Consecutive Pairs)**.** *Suppose P is a matrix and $\sigma(P)$ is its ordered spectrum. Then, let $\tilde{\pi}_j$ denote a pair of consecutive eigenvalues, the largest of the two being the j^{th} largest eigenvalue. So, $\tilde{\pi}_j = (\lambda_{j-1}, \lambda_j)$.*

Since the consecutive pair scheme can be sufficiently indexed by one index (j), we will use the convention of omitting the index of the smaller eigenvalue to be more concise and idiomatic. So, whenever one sees the notation $\tilde{\pi}_j$, think the j^{th} largest eigenvalue and its smaller neighbour.

Example (The Largest Eigenvalues)**.** *With this notation at hand, we say that $\tilde{\pi}_1$ represents the pair of the two largest eigenvalues in an ordered spectrum.*

$$\tilde{\pi}_1 = (\lambda_2, \lambda_1)$$

Without further ado, we now motivate the dispersion metric.

3.1.1 Dispersion Metrics

Before we may even start to consider studying dispersions of eigenvalues, we must first formalize and make clear what "metric" of spacing we are using. To do so, we motivate the dispersion metric. In simple words, a dispersion metric is a function that takes in a pair of eigenvalues and returns a positive real number that represents some notion of dispersion or spacing.

Definition 3.1.3 (Dispersion Metric)**.** *A dispersion metric $\delta : \mathbb{C} \times \mathbb{C} \to \mathbb{R}^+$ is defined as a function from the space of pairs of complex numbers to the positive reals. In simple terms, it is a way of measuring "space" between two complex numbers - our eigenvalues.*

In the scope of this thesis, we consider the following dispersion metrics below.

1. The standard norm: $\delta(z, z') = |z' - z|$
2. The β-norm: $\delta_\beta(z, z') = |z' - z|^\beta$
3. The difference of absolutes: $\delta_{\text{abs}}(z, z') = |z'| - |z|$

Remark (Symmetric Metrics)**.** *Note that the standard norm and the β-norm are symmetric operations. This means that switching the order of arguments will have no effect on the function's output. Otherwise, the difference of absolutes metric is not symmetric.*

While we have defined disperison metrics to be functions from $\mathbb{C}^2$, there is one special case where we can make an exception so that the domain of δ is not $\mathbb{R}^+$.

Remark (Identity Difference Heuristic)**.** *Suppose we take the arithmetic difference of two complex numbers. Then, the range of δ is $\mathbb{C}$. For this reason, we won't consider the arithmetic difference a formal dispersion metric, but we will honor it as a dispersion huerestic. As such, we will denote this as the "identity difference" huerestic and call it δ_{id}. So, we define $\delta_{id} : (z, z') \mapsto z' - z$*

Summary Table of Dispersion Metrics

For every dispersion metric, assume the functions' order of arguments is $\delta(z, z')$.

Table of Dispersion Metrics				
Metric*	Notation	Formula	Symmetric	Parameters
Standard Norm	δ	$\lvert z' - z\rvert$	True	-
β-Norm	δ_β	$\lvert z' - z\rvert^\beta$	True	$\beta \in \mathbb{N}$
Difference of Absolutes	δ_{abs}	$\lvert z'\rvert - \lvert z\rvert$	False	-
Identity Difference	δ_{id}	$z' - z$	False	-

*Note that the identity difference is a heurestic, and not a formal metric.

3.1.2 Pairing Schema

The next thing we need to motivate before talking about eigenvalue dispersions are pairing schema. In simple terms, pairing schema are templates (for some $N \in \mathbb{N}$) for which eigenvalue pairs to pick. There are many subtle reasons why this is important, which will be covered in detail later. Without further ado, we motivate the pairing schema.

Before we may begin talking about pairing scheme, we define an auxilliary object, the spectral pairs of a matrix, which we denote $\sigma^{(2)}$. Since we are now talking about eigenvalue pairs, it is helpful to define this object before proceeding.

Definition 3.1.4 (Spectral Pairs)**.** *Suppose P is an $N \times N$ random matrix. Then, taking the spectral pair of P, denoted $\sigma^{(2)}(P)$ is equivalent to taking the Cartesian product of its ordered spectrum $\sigma(P)$. That is, $\sigma^{(2)}(P) = \sigma(P) \times \sigma(P) = \{(\lambda_i, \lambda_j) \mid i, j \in \mathbb{N}_N\}$.*

Now, to select eigenvalue pairs, we motivate the pairing scheme - this is what tells us which indices to select.

Definition 3.1.5 (Pairing Scheme)**.** *Suppose P is any $N \times N$ matrix and $\sigma^{(2)}(P)$ are its spectral pairs. A pairing scheme for the matrix P is a subset of $\mathbb{N}_N \times \mathbb{N}_N$ - a subset of pairs of numbers from $\mathbb{N}_N = \{1, \ldots, N\}$. In other words, it is a subset of pair indices for N objects - in our case, eigenvalues. We denote a pairing scheme as a set $\Pi = \{(\alpha, \beta) \mid \alpha, \beta \in \mathbb{N}_N\} \subseteq \mathbb{N}_N \times \mathbb{N}_N$. To take a matrix's spectral pairs with respect to Π, we simply take the set of eigenvalue pairs with the matching indices, $\sigma^{(2)}(P \mid \Pi) = \{(\lambda_\alpha, \lambda_\beta) \mid (\alpha, \beta) \in \Pi\}$.*

The reader might find this definition slightly obscure, and rightfully so. The definition is a mere formality, as we will usually only consider a few specific pairing schema. Seeing the explicit examples will hopefully make things more clear. With all the technical details aside, we can just say that a pairing scheme tells us which subset of eigenvalue pairs to consider. If one visualizes an array with N objects on two axes, we are simply choosing a subset of that plane. In fact, consider the following.

Remark (Proper Subset)**.** *If $\sigma^{(2)}(P)$ is the spectral pairs of the matrix P, then for any pairing scheme Π, we will find that $\sigma^{(2)}(P \mid \Pi) \subseteq \sigma^{(2)}(P)$. By definition, taking a spectral pair with respect to some pairing scheme constricts which pairs to select - meaning it is a proper subset of the general spectral pairs.*

Common Pairing Schema

Suppose P in an $N \times N$ square matrix, and $\sigma^{(2)}(P)$ are its spectral pairs. For every pairing scheme, assume $i, j \in \mathbb{N}_N$.

1. The unique pair combinations schema are two complementary pair schema. By specifying **either** $i > j$ or $i < j$, we characterize this scheme to entail all unique pair combinations of eigenvalues without repeats. The reason we call them upper and lower pair combinations is an allusion to the indices of the upper and lower triangular matrices.

   (a) Let $\Pi_>$ be the lower-pair combinations of ordered eigenvalues. This will be the standard unique pair combination scheme used in lieu of the argument orders of our dispersion metrics (more later). In this pairing scheme, the eigenvalue with the lower rank is always listed first, and the higher rank second.
   $$\sigma^{(2)}(P \mid \Pi_>) = \{\pi_{ij} = (\lambda_i, \lambda_j) \mid i > j\}_{i=1}^{N-1}$$

   (b) For completeness, we will also define the upper-pair scheme. $\Pi_<$ is the set of upper-pair unique combinations of ordered eigenvalues. Wont be used because we want bigger - smaller to make positive definite.
   $$\sigma^{(2)}(P \mid \Pi_<) = \{\pi_{ij} = (\lambda_i, \lambda_j) \mid i < j\}_{i=1}^{N-1}$$

   Benefits: Solves the issue of repeated pairs for symmetric dispersion metrics.

2. Let Π_C be the consecutive pairs of eigenvalues in a spectrum.
   $$\sigma^{(2)}(P \mid \Pi_C) = \{\tilde{\pi}_j = (\lambda_{j+1}, \lambda_j)\}_{j=1}^{N-1}$$

   Benefits: This pairing scheme gives us the minimal information needed to express important bounds and spacings in terms of its elements.

3. Let Π_0 be all the pairs in a spectrum. For completeness, we define this pairing scheme as the implicit pairing scheme for the spectral pairs of a matrix.
   $$\sigma^{(2)}(P \mid \Pi_0) = \sigma^{(2)}(P) = \{\pi_{ij} = (\lambda_i, \lambda_j) \mid i, j \in \mathbb{N}_N\}$$

Consider some examples of generating pairing schemes in R. These methods are not exported in the **RMAT** package.

Code Example (Consecutive Pairing Scheme)**.** *We generate the pairing scheme (indices) for what would be a* 5×5 *matrix.*

```
# Helper function in the source code
pair_indices <- .consecutive_pairs(N = 5)
# Outputs the following
pair_indices
...
     i j
[1,] 2 1
[2,] 3 2
[3,] 4 3
[4,] 5 4
```

Code Example (Lower Pairing Scheme)**.** *We generate the pairing scheme (indices) for what would be a* 4×4 *matrix.*

```
# Helper function in the source code
pair_indices <- .unique_pairs_lower(N = 4)
# Outputs the following
pair_indices
...
     i j
[1,] 2 1
[2,] 3 1
[3,] 3 2
[4,] 4 1
[5,] 4 2
[6,] 4 3
```

Think of the pairing schemes as a conditional argument when considering dispersions. You ask yourself: which pairs do I want to look at? There will be a discussion in **Section 3.2** about why we have to formalize the notions of dispersion metrics and pairing schemes when we eventually talk about dispersions.

Additionally, making the pairing schemes independent from the matrices (taking only their dimension, $N \in \mathbb{N}$) was a difficult decision. It may cloud understanding initally but the payoff comes if you consider the implementation of the functions in **RMAT**. As a whole, you should not be bothered by the details of formalizing the pairing schemes but you should definitely understand what they do. In fact, it's totally fair that you do. The **RMAT** implementation abstracts this all away in the dispersion function. The user would input a string denoting which pairing scheme and the function would handle it. Similarly, you should do the same!

Summary Table of Pairing Schema

Suppose P is an $N \times N$ matrix. Then, its spectral pairs $\sigma^{(2)}(P)$ may take on the following pairing schemes.

Table of Pairing Schema		
Scheme	Notation	Formula
Lower	$\Pi_<$	$\{(i,j) \mid i < j \text{ for } i,j \in \mathbb{N}_N\}$
Upper	$\Pi_>$	$\{(i,j) \mid i > j \text{ for } i,j \in \mathbb{N}_N\}$
Consecutive	Π_C	$\{(i,j) \mid i = j+1 \text{ for } i,j \in \mathbb{N}_N\}$
All	Π_0	$\{(i,j) \mid i,j \in \mathbb{N}_N\}$

Another way we can define the pairing scheme is defining an auxilliary object: the eigenvalue (pair) matrix.

An Alternative Represetation: Eigenvalue Matrix

Definition 3.1.6 (Eigenvalue Matrix)**.** *Suppose P is an $N \times N$ square matrix and $\sigma^{(2)}(P)$ are its spectral pairs. Then, the eigenvalue matrix of P, given by $\Lambda(P)$ is the matrix with entries $\pi_{ij} = (\lambda_i, \lambda_j)$ for $\pi_{ij} \in \sigma^{(2)}(P)$. Again, it is given that the eigenvalues λ_i come from some **ordered** spectrum $\sigma(P)$ as per the definition of spectral pairs.*

With the matrix analogy, describing the pairing schemes can be much simpler. For instance, the **lower pairs** are given by the indices of the entries in the lower triangle of the matrix. Analogously, the **upper pairs** by those of the upper triangle of the matrix. The **consecutive pairs**, given our default lower pair scheme, are given by the lower main off-diagonal band of the matrix.

Example (Eigenvalue Matrix for a 5×5 Matrix)**.** *Suppose we have a 5×5 matrix P and its corrosponding spectral pairs $\sigma^{(2)}(P)$. Then, its eigenvalue matrix has the following strucutre:*

$$\begin{bmatrix} - & \pi_{12} & \pi_{13} & \pi_{14} & \pi_{15} \\ \tilde{\pi}_1 & - & \pi_{23} & \pi_{24} & \pi_{25} \\ \pi_{31} & \tilde{\pi}_2 & - & \pi_{34} & \pi_{35} \\ \pi_{41} & \pi_{42} & \tilde{\pi}_3 & - & \pi_{45} \\ \pi_{51} & \pi_{52} & \pi_{53} & \tilde{\pi}_4 & - \end{bmatrix}$$

3.1.3 Dispersions

Now, with dispersion metrics and pairing schemes defined, we are finally able to motivate the definition of a matrix dispersion for both singleton matrices and their ensemble counterparts.

Definition 3.1.7 (Dispersion). *Suppose P is an $N \times N$ matrix, and $\sigma^{(2)}(P)$ are its spectral pairs. The dispersion of P with respect to the pairing scheme Π and dispersion metric δ is denoted by $\Delta_\delta(P \mid \Pi)$ and it is given by the following:*

$$\Delta_\delta(P \mid \Pi) = \{\delta(\pi_{ij}) \mid \pi_{ij} \in \sigma^{(2)}(P \mid \Pi)\}$$

Consider the following code example, generating the dispersion of standard normal 5×5 matrix with respect to the consecutive pairing scheme.

Code Example (Consecutive Pair Dispersion of a Standard Normal Matrix). *In our notation, we are simulating the disperison of $P \sim \mathcal{N}(0, 1)$ where $P \in \mathbb{R}^{5\times5}$. Specifically, we are simulating $\Delta(P \mid \Pi_C)$. In the code implementation, we obtain an array for every dispersion metric δ.*

```
library(RMAT)
P <- RM_norm(N = 5, mean = 0, sd = 1)
disp_P <- dispersion(P, pairs = "consecutive")
# Outputs the following
disp_P
...
i  j  eig_i        eig_j        id_diff     iddiff_norm abs_diff diff_ij
2  1  -0.54-1.35i  -0.54+1.35i   0.00+2.71i 2.71          0.00     1
3  2   0.23+1.43i  -0.54-1.35i  -0.77-2.78i 2.88          0.02     1
4  3   0.23-1.43i   0.23+1.43i   0.00+2.85i 2.85          0.00     1
5  4  -0.87+0.00i   0.23-1.43i   1.09-1.43i 1.80          0.57     1
```

Next, we extend the definition of dispersion for an ensemble as we usually do.

Definition 3.1.8 (Ensemble Dispersion). *If we have an ensemble $\mathcal{E}$, then we can naturally extend the definition of $\Delta_\delta(\mathcal{E} \mid \Pi)$. To take the dispersion of an ensemble, simply take the union of the dispersions of each of its matrices. In other words, if $\mathcal{E} = \{P_i \sim \mathcal{D}\}_{i=1}^K$, then its dispersion is given by:*

$$\Delta_\delta(\mathcal{E} \mid \Pi) = \bigcup_{i=1}^{K} \Delta_\delta(P_i \mid \Pi)$$

Consider the following code example, generating the dispersion of a beta ensemble with respect to the consecutive pairing scheme.

Code Example (Consecutive Pair Dispersions of a Beta Ensemble). *In our notation, we are simulating the disperison of $\mathcal{E} \sim \mathcal{H}(\beta = 4)$ where $P \in \mathbb{R}^{5\times5}$. Specifically, we are simulating $\Delta(P \mid \Pi_C)$. In the code implementation, we obtain an array for every dispersion metric δ.*

```
library(RMAT)
ens <- RME_beta(N = 4, beta = 4, size = 3)
disp_ens <- dispersion(ens, pairs = "consecutive")
# Outputs the following
```

```
disp_ens
...
i j eig_i     eig_j     id_diff  iddf_norm abs_diff diff_ij
2 1 -3.78+0i 4.00+0i   7.78+0i 7.78       0.22      1
3 2 2.06+0i -3.78+0i  -5.84+0i 5.84       1.72      1
4 3 0.19+0i   2.06+0i   1.88+0i 1.88       1.88      1
2 1 3.80+0i -4.00+0i  -7.80+0i 7.80       0.20      1
3 2 -1.80+0i 3.80+0i   5.60+0i 5.60       2.00      1
4 3 0.89+0i -1.80+0i  -2.69+0i 2.69       0.92      1
2 1 3.51+0i -3.53+0i  -7.04+0i 7.04       0.03      1
3 2 1.35+0i   3.51+0i   2.16+0i 2.16       2.16      1
4 3 -0.67+0i 1.35+0i   2.02+0i 2.02       0.68      1
```

3.2 Dispersion Analysis

With dispersions well defined, we provide a few guidelines for analyzing a dispersion of a matrix. We will synthesize all the techniques, notations, and remarks in the previous section to provide a comprehensive manual on how to analyze the dispersion of a matrix.

3.2.1 Considerations

Here are some considerations that should be taken into account when studying dispersions.

Dispersion metrics, linear combinations, and the relation to pairing schemes.

Bounds and Considerations

One thing to consider from this dispersion set is what we could learn from it. The dispersion of a matrix P, $\Delta_d(P)$ could tell us quite a few things. One thing it could tell us is the distribution of the dispersion of eigenvalues given that we uniformly and randomly select a pair of eigenvalues; this works when d is a symmetric function. However, if we are considered about spacings, then we must taking into consideration a few facts.

Linear Combinations

The identity difference metric is not necessarily a "bad metric". In fact, it is one of the few metrics which allow transitivity under composition - which brings up an important detail. In the total dispersion $\Delta_d(P)$, there is some redundant information if we are totally concerned with spacings. Namely, consider the following fact:

$$\lambda_1 - \lambda_3 = (\lambda_1 - \lambda_2) - (\lambda_3 - \lambda_2)$$

In other words, if d is the ID dispersion metric, then $d(\pi_{13}) = d(\pi_{12}) - d(\pi_{32})$. We could say the second eigenvalue is a pivot in which we could perform right-cancellation.

Triangle Inequality

However, as we said previously, only the identity difference metric gives us transitive properties. On the other hand, the other metrics provide us bounds when we compose them. They are given by a simple application of the reverse triangle inequality. Namely, consider the following fact:

$$||y| - |x|| \leq |y - x|$$

If we let d_1 be the difference of absolutes metric and d_2 be the standard norm, then we could rephrase this as saying $|d_1(\pi_{ij})| \leq d_2(\pi_{ij})$. This is just a rephrasing of the inequality. However, if we are careful enough, we can remove the absolute value to obtain a more meaningful upper bound. The value of $d_1(\pi_{ij})$ can be interpreted as the difference between the sizes of λ_j and λ_i. However, recall that in the standard definition of a dispersion, we use the **lower** pair combinations so that we get $i > j$, making the value $|\lambda_j|$ always greater than $|\lambda_i|$ by construction. As such, the left-hand value would always be positive, allowing us to drop the absolute value. So, we could say:

$$d_1(\pi_{ij}) \leq d_2(\pi_{ij}) \text{ given } \pi_{ij} \in \Pi_{>}$$

Sufficiency of Consecutive Pairs

Recall from previously that with respect to the identity difference, some elements of the dispersion $\Delta_d(P)$ can be expressed as a linear combination of other elements. Note that using those linear combinations and the triangle inequality, we could derive the analogous bounds. It is quite useful to impose a condition that none of the values are a linear combination of the other.

So, we motivate a more restrictive definition of matrix dispersion: the consecutive-values dispersion $\Delta_d(P, \Pi_c)$. This dispersion is essentially a special case of the regular dispersion, but with a particular pairing scheme Π_c. To avoid this issue of linear combinations, we simply take the pairs of consecutive eigenvalues. In other words, $\Pi_c = \{(\lambda_i, \lambda_j) \in \sigma_P \times \sigma_P \mid j = i + 1\}_{i=1}^{N-1}$. This pairing scheme enumerates the pairs such that the dispersion metric has a natural mapping - it is simply the "distance" between an eigenvalue and the one that is one rank smaller than it.

3.2.2 Order Statistics

With eigenvalue dispersions and eigenvalue orderings well-defined, we may proceed to start talking about their order statistics.

In **Chapter 2**, we observed simple order statistics regarding the eigenvalues. This time around, we have two eigenvalues being compared at once, so there needs to be further setup with regards to dispersion order statistics.

However, prior to introducing any new concepts, there is one scenario where using one index is sufficient in terms of discussing dispersion order statistics.

Remark (Consecutive Pair Dispersions)**.** *When we consider the dispersion with repsect to the consecutive pairs, things are much simpler since our pairs are intrinsically defined by one index (j). This is because we are observing the j^{th} eigenvalue and its smaller neighbour. As such, we will consider dispersion statistics in the form of $\mathbb{E}(\tilde{\pi}_j \mid j)$ and $Var(\tilde{\pi}_j \mid j)$.*

Otherwise, we synthesize a new order statistic that takes in two indices (from a given pair) called the **ranking difference class**. Since we are no longer observing a single eigenvalue at a given rank, we will need a way to standardize observing a pair of eigenvalues at a time. To do so, we introduce a new **equivalence relation** (see A.x) called the **ranking difference**. As the name suggests, it is precisely the integer difference of the eigenvalue orders.

Ranking Difference

Definition 3.2.1 (Ranking Difference)**.** *The ranking difference is a function $\rho : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ which takes the index of two ordered eigenvalues and returns their difference. In other words, it is the function $\rho : (\lambda_i, \lambda_j) \mapsto (i - j)$.*

With the ranking difference, we may now take the spectral pairs of some random matrix with respect to either the lower ($\Pi_<$) or upper ($\Pi_>$) pairing scheme and partition the eigenvalues into distinct equivalence classes.

Remark (Equivalence Relation)**.** *Formally speaking, we may define $\sim_\rho$ as an equivalence relation. It satsifies all three properties of reflexivity, symmetry, and transitivity. More precisely, it is an equivalence class which partitions $\mathbb{N}_N \times \mathbb{N}_N$ into $N - 1$ equivalence classes given by $[r]_\rho$ for $r \in \mathbb{N}_{N-1}$. We would define the equivalence relation $\sim_\rho$ as follows:*

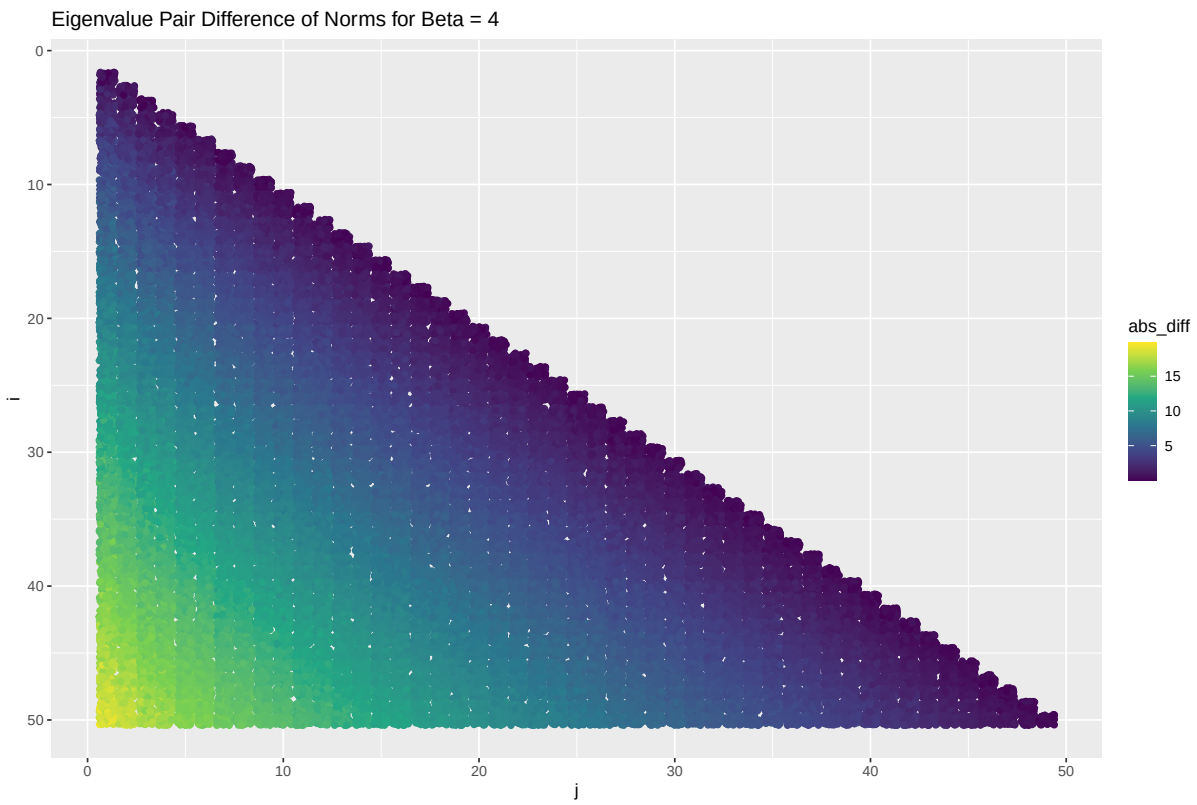
$$(\lambda_a, \lambda_b) \sim_\rho (\lambda_c, \lambda_d) \iff (a - b) = (c - d)$$

So, two eigenvalue pairs would belong to the same class $[r]_\rho$ if their ranking difference is the same. That is,

$$[r]_\rho = \{(\lambda_\alpha, \lambda_\beta) \mid \alpha - \beta = r\}$$

Remark (Matrix Analogy)**.** *Using the eigenvalue matrix analogy, $\sim_\rho$ partitions the matrix into diagonal bands. The diagonal represents $[0]_\rho$, the first off-diagonal represents $[1]_\rho$, and so on... Notice that this means every equivalence class has a different size.*

[Plot takeaways]



With this partition in mind, we can consider various statistics conditioning on the value of r. Conditioning on r will be especially useful in the cases where we are considering matrices like the Hermite-β matrices; the eigenvalues of those matrices tend to *repel*, so to speak, and we can observe these patterns using r.

Consider the following example.

Example (Ranking Differences for a 4×4 Matrix)**.** *Suppose P is a 4×4 random matrix. Then, if we take the spectral pairs with respect to the lower pairs $\sigma^{(2)}(P \mid \Pi_<)$ and partition it by ρ, we get the following classes:*

1. $[1]_\rho = \{\tilde{\pi}_1, \tilde{\pi}_2, \tilde{\pi}_3\}$
2. $[2]_\rho = \{\pi_{31}, \pi_{42}\}$
3. $[3]_\rho = \{\pi_{41}\}$

3.2.3 Conditional Statistics

We will considering the conditional statistics $\mathbb{E}(\rho_{ij} \mid \rho)$ and $\mathrm{Var}(\rho_{ij} \mid \rho)$.

[Plots]

3.3 Analytical Results

3.3.1 Wigner's Surmise

Wigner's surmise is a result found by Eugene Wigner regarding the limiting distribution of eigenvalue spacings of for symmetric matrices. To start talking about this, we must talk about normalized spacings, which are the precise items considered in the distribution. Before, we can talk about the normalized spacing, we define the mean spacing.

Definition 3.3.1 (Mean Spacing)**.** *Suppose P is an $N \times N$ symmetric matrix, and $\sigma(P)$ are its real, sign-ordered eigenvalues. Then, the mean (eigenvalue) spacing, denoted $\langle s \rangle$ is the average distance between two consecutive eigenvalues. That is,*

$$\langle s \rangle = \mathbb{E}[\Delta_\delta(P \mid \Pi_C)] = \mathbb{E}[\delta(\tilde{\pi}_j)]_{j=1}^{N-1}$$

So, with the mean spacing defined, we now define the normalized spacing between a pair of consecutive eigenvalues below.

Definition 3.3.2 (Normalized Spacing)**.** *Suppose P is an $N \times N$ symmetric matrix, and $\sigma(P)$ are its real, sign-ordered eigenvalues. Then, the normalized spacing of the j^{th} pair of eigenvalues, denoted s_j is given by the following formula.*

$$s_j = \frac{(\lambda_j - \lambda_{j+1})}{\langle s \rangle} = \frac{\delta(\tilde{\pi}_j)}{\langle s \rangle}$$

Beta Ensembles

The analytical results for the Hermite β-ensembles for $\beta = 1, 2, 4$, denoted by w_β, are stated below. These are the asymptotic densities for the normalized spacings between consecutive eigenvalues for each of the matrices for $\beta = 1, 2, 4$.

$$w_1(s) = \frac{\pi}{2} \exp\left(-\frac{\pi}{4}s^2\right)$$
$$w_2(s) = \frac{32}{\pi^2}s^2 \exp\left(-\frac{4}{\pi}s^2\right)$$
$$w_4(s) = \frac{2^{18}}{3^6\pi^3}s^4 \exp\left(-\frac{64}{9\pi}s^2\right)$$

Additionally, their simulated densities are given in the plot below.

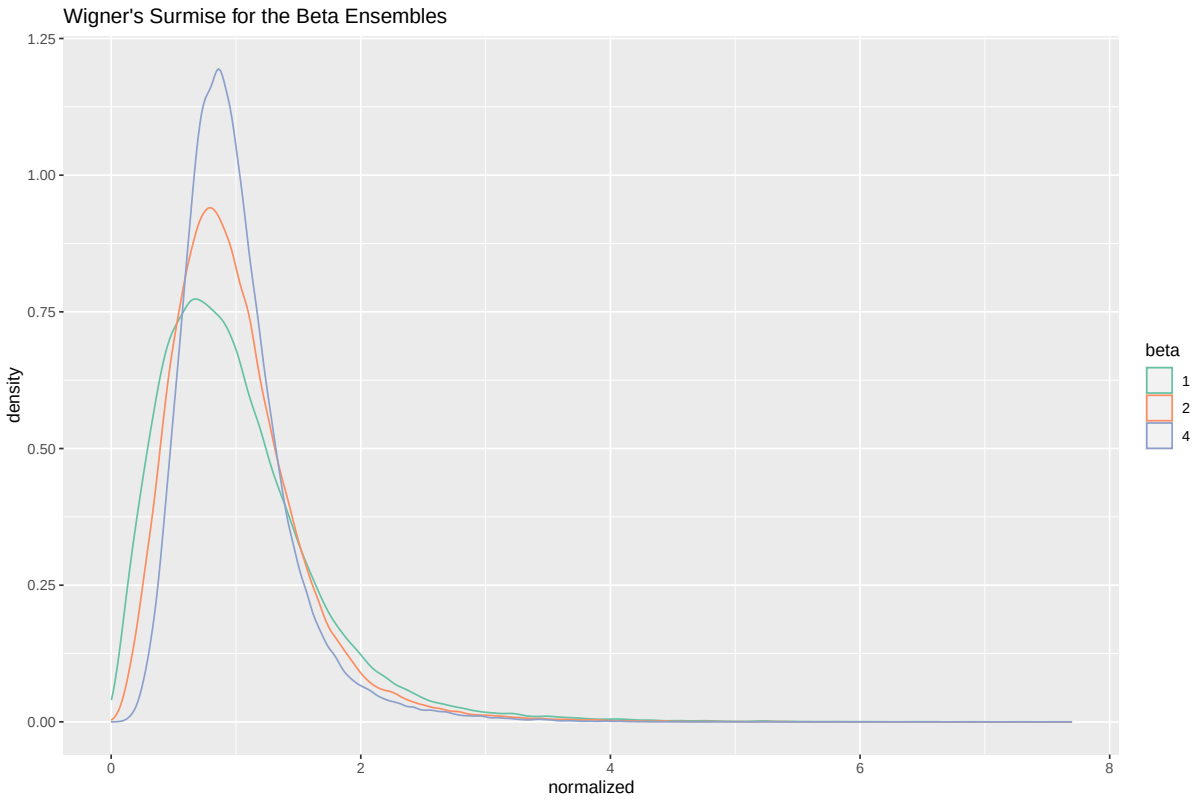


Figure 3.1: Wigner's Surmise for standard Beta Ensembles

Plot Takeaways

In 1 or 2 paragraphs, describe the most important observations you want your reader to take away from this plot

Extending Beta Ensembles

Below, we extended the simulations beyond the standard values of $\beta = 1, 2, 4$. We notice a continuing pattern! The densities seem to have less variance as beta increases.

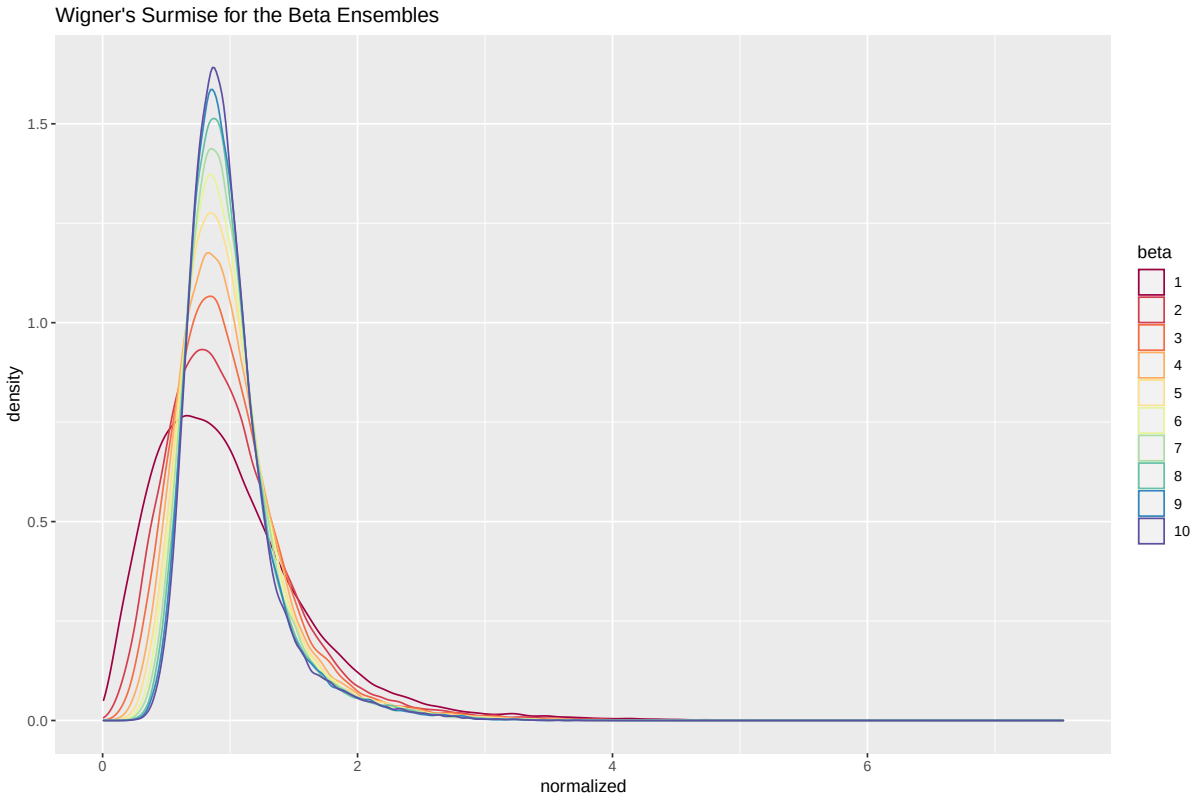


Figure 3.2: Wigner's Surmise for non-standard Beta Ensembles

Plot Takeaways

In 1 or 2 paragraphs, describe the most important observations you want your reader to take away from this plot

3.4 Checkpoint

Here's what we've done so far.

Chapter 1

1. Formalized the following items
   (a) Random Matrices using $\mathcal{D}$-distributions.
   (b) Random Matrix Ensembles
2. Some item

Chapter 2

1. Formalized the following items
   (a) Spectrum of a Random Matrix (Ensemble)
2. Some item

Chapter 3

1. Formalized the following items
   (a) Spectral Pairs: the set of eigenvalue pair combinations
   (b) Pair schemes: which eigenvalue pairs to choose
   (c) Dispersion metrics: what distance metric to use on the pairs
   (d) Dispersion of a Random Matrix (Ensemble)
2. Some item

Segway to Chapter 4

Here's how we'll segway to Chapter 4, which doesn't flow without this segway.

Chapter 4

β-Ensembles

> With thermodynamics, one can calculate almost everything crudely; with kinetic theory, one can calculate fewer things, but more accurately; and with statistical mechanics, one can calculate almost nothing exactly.
>
> *Eugene Wigner*

4.1 Introduction

In this chapter, we will talk about the Hermite β-ensembles more in depth. The beta ensembles have wide applications in statistical physics, eningeering, and many other places. They are defined by the joint density of their eigenvalues, and have a special characterization discussed in the next subsection.

4.1.1 Hermite β-Ensembles

The Hermite β-ensembles, also called the Gaussian ensembles, are an important class of random matrix ensembles studied in engineering, statistical physics, and probability theory. Parameterized by $\beta \in \mathbb{N}$ through the Dyson index, this ensemble is charecterized by a few things.

- The Dyson index β corrosponds to the number of real number of components the subject matrices have.
- The subject matrices are classically defined for $\beta = 1, 2, 4$ and they corrospond to matrices with real, complex, and quaternionic entries. The corrosponding fields are $\mathbb{R}$, $\mathbb{C}$, and $\mathbb{H}$.

- The matrices in this ensemble, most importantly, have a feature called conjugation invariance. With respect to the conjugation by the respective group of matrices.
- Most importantly, the eigenvalues are determined by the joint probability density function given below.

Definition 4.1.1 (Hermite β-ensembles)**.** *The Hermite β-ensembles, commonly known as the Gaussian ensembles, are an ensemble of random matrices parameterized by β, and their eigenvalues have the joint probability density function:*

$$f_\beta(\Lambda) = c_H^\beta \prod_{i<j} |\lambda_i - \lambda_j|^\beta e^{-1/2 \sum_i \lambda_i^2}$$

where the normalization constant c_H^β is given by:

$$c_H^\beta = (2\pi)^{-n/2} \prod_{j=1}^{n} \frac{\Gamma(1+\beta/2)}{\Gamma(1+\beta j/2)}$$

To simulate matrices from the β-ensemble, we will be using a recent result published in "Matrix Models for Beta Ensembles" Dumitriu & Edelman (2018). This makes the β-ensemble a canonical example of an implicity distributed matrix; we do not care about the actual distribution of the entries, but rather the effect they have on the eigenvalues (trace) of the matrices. The algorithm used is directly cited from the results of Dumitriu's paper, and can be found in Appendix B.X.

Code Example (Hermite Beta = 2 Ensemble)**.** *Let $\mathcal{D} = \mathcal{H}(\beta = 2)$. We can generate $\mathcal{E} \sim \mathcal{D}$, an ensemble of 4×4 Hermite matrices ($\beta = 2$) of size 10 as such:*

```
library(RMAT)
ensemble <- RME_beta(N = 4, beta = 2, size = 10)
# Outputs the following
ensemble
...
[[10]]
           [,1]       [,2]         [,3]     [,4]
[1,] 0.7246302 1.8893868  0.00000000 0.000000
[2,] 1.8893868 1.5278221  0.68840045 0.000000
[3,] 0.0000000 0.6884004 -0.03876104 1.944495
[4,] 0.0000000 0.0000000  1.94449533 1.042741
```

Dumitriu's Matrix Model

To generate Hermite β matrices, we consider the result given in the Matrix Models of Beta Ensembles paper Dumitriu & Edelman (2018). We obtain the following algorithm.

Algorithm 4.1.1 (Dumitriu's Beta Matrix)**.**

1. *To simulate an $N \times N$ beta matrix, fix $N \in \mathbb{N}$.*
2. *Start by taking a diagonal of $\mathcal{N}(0, 2)$ variables.*
3. *Set both of the nearest off-diagonals to the row that samples from a $\chi(df = c_j)$ where $c_j = \beta \cdot j$ for columns spanning $j = 1, \ldots, n-1$.*

4.1.2 A Physical Interpretation

4.2 Spectra

We know that β-matrices must be symmetric. So, their eigenvalues must be real. Any imaginary component we observe is simply computational error, and we may safely ignore it. It is good to see that the error is uniform and small.

Remark (Implicit Distribution)**.** *Note that the beta ensemble is an ensemble characterized by some joint density function on its eigenvalues. So, this is a specific instance of an implcitly distributed matrix. There are many things to consider about* ***identifiability*** *that are subtle, but important. Recall that in our formalization of a specturm as a formal statistic, we vectorized the matrix then tooks the determinant of the characteristic polynomial that came about it. For this reason, we can see that while Dumitriu's model provides an explicit formula, there is an issue of identifiability. That is, given some eigenvalues, there is no injective function to the characteristic polynomial that produced it. Rather, there are infinitly many equivalence classes of characteristic polynomials (and such, random matrices) that surjectively produce a given multiset of eigenvalues.*

Remark (Floating Point Errors)**.** *Because the model involves diving by $\sqrt{2}$, floating point errors and algorithmic systemic error will yield small, but negligible imaginary components. This is on top of the floating errors from rounding, as discussed in* ***Section 2.3****.*

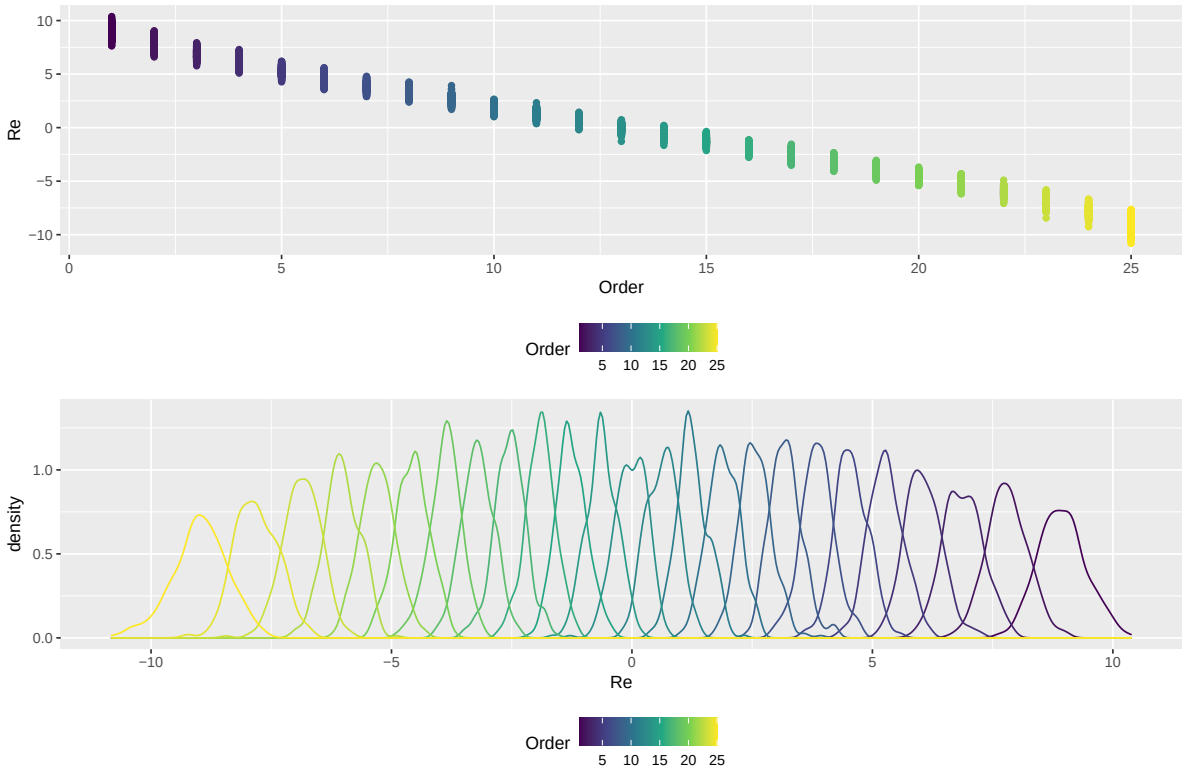


Figure 4.1: Wigner's Surmise for non-standard Beta Ensembles

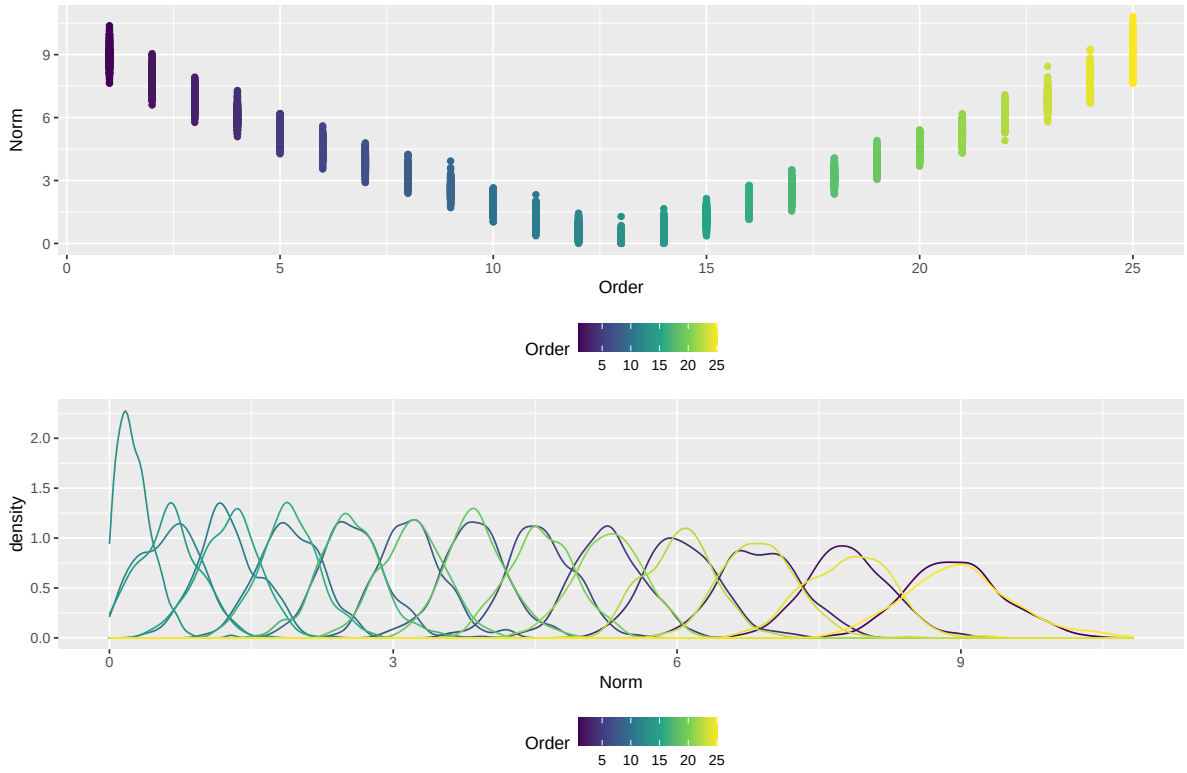


Figure 4.2: Wigner's Surmise for non-standard Beta Ensembles

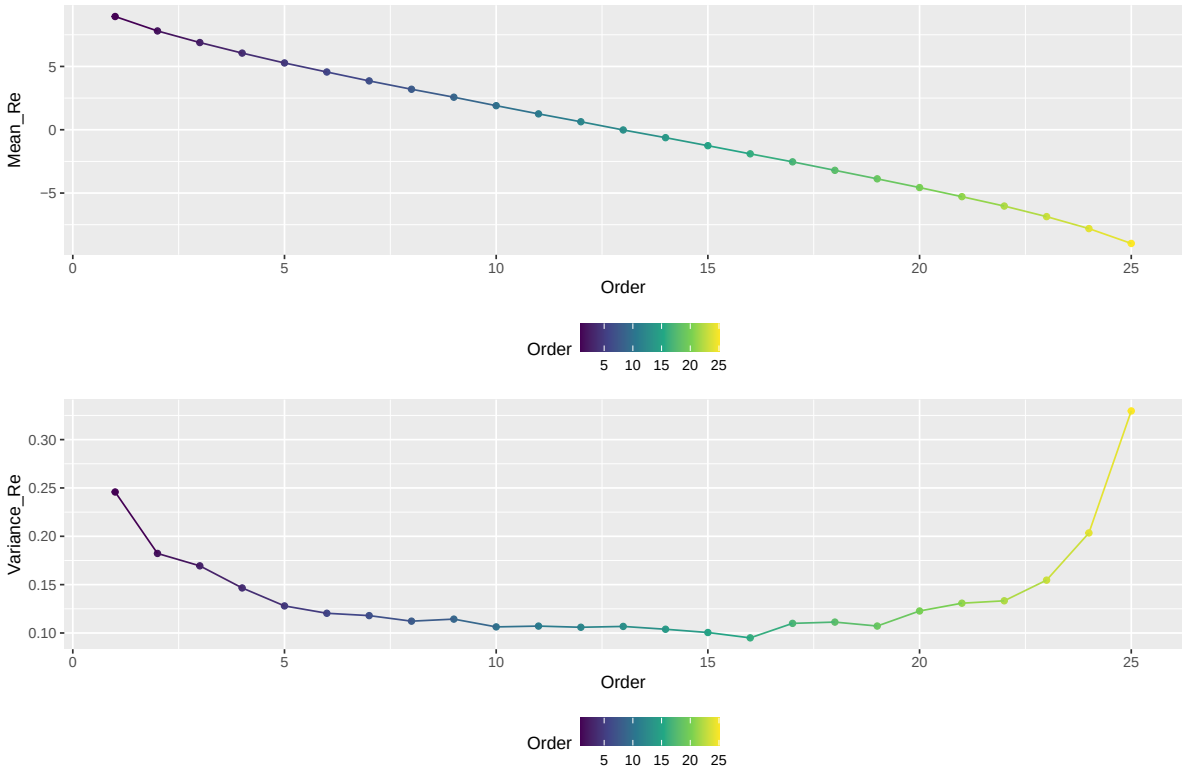


Figure 4.3: Wigner's Surmise for non-standard Beta Ensembles

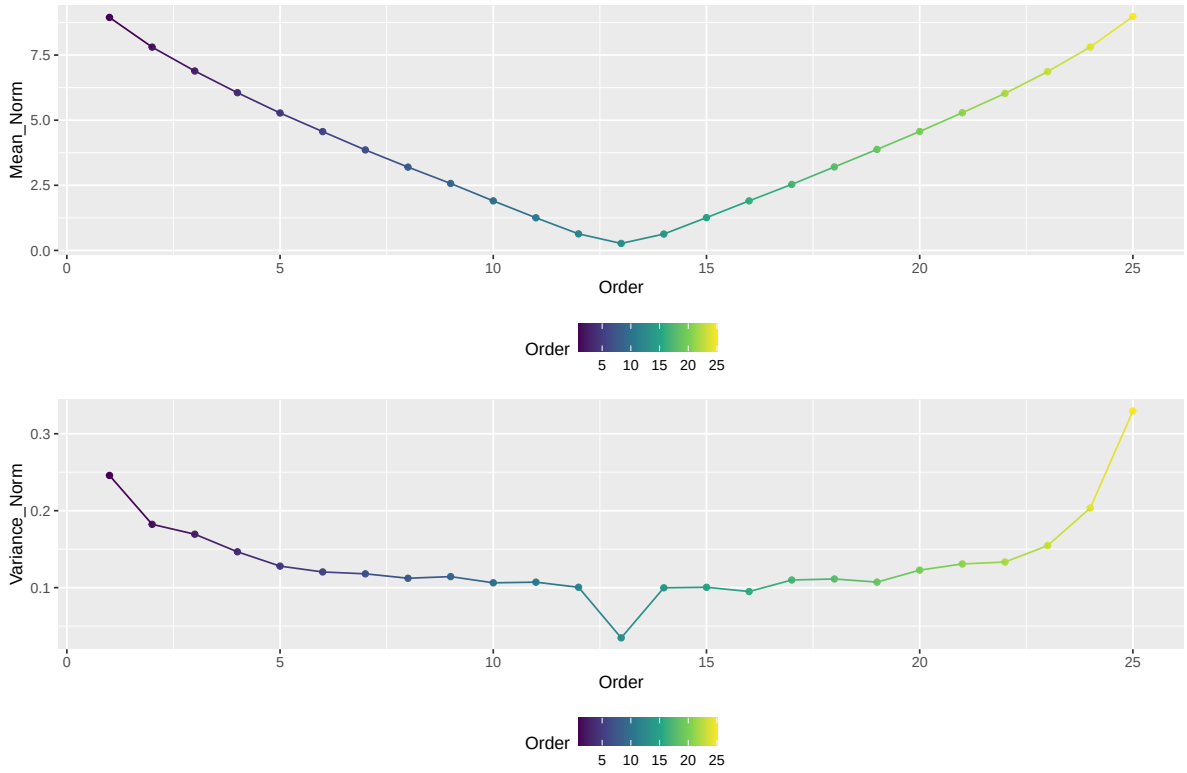


Figure 4.4: Wigner's Surmise for non-standard Beta Ensembles

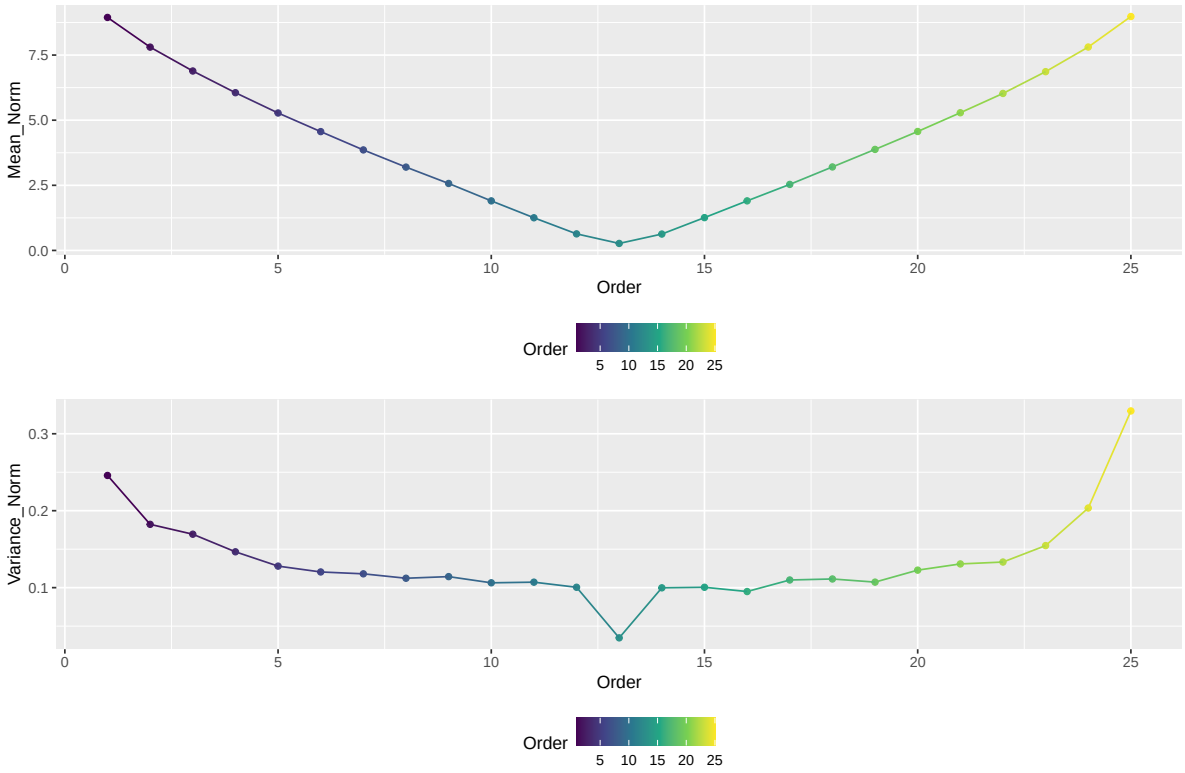


Figure 4.5: Wigner's Surmise for non-standard Beta Ensembles

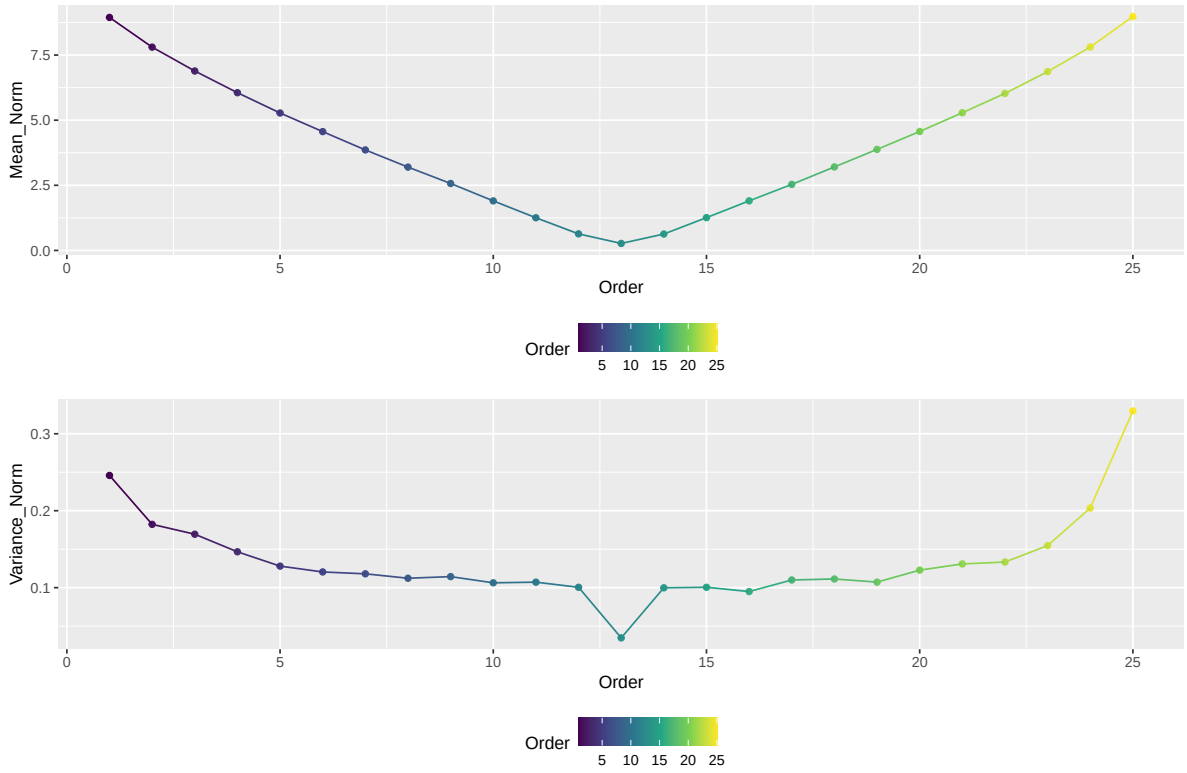


Figure 4.6: Wigner's Surmise for non-standard Beta Ensembles

4.3 Dispersions

Consider the following plot of the sign-sorted eigenvalue dispersions below.

Appendix A

Math Review

In the first section of this appendix, we list a few items in Linear Algebra worth reviewing to make sure there are no gaps in knowledge necessary to understand concepts in this thesis. Additionally, there is another subsection including a proof that was written in the research process of this thesis. Albeit this proof has no direct relation to the primary topic, spectral statistics, it is loosely relevant and displays the tools and techniques relevant to the material.

A.1 Linear Algebra

A.1.1 Matrices

Definition A.1.1 (Eigenvalue)**.** *Suppose $P \in \mathbb{F}^{n\times n}$ is a square matrix. Then, the eigenvalues of the matrix P are precisely the roots of the characteristic polynomial of P, given by $char_P(\lambda) = \det(P - \lambda I)$. The polynomial $char_P(\lambda)$ has degree n. So by the Fundamental Theorem of Algebra, P has a multiset of n eigenvalues.*

Definition A.1.2 (Inverse Matrix)**.** *Suppose there is a matrix $P \in \mathbb{F}^{m\times n}$. Then, P^{-1} is its inverse matrix iff multiplying it by P returns the identity matrix. That is, the inverse matrix must satisfy:*

$$P^{-1}P = I$$

Definition A.1.3 (Transpose Matrix)**.** *Suppose there is a matrix $P = (p_{ij}) \in \mathbb{F}^{m\times n}$. Then, its transpose matrix, $P^T = (q_{ij}) = (p_{ji}) \in \mathbb{F}^{n\times m}$ matrix whose columns are the rows of the original matrix.*

Definition A.1.4 (Conjugate Transpose Matrix)**.** *Suppose there is a matrix $P = (p_{ij}) \in \mathbb{C}^{m\times n}$. Then, its conjugate transpose matrix, $P^\dagger = (q_{ij}) = (\overline{p_{ji}}) \in \mathbb{F}^{n\times m}$ is a matrix whose columns are the rows of the original matrix.*

Definition A.1.5 (Symmetric Matrix)**.** *A matrix P is symmetric iff it is equal to its transpose:*

$$P = P^T$$

Definition A.1.6 (Hermitian Matrix)**.** *A matrix P is Hermitian iff it is equal to its conjugate transpose:*

$$P = P^\dagger$$

Definition A.1.7 (Orthogonal Matrix)**.** *A matrix P is called orthogonal iff its transpose is its inverse:*

$$P^T = P^{-1}.$$

Definition A.1.8 (Unitary Matrix)**.** *A matrix P is called unitary iff its conjugate transpose is its inverse:*

$$P^\dagger = P^{-1}.$$

A.1.2 Other

As mentioned in **Section 2.3**, here is the statement of the Perron-Frobenius Theorem. Specifically, we consider only the application to ergodic Markov Chains by means of limiting scope to stochastic/transition matrices.

Theorem A.1.1 (Perron-Frobenius Theorem)**.** *bla bla*

For β-ensembles, we mention the invariance criterion.

Theorem A.1.2 (Orthogonal Group)**.** *The set of* all *orthogonal matrices in $\mathbb{F}^{n\times n}$ is a matrix group. (It is called the orthogonal group.)*

Theorem A.1.3 (Unitary Group)**.** *The set of* all *unitary matrices in $\mathbb{C}^{n\times n}$ is a matrix group. (It is called the unitary group.)*

A.1.3 Proof: Real Symmetric Matrices have Real Eigenvectors

How does this relate to the rest of the thesis?

Eigenvectors have a wide range of applications in various fields. As such, a result **guaranteeing** the existence of real-valued eigenvectors is a strong, non-trivial one. For instance, the stationary distribution of a Markov chain is an eigenvector of its transition matrix of eigenvalue 1. In other scenarios, right-eigenvectors represent the conditional expectation of a transition matrix, and so forth. Altogether, this proof showcases techniques and tools used in this related domains of study.

Notation. For notational convenience, for any $N \in \mathbb{N}$, let $\mathbb{N}_N = \{1, \ldots, N\}$.

The Proof

In this section, we will prove that for any $M \times M$ real symmetric matrix, $S_M \in \mathbb{R}^{M \times M}$, there exists for some eigenvalue λ, a corrosponding **real** eigenvector $\vec{v} \in \mathbb{R}^M$. Prior to starting the main proof, we begin by proving a auxilliary lemma.

Lemma. Suppose we have a $M \times M$ real symmetric matrix with a some eigenvalue λ. If there we have a corrosponding eigenvector $v \in \mathbb{C}^M$, then every entry of v, say v_i is equal to a **real** linear combination of the other entries $v_j \mid j \neq i$. So, we will show that:

$$\forall i \in \mathbb{N}_M : v_i = \sum_{j \neq i} c_j v_j \quad (c_j \in \mathbb{R})$$

Proof of Lemma. Begin by taking a real symmetric matrix S_M for some $M \in \mathbb{N}$. Suppose we have an eigenvalue λ. Then, if we have some eigenvector v, we know that:

$$(1) : \forall i \in \mathbb{N}_M : a_1 v_1 + \cdots + d_i v_i + \cdots + a_{m-1} v_m = \lambda v_i \quad (a_j \in \mathbb{R})$$

We obtain (1) by expanding the equality $A\vec{v} = \lambda \vec{v}$ and noticing that every row of Av is expressible as the sum of the non-diagonal entries multiplied by $v_j \mid j \neq i$ plus $d_i v_i$. Next, we collect the terms to solve for v_i:

$$\forall i \in \mathbb{N}_M : a_1 v_1 + \cdots + a_{m-1} v_m = v_i(\lambda - d_i)$$

Since S_M is a real symmetric matrix, the a_j terms are real so we can say:

$$\forall i \in \mathbb{N}_M : v_i(\lambda - d_i) = \sum_{j \neq i} a_j v_j \quad (a_j \in \mathbb{R})$$

Finally, divide both sides by $(\lambda - d_i)$. On the right hand side, the coefficients of v_j become $\frac{a_j}{(\lambda - d_i)}$. Denote this coefficient $a'_j = \frac{a_j}{(\lambda - d_i)}$. Since S_M is a real symmetric matrix, we know its eigenvalues are real so $\lambda \in \mathbb{R}$ and that its entries are real so

$a_i, d_i \in \mathbb{R}$. Since a'_j is an arithmetic expression involving real numbers, then it follows that for every j, $a'_j \in \mathbb{R}$. As such, we can rewrite v_j as the following sum.

$$\forall i \in \mathbb{N}_M : v_i = \sum_{j \neq i} c_j v_j \quad (\forall j : a'_j \in \mathbb{R})$$

Thus, for any $M \in \mathbb{N}$, a real symmetric matrix $S_M \in \mathbb{R}^{M \times M}$ with eigenvalue λ must have a corrosponding eigenvector v such that each of its entries is expressible as a real linear combination of the other entries. So, the proof is complete. $\square$

Now, leveraging the result of this lemma, we will prove the main theorem.

Theorem A.1.4 (Taqi)**.** *Suppose we have a $M \times M$ real symmetric matrix, S_M. Then, we will show that there exists for some eigenvalue λ, a corrosponding **real** eigenvector $\vec{v} \in \mathbb{R}^M$.*

Proof. For this proof we will induct on the dimension of the matrix, M. So, let the inductive statement be:

$$f(M) : S_M \text{ has a real eigenvector } v \text{ corrosponding to an eigenvalue } \lambda$$

Base Case. Take the base case $M = 2$. This proof is left to the reader as an exercise. Begin by taking a 2×2 symmetric matrix, and show that there exist real coefficients for the eigenvector corrosponding to λ by using Gaussian Elimination.

Inductive Step. For our inductive step, we need to show that $f(M) \Rightarrow f(M+1)$. So, let us assume $f(M)$. This means that we can assume any real symmetric matrix S_M has a real eigenvector $v \in \mathbb{R}^M$ corrosponding to λ.

Next, we will write S_{M+1} as the matrix S_M augmented by some $u \in \mathbb{R}^M$ as follows:

$$S_{M+1} = \left[\begin{array}{c|c} S_M & u \\ \hline u^T & d_{M+1} \end{array}\right]$$

From our lemma, we use the fact that S_{M+1} is symmetric and our lemma to obtain (1) and our assumption of $f(M)$ to obtain (2), listed below:

$$(1) : \forall i \in \mathbb{N}_{M+1} : v_i = \sum_{j \neq i} c_j v_j \quad (c_j \in \mathbb{R})$$

$$(2) : \forall i \in \mathbb{N}_M : v_i \in \mathbb{R}$$

In particular for (2), we know that $v_i = \left(\sum_{j \neq i} \frac{a_j}{d_i - \lambda} v_j\right)$. From (1), we know that for the $(m+1)^{th}$ row, $v_{m+1} = \sum_{j \neq m+1} c_j v_j$ for real coefficients $c_j \in \mathbb{R}$. By (2), this is a linear combination of real entries v_i. Since $v_{m+1} \in \mathbb{R}$, this means we have shown that:

$$\forall i \in \mathbb{N}_{M+1} : v_i \in \mathbb{R}$$

In other words, we have established that $f(M) \Rightarrow f(M+1)$. By induction, the proof of the theorem is complete. $\square$.

A.2 Probability Theory

Please refer to Hwang & Blitzstein (2019) as a resource; the definitions were sourced from there. For the definitions and theorems covered in Section A.1, consider checking out Chapter 3: Random Variables and their distributions. For Order Statistics, consider Chapter 8, Section 6: Order Statistics.

A.2.1 Random Variables

Definition A.2.1 (Random Variable)**.** *A random variable $X : \Omega \to \mathbb{R}$ is a function from some sample space $\Omega = \{s_i\}_{i=1}^n$ to the real numbers $\mathbb{R}$. The sample space is taken to be any set of events such that the probability function corrosponding to the random variable, p_X exhausts over all the events in Ω. In other words, we expect $\int_\Omega p_X(s) = 1$.*

PDFs

Definition A.2.2 (Probability Density Function)**.** *For a continuous r.v. X with CDF F, the probability density function (PDF) of X is the derivative f of the CDF, given by $f(x) = F'(x)$. The support of X, and of its distribution, is the set of all x where $f(x) > 0$.*

Theorem A.2.1 (Characterizing the PDF)**.** *A probability density function is characterized by a few properties that are necessary for it to be valid. They are as follows:*

1. *Non-negativity: the PDF must be a non-negative valued function everywhere.*

$$f(x) \geq 0$$

2. *Integrates to 1: the PDF must integrate to 1 when integrated over its entire support.*

$$\int_{-\infty}^{\infty} f(x)dx = 1$$

Example (Normal PDF)**.** *For example, consider the probability density function for a r.v. $X \sim \mathcal{N}(\mu, \sigma)$.*

$$\boldsymbol{P}(X = x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

CDFs

Definition A.2.3 (Cumulative Distribution Function)**.** *The cumulative distribution function (CDF) of an r.v. X is the function F_X given by $F_X(x) = \boldsymbol{P}(X \leq x)$. When there is no risk of ambiguity, we sometimes drop the subscript and just write F (or some other letter) for a CDF.*

Theorem A.2.2 (Characterizing the CDF)**.** *A cumulative distribution function is characterized by a few properties that are necessary for it to be valid. They are as follows:*

1. *Monotonic function: the CDF must always be a monotonically increasing function. That is:*
$$x_0 \leq x_1 \implies F(x_0) \leq F(x_1)$$
2. *Right-continuous: The CDF is continuous except possibly for having some jumps. Wherever there is a jump, the CDF is continuous from the right. That is, for any a, we have:*
$$F(a) = \lim_{x \to a^+} F(x)$$
3. *Converges to 0 and 1 in its limits: since the CDF represents a cumulative probability, its limits must reflect that aspect of probability spaces. So, the CDF must satisfy:*
$$\lim_{x \to -\infty} F(x) = 0 \text{ and } \lim_{x \to \infty} F(x) = 1$$

Independence

Definition A.2.4 (Independence)**.** *Random variables X and Y are said to be independent if $\forall x, y \in \mathbb{R}$:*
$$\boldsymbol{P}(X \leq x, Y \leq y) = \boldsymbol{P}(X \leq x)\boldsymbol{P}(Y \leq y)$$
If the variables are discrete, then this is equivalent to the condition:
$$\boldsymbol{P}(X = x, Y = y) = \boldsymbol{P}(X = x)\boldsymbol{P}(Y = y)$$

Definition A.2.5 (i.i.d)**.** *A vector of random variables $\vec{X} = (X_i)_{i=1}^N$ is said to be i.i.d (independent and identially distributed) is each of its entries X_i are exactly so.*

A.2.2 Statistics

Definition A.2.6 (Statistic)**.** *A statistic is formally defined as a function of a vector of random variables. So, for instance f is a statistic of a vector of random variables $\bar{X}$. Its observed value is given by $f(\vec{X})$.*

Example (Mean Statistic)**.** *For instance, the mean of a random sample $\vec{X}$ is a statistic. Formally, we would define the statistic $f : \vec{X} \to \frac{\sum_i x_i}{N}$. So the mean of the random sample is defined as $\bar{x} = f(\vec{X})$.*

Order Statistics

Definition A.2.7 (Order Statistic)**.** *Suppose $\vec{X} = \{X_i\}_{i=1}^N$ is an ordered vector of random variables. Then, the i^{th} order statistic of X is given by X_i.*

Example (Order Statistic)**.** *Suppose $X = (20, 7, 2, 1)$. The smallest (fourth) order statistic is 1. The second order statistic is 7. The largest (first) order statistic is 20.*

Theorem A.2.3 (Order Statistics of Uniform i.i.d Variables)**.** *Let $U_0, U_1, \ldots, U_n$ be i.i.d. Unif$(0, 1)$. Then, the j^{th} order statistic $U_{(j)}$, is distributed as $U_{(j)} \sim Beta(j, n - j + 1)$.*

A.3 Markov Chains

How does this relate to the rest of the thesis?

Tie back the relevance of this section to the rest of this thesis. This will come up in **Appendix D**. It is also important because transition matrices are representations of markov chains. Please refer to Hwang & Blitzstein (2019), Chapter 11 for all this information.

Definition A.3.1 (Markov Chain)**.** *Say a set of random variables X_i each take a value in a set, called the state space, $S_M = \{1, 2, \ldots, M\}$. Then, a sequence of such random variables $X_0, X_1, \ldots, X_n$ is called a Markov Chain if the following conditions are satisifed:*

- $\forall X_i : X_i$ has support and range $S_M = \{1, 2, ..., M\}$.
- **(Markov Property)** The transition probability from state $i \to j$, given by $\mathbf{P}(X_{n+1} = j \mid X_n = i)$ is conditionally independent from all past events in the sequence $X_{n-1} = i', X_{n-2} = i'', \ldots, X_0 = i^{(n-1)}$, excluding the present/last event in the sequence. In other words, given the present, the past and the future are conditionally independent. So, $\forall i, j \in S_M$, we observe:

$$\mathbf{P}(X_{n+1} = j \mid X_n = i) = \mathbf{P}(X_{n+1} = j \mid X_n = i, X_{n-1} = i', \ldots, X_0 = i^{(n-1)})$$

Transition Matrices

Definition A.3.2 (Transition Matrix)**.** *Take a Markov Chain with states $\{1, \ldots, M\}$. Letting $q_{ij} = \boldsymbol{P}(X_{n+1} = j \mid X_n = i)$ be the transition probability from $i \to j$, then the matrix $Q = (q_{ij})$ is the transition matrix of the chain.*

- Q is a non-negative matrix. So every entry $q_{ij} \in \mathbb{R}^+$ is non-negative. This follows because probabilities are necessarily non-negative values.
- For this transition matrix to be valid, its rows have to be stochastic, meaning their entries sum to 1. This may be understood as applying the law of total probability to the event of transitioning from any given state $i \in S_M$. In other words, the chain has to go somewhere with probability 1.

$$\forall i \in \mathbb{N}_M : \sum_{j \in \mathbb{N}_M} q_{ij} = 1$$

- Note, it is **not** necessary that the converse holds. The columns of our transition matrix need not sum to 1 for it to be a valid transition matrix.

Definition A.3.3 (n-step Transition Probability)**.** *The n-step transition probability of $i \to j$ is the probability of being at j exactly n steps after being at i. We denote this value $q_{ij}^{(n)}$:*

$$q_{ij}^{(n)} : \mathbf{P}(X_n = j \mid X_0 = i)$$

Realize:

$$q_{ij}^{(2)} = \sum_{k \in S_M} q_{ik} \cdot q_{kj}$$

Because by definition, a Markov Chain is closed under a support/range of S_M so the event $i \to j$ may have taken any intermediate step $k \in S_M$. Realize by notational equivalence, $Q^2 = (q_{ij}^{(2)})$. Inducting over n, we then obtain that:

$$q_{ij}^{(n)} \text{ is the } (i,j) \text{ entry of } Q^n$$

Definition A.3.4 (Marginal Distribution of X_n)**.** *Let $\boldsymbol{t} = (t_1, t_2, \ldots, t_M)$ such that $\forall i \in S_M : t_i = \boldsymbol{P}(X_0 = i)$. So, $\boldsymbol{t} \in \mathbb{R}^{1 \times M}$. Then, the marginal distribution of X_n is given by the product of the vector $\boldsymbol{t}Q^n \in \mathbb{R}^{1 \times M}$. That is, the j^{th} component of that vector is $\boldsymbol{P}(X_n = j)$ for any $j \in S_M$. We may call $\boldsymbol{t}$ an initial state distribution.*

Classification of states

- A state $i \in S_M$ is said to be **recurrent** if starting from i, the probability is 1 that the chain will *eventually* return to i. If the chain is not recurrent, it is **transient**, meaning that if it starts at i, there is a non-zero probability that it never returns to i.
- Caveat: As we let $n \to \infty$, our Markov chain will gurantee that all transient states will be left forever, no matter how small the probability is. This can be proven by letting the probability be some ε, then realizing that by the support of $\text{Geom}(\varepsilon)$ is always some finite value, then the equivalence between the Markov property and independent Geometric trials gurantees the existence of some finite value such that there is a success of never returning to i.

Definition A.3.5 (Reducibility)**.** *A Markov chain is said to be **irreducible** if for any $i, j \in S_M$, it is possible to go from $i \to j$ in a finite number of steps with positive probability. In other words:*

$$\forall i, j \in S_M : \exists n \in \mathbb{N} : q_{ij}^{(n)} > 0$$

- From our quantifier formulation of irreducible Markov chains, note that we can equivalently say that a chain is irreducible if there is integer $n \in \mathbb{N}$ such that the (i,j) entry of Q^n is positive for any i, j.
- A Markov chain is **reducible** if it is not **irreducible**. Using our quantifier formulation, it means that it suffices to find transient states so that:

$$\exists i, j \in S_M : \nexists n \in \mathbb{N} : q_{ij}^{(n)} > 0$$

Appendix B

Algorithm Appendix

In this appendix, we will enumerate the various algorithms used in this thesis for the purpose of transparency and reproducibility. Every one of these algorithms have their code counterpart in the **RMAT** package. For the most part, the implicit $\mathcal{D}$-matrices are the non-trivial algorithms to peruse. Changing methodology could impact results, and as such, all results are tied to the algorithms in this appendix.

B.1 Implicit $\mathcal{D}$-Matrices

Algorithm B.1.1 (Stochastic Row)**.**

1. *To sample a row r of size N, fix $N \in \mathbb{N}$.*
2. *Sample a vector $\vec{X}$ with N i.i.d entries between $[0, 1]$. So, sample $\vec{X} \sim Unif(0, 1)$.*
3. *Assign $r \leftarrow \vec{X}$, and then normalize the row by diving each entry by the row sum; so assign $r \leftarrow r \cdot \frac{1}{\sum_{i=1}^{N} r_i}$*
4. *Return the stochastic row r.*

Algorithm B.1.2 (Stochastic Matrix)**.**

1. *To generate a stochastic square matix P of size $N \times N$, fix $N \in \mathbb{N}$.*
2. *Then, for every row of P, randomly sample a stochastic row of size N and assign it.*
3. *Return the stochastic matrix P.*

Algorithm B.1.3 (Symmetric Stochastic Matrix)**.**

1. *To sample a symmetric stochastic matrix P of size N, fix $N \in \mathbb{N}$.*
2. *Sample a random stochastic matrix Q of size N.*
3. *Choosing one of the triangles of Q, set both the upper and lower of triangles of P to be that triangle.*

4. *Set the diagonal of the matrix P to be equal to 1 minus the sum of the non-diagonal entries.*

5. *Return the symmetric stochastic matrix P.*

Algorithm B.1.4 (Transition Matrix for an Erdos-Renyi Graph)**.**

1. *Fix $N \in \mathbb{N}$ and $p \in [0, 1]$.*

2. *Generate a matrix Q such that every entry $i, j \in 1, \ldots, N$ is $x_{ij} \sim$ Unif$(0, 1)$.*

3. *For each row r_i in $\{1, \ldots, N\}$, generate $deg(v_i) \sim$ Bin(N, p).*

4. *Randomly chose $N - deg(v_i)$ vertices, set the entries x_{ij} in the j columns to 0 to sever them.*

5. *Renormalize the matrix by dividing each row by its sum; let $(x_i) \leftarrow (x_i)/\sum_j (x_i)$.*

B.2 Explicit $\mathcal{D}$-Matrices

Algorithm B.2.1 (Homogenous Explicit $\mathcal{D}$-Matrix)**.**

1. *To simulate a $\mathcal{D}$-distributed square matrix P of size N, fix $N \in \mathbb{N}$.*

2. *Sample a vector $\vec{X}$ with N i.i.d entries from $\mathcal{D}$. So, generate $\vec{X} = X_1, \ldots, X_N$ where $X_i$is i.i.d $\mathcal{D}$.*

3. *Assign the vector $\vec{X}$ as a row of the matrix P. Repeat for every other row.*

4. *Return the $\mathcal{D}$-distributed matrix P.*

Algorithm B.2.2 (Dumitriu's Beta Matrix)**.**

1. *To simulate an $N \times N$ beta matrix, fix $N \in \mathbb{N}$.*

2. *Start by taking a diagonal of $\mathcal{N}(0, 2)$ variables.*

3. *Set both of the nearest off-diagonals to the row that samples from a $\chi(df = c_j)$ where $c_j = \beta \cdot j$ for columns spanning $j = 1, \ldots, n - 1$.*

Appendix C

Code Appendix

The RMAT Package.
The RMAT package is composed of two primary modules, the random matrix module (matrices.R) and the spectral statistics module (spectrum.R) and (dispersion.R). They can be further subdivided into smaller modules as follows.
Note. The source code was functional as of R 4.0.5.

1. **Random Matrix Module**
   (a) Explicitly Distributed Matrices
   (b) Implicitly Distributed Matrices
   (c) Ensemble Extensions
2. **Spectral Statistics Module**
   (a) Spectrum
   (b) Dispersions
   (c) Parallel Extensions

C.1 Matrix Module

Random matrices can either be explicitly or implicitly distributed. If they are explicitly distributed, their entries have a specific distribution. Otherwise, the entries have an implicit distribution imposed by generative algorithm the matrix uses.

C.1.1 Explicitly Distributed Matrices

For (homogenous) explicitly distributed matrices, we can use a "function factory" method to be concise. The actual implementation is more verbose for the purposes of argument documentation, but the following code is minimal and fully functional.

Homogenously Explicit Distributions

```
# ... represents all the arguments taken in by the rdist function
RM_explicit <- function(rdist){
  function(N, ..., symm = FALSE){
    # Create an [N x N] matrix sampling the rows from rdist, passing ... to rdist
    P <- matrix(rdist(N^2, ...), nrow = N)
    # Make symmetric if prompted
    if(symm){P <- .makeHermitian(P)}
    # Return P
    P
  }
}

# A version where we add an imaginary component
RM_explicit_cplx <- function(rdist){
  RM_dist <- function(N, ..., symm = FALSE, cplx = FALSE, herm = FALSE){
    # Create an [N x N] matrix sampling the rows from rdist, passing ... to rdist
    P <- matrix(rdist(N^2, ...), nrow = N)
    # Make symmetric/hermitian if prompted
    if(symm || herm){P <- .makeHermitian(P)}
    # Returns a matrix with complex (and hermitian) entries if prompted
    if(cplx){
      # Recursively add imaginary components as 1i * instance of real-valued matrix.
      Im_P <- (1i * RM_dist(N, ...))
      # Make imaginary part Hermitian if prompted
      if(herm){P <- P + .makeHermitian(Im_P)}
      else{P <- P + Im_P}
    }
    P # Return the matrix
  }
}
```

With our function factories set up, we can quickly generate all the random matrix functions for all the distributions our hearts could desire.

```
RM_unif <- RM_explicit_cplx(runif)
RM_norm <- RM_explicit_cplx(rnorm)
```

Beta Matrices

For the β-ensemble matrices, we simply use the algorithm provided in Dimitriu's paper. Doing so, we get the function:

```
# Generate a Hermite beta matrix using Dumitriu's Matrix Model
RM_beta <- function(N, beta){
  # Set the diagonal ~ N(0,2)
  P <- diag(rnorm(n = N, mean = 0, sd = sqrt(2)))
  # Get degrees of freedom sequence for offdigonal
  df_seq <- beta * (N - seq(1, N-1))
  # Set the off-1 diagonals as chi squared variables with df(beta_i)
  P[row(P) - col(P) == 1] <- P[row(P) - col(P) == -1] <- sqrt(rchisq(N-1, df_seq))
  # Rescale the entries by 1/sqrt(2)
  P <- P/sqrt(2)
  # Return the beta matrix
  P
}
```

C.1.2 Implicitly Distributed Matrices

In the case of implicitly distributed matrices, we have various types of stochastic matrices.

Stochastic Matrices.

For stochastic matrices, we require slightly more setup. First, we setup the row functions to sample probability vectors:

```
# Generates stochastic rows of length N
.stoch_row <- function(N){
  # Sample a vector of probabilities
  row <- runif(n = N, min = 0, max = 1)
  # Return the normalized row (sums to one)
  row / sum(row)
}
```

For random introduced sparsity, we define the following row function.

```
# Generates same rows as in .stoch_row(N), but with randomly introduced sparsity
.stoch_row_zeros <- function(N){
  # Sample a vector of probabilities
  row <- runif(n = N, min = 0, max = 1)
  # Sample a vertex degree of at least one (as to ensure row is stochastic)
  degree_vertex <- sample(x = 1:(N-1), size = 1)
  # Sever a random selection of edges to set the vertex degree
  row[sample(1:N, size = N - degree_vertex)] <- 0
  # Return normalized row
  row / sum(row)
}
```

Once this is done, we can use this function iteratively. With some magic, we can incorporate an option to make the matrix symmetric, and we get the following function.

```
RM_stoch <- function(N, symm = F, sparsity = F){
  # Choose row function depending on sparsity argument
  if(sparsity){row_fxn <- .stoch_row_zeros} else {row_fxn <- .stoch_row}
  # Generate the [N x N] stochastic matrix stacking N stochastic rows
  P <- do.call("rbind", lapply(X = rep(N, N), FUN = row_fxn))
  # Make symmetric (if prompted)
  if(symm){ P <- .makeStochSymm(P) }
  # Return the matrix
  P
}
```

Erdos-Renyi Stochastic Matrices.

For the Erdos-Renyi walks, we do something similar by defining a parameterized row function.

```
# Generates a stochastic row with parameterized sparsity of p
.stoch_row_erdos <- function(N, p){
  # Sample a vector of probabilities
  row <- runif(n = N, min = 0, max = 1)
  # Sample the vertex degree so that it is ~ Bin(n,p)
  degree_vertex <- rbinom(n = 1, size = N, prob = 1 - p)
  # Sever a random selection of edges to set the vertex degree
  row[sample(1:N, degree_vertex)] <- 0
  # Return normalized row only if non-zero (cannot divide by 0)
  if(sum(row) != 0){
    row / sum(row)
  } else{
    .stoch_row_erdos(N, p) # Otherwise, try again
  }
}
```

And we again use the row function iteratively to get the following function.

```
RM_erdos <- function(N, p, stoch = T){
  # Generate an [N x N] Erdos-Renyi stochastic matrix by stacking N p-stochastic rows
  P <- do.call("rbind", lapply(X = rep(N, N), FUN = .stoch_row_erdos, p = p))
  # Return the Erdos-Renyi transition matrix
  P
}
```

And as such, we have minimal, functional implementations of functions that sample random matrices! In total, we only needed two helper functions. The '.off-diagonalEntries' function was used to normalize the probabilities in 'RM-stoch' and 'RM-erdos'.

```
# Returns a Hermitian version of a matrix by manual assignment
.makeHermitian <- function(P){
  for(i in 1:nrow(P)){
    for(j in 1:ncol(P)){
      # Select the entries in the upper triangle (i < j)
      if(i < j){
        # Make the upper triangle equal to the conjugate transpose of the lower triangle
        P[i,j] <- Conj(P[j,i])
      }
    }
  }
  # Return the Hermitian Matrix
  P
}

# Return the off-diagonal entries of row i
.offdiagonalEntries <- function(row, row_index){row[which(1:length(row) != row_index)]}
```

C.1.3 Ensemble Extensions

Lastly, we have the ensemble extensions. These functions are quite simple to implement user a "function factory". Again, the actual implementations are more verbose due to the argument descriptions, but otherwise, are exactly the same.

```
# Extends a RM_dist function to its RME_dist ensemble counterpart
RME_extender <- function(RM_dist){
  # Function returns a list of replicates of the RM_dist function with '...' as arguments
  function(N, ..., size){
    lapply(X = rep(N, size), FUN = RM_dist, ...)
  }
}
```

Now, we extend the functions as follows, and we are done with the matrix module!

```
RME_unif <- RME_extender(RM_unif)
RME_norm <- RME_extender(RM_norm)
RME_beta <- RME_extender(RM_beta)
RME_stoch <- RME_extender(RM_stoch)
RME_erdos <- RME_extender(RM_erdos)
```

C.2 Spectral Statistics Module

C.2.1 Spectrum

```
spectrum <- function(array, components = T, norm_order = T, singular = F, order = NA){
  # Digits to round values to
  digits <- 4
  # Get the type of array
  array_class <- .arrayClass(array)
  # For ensembles, iteratively rbind() each matrix's spectrum
  if(array_class == "ensemble"){
    map_dfr(array, .spectrum_matrix, components, norm_order, singular, order, digits)
  }
  # From matrices, call the function returning the ordered spectrum for a singleton matrix
  else if(array_class == "matrix"){
    .spectrum_matrix(array, components, norm_order, singular, order, digits)
  }
}
```

```
# Helper function returning tidied eigenvalue array for a matrix
.spectrum_matrix <- function(P, components, norm_order, singular, order, digits = 4){
  # For singular values, take P as product of the itself and its tranpose
  if(singular){P <- P %*% t(P)}
  # Get the eigenvalues of P
  eigenvalues <- eigen(P, only.values = TRUE)$values
  # Take the square root of the eigenvalues to obtain singular values
  if(singular){eigenvalues <- sqrt(eigenvalues)}
  # Sort the eigenvalues to make it an ordered spectrum
  eigenvalues <- .sortValues(eigenvalues, norm_order)
  # If uninitialized, select all orders; otherwise, use c() so singletons => vectors
  if(class(order) == "logical"){order <- 1:nrow(P)} else{order <- c(order)}
  # Return the spectrum of the matrix
  purrr::map_dfr(order, .resolve_eigenvalue, eigenvalues, components, digits)
}
```

```r
# Read and parse an eigenvalue from a sorted eigenvalue array
.resolve_eigenvalue <- function(order, eigenvalues, components, digits){
  # Read from a sorted eigenvalue array at that order
  eigenvalue <- eigenvalues[order]
  # Get norm and order columns
  features <- data.frame(Norm = abs(eigenvalue), Order = order)
  if(components){
    # If components are sought, resolve the eigenvalue into seperate columns first
    res <- cbind(data.frame(Re = Re(eigenvalue), Im = Im(eigenvalue)), features)
  } else{
    # Otherwise, don't resolve the eigenvalue components
    res <- cbind(data.frame(Eigenvalue = eigenvalue), features)
  }
  # Round entries and return the resolved eigenvalue
  res <- round(res, digits)
  return(res)
}
```

```r
# Parses an array to see classify it as a matrix or an ensemble of matrices.
.arrayClass <- function(array){
  # Sample an element from the array and get its class
  elem <- array[[1]]
  types <- class(elem)
  # Classify it by analyzing the element class
  if("numeric" %in% types || "complex" %in% types){
    return("matrix")
  }
  else if("matrix" %in% types){
    return("ensemble")
  }
}

# Sort an array of numbers by their norm (written for eigenvalue sorting)
.sortValues <- function(vals, norm_order){
  values <- data.frame(value = vals)
  # If asked to sort by norms, arrange by norm and return
  if(norm_order){
    values$norm <- abs(values$value)
    values <- values %>% arrange(desc(norm))
    # Return the norm-sorted values
    values$value
  }
  # Otherwise, sort by sign and return
  else{ sort(vals, decreasing = TRUE) }
}
```

C.2.2 Dispersions

```r
# Compute the dispersion of a matrix or matrix ensemble
dispersion <- function(array, pairs = NA, norm_order = T, singular = F, pow_norm = 1){
  # Digits to round values to
  digits <- 4
  # Get the type of array
  array_class <- .arrayClass(array)
  # Parse input and generate pair scheme (default NA), passing on array for dimension
  pairs <- .parsePairs(pairs, array, array_class)
  # For ensembles; iteratively rbind() each matrix's dispersion
  if(array_class == "ensemble"){
    map_dfr(array, .dispersion_matrix, pairs, norm_order, singular, pow_norm, digits)
  }
  # Array is a matrix; call function returning dispersion for singleton matrix
  else if(array_class == "matrix"){
    .dispersion_matrix(array, pairs, norm_order, singular, pow_norm, digits)
  }
}
```

```r
# Find the eigenvalue dispersions for a given matrix
.dispersion_matrix <- function(P, pairs, norm_order, singular, pow_norm, digits = 4){
  # Get the ordered spectrum of the matrix
  eigenvalues <- spectrum(P, norm_order = norm_order, singular = singular)
  # Generate norm function to pass along as argument (Euclidean or Beta norm)
  norm_fn <- function(x){ (abs(x))^pow_norm }
  # Compute and return the dispersion
  map2_dfr(pairs[["i"]], pairs[["j"]], .resolve_dispersion, eigenvalues, norm_fn, digits)
}
```

```r
# Read and parse a dispersion observation between eigenvalue i and j.
.resolve_dispersion <- function(i, j, eigenvalues, norm_fn, digits){
  # Initialize dispersion dataframe by adding order of eigenvalues compared
  disp <- data.frame(i = i, j = j)
  # Add the eigenvalues
  disp$eig_i <- .read_eigenvalue(i, eigenvalues)
  disp$eig_j <- .read_eigenvalue(j, eigenvalues)
  # Get the identity difference
  disp$id_diff <- disp$eig_j - disp$eig_i
  # Compute norm of the identity difference (standard norm metric)
  disp$id_diff_norm <- norm_fn(disp$id_diff)
  # Compute the difference of absolutes
  disp$abs_diff <- norm_fn(disp$eig_j) - norm_fn(disp$eig_i)
  # Round digits
  disp <- round(disp, digits)
  # Get the ranking difference
  disp$diff_ij <- disp$i - disp$j
  # Return the resolved dispersion observation
  disp
}
```

```r
# Parse a string argument for which pairing scheme to utilize
.parsePairs <- function(pairs, array, array_class){
  # Valid schemes for printing if user is unaware of options
  valid_schemes <- c("largest", "lower", "upper", "consecutive", "all")
  # Set default to be the consecutive pair scheme
  if(class(pairs) == "logical"){pairs <- "consecutive"}
  # Stop function call if the argument is invalid
  if(!(pairs %in% valid_schemes)){
    scheme_list <- paste(valid_schemes, collapse = ", ")
    stop(paste("Invalid pair scheme. Try one of the following: ", scheme_list, ".", ""))
  }
  # // Once we verify that we have a valid pair scheme string, try to parse it.
  # First, obtain a matrix by inferring array type; if ensemble take first matrix
  if(array_class == "ensemble") { P <- array[[1]] }
  else if(array_class == "matrix") { P <- array }
  # Obtain the dimension of the matrix
  N <- nrow(P)
  # Parse the pair string and evaluate the pair scheme
  if(pairs == "largest"){pair_scheme <- data.frame(i = 2, j = 1)}
  else if(pairs == "consecutive"){pair_scheme <- .consecutive_pairs(N)}
  else if(pairs == "lower"){pair_scheme <- .unique_pairs_lower(N)}
  else if(pairs == "upper"){pair_scheme <- .unique_pairs_upper(N)}
  else if(pairs == "all"){pair_scheme <- .all_pairs(N)}
  # Return pair scheme
  return(pair_scheme)
}
```

Appendix D

Mixing Time Simulations

D.1 Introduction

In this chapter, we'll talk about ratio-mixing time simulations. Essentially, these simulations are a method of approximating the distribution of a random transition matrix's mixing time. There will be a fun exploration of the Erdos-Renyi matrix ensembles and we will computationally show that the parameterized ensemble has a mixing time inversely proportional to graph sparsity.

D.2 Mixing Time Simulations

With the Erdos-Renyi graph defined, we may now motivate the simulation of random walks on them. First, however, we need to generate their corrosponding transition matrices. An algorithm for this is outlined below.

Suppose we have simulated a transition matrix for an Erdos-Renyi graph called Q. Now, fixing some initial probability distribution $\vec{x} \in \mathbb{R}^M$, we may consider the evolution sequence of a random walk on this Erdos-Renyi graph by taking its evolution sequence $\mathcal{S}(Q, x)$.

Definition D.2.1 (Random Batches)**.** *Let $\mathbb{F}$ be a field, and fix some $M \in \mathbb{N}$. Let $\mathcal{B}_\lambda \subset \mathbb{F}^M$ be a uniformly random batch of points in the M-hypercube of length λ. That is,*

$$\mathcal{B}_\lambda = \{\vec{x} \mid x_i \sim \textit{Unif}(-\lambda, \lambda) \textit{ for } i = 1, \ldots, M\}$$

Note: *If $\mathbb{F} = \mathbb{C}$, then take $\vec{x} \in \mathcal{B}_\lambda$ to mean $\vec{x} = a + bi$ where $a, b \sim \textit{Unif}(-\lambda, \lambda)$.*

Definition D.2.2 (Evolution Sequence)**.** *An evolution sequence of a vector $\vec{\pi}$ and a transition matrix Q is defined as the sequence $\mathcal{S}(Q, \pi) = (\pi'_n)_{n=1}^N$ where $\pi'_n = \pi Q^n$*

Definition D.2.3 (Finite Evolution Sequences)**.** *Suppose we sample a random point from $\mathcal{B}_\lambda$, emulating a random point $\vec{v} \in \mathbb{F}^M$. Additionally, let $Q \in \mathbb{F}^{M \times M}$ be a transition matrix over $\mathbb{F}$. Fixing a maximum power ('time') $T \in \mathbb{N}$, define the evolution sequence of $\vec{v}$ as follows:*

$$\mathcal{S}(v, Q, T) = (\alpha_n)_{n=1}^T \textit{ where } \alpha_k = vQ^k$$

If we do not impose a finiteness constraint on the sequence, we consider powers for $n \in \mathbb{N}$ or $t = \infty$

Definition D.2.4 (Consecutive Ratio Sequences)**.** *Accordingly, define the consecutive ratio sequence (CST) of $\vec{v}$ as follows:*

$$\mathcal{R}(v, Q, T) = (r_n)_{n=2}^{T} \text{ where } (r_n)_j = \frac{(\alpha_n)_j}{(\alpha_{n-1})_j} \text{ for } j = 1, \ldots, M$$

In other words, the consecutive ratio sequence of v can be obtained by performing ***component-wise division*** *on consecutive elements of the evolution sequence of v.*

Definition D.2.5 (Near Convergence)**.** *Because these sequences may never truly converge to eigenvectors of the matrix, we formalize a notion of "near convergence". As a prelimenary, we first define ε-equivalence. Let $\mathbb{F}$ be a field, and fix $\varepsilon \in \mathbb{R}^+$. Suppose we have vectors $v, v' \in \mathbb{F}^M$. Then, $v \sim_\varepsilon v'$ if $||v - v'|| < \varepsilon$ where $|| \cdot ||$ is the norm on $\mathbb{F}$.*

Let $\varepsilon \in \mathbb{R}^+$, and suppose we have an evolution sequence $(a[\vec{v}])_n$. Then, a_n ε-converges at $N \in \mathbb{N}$ if:

$$\forall n \geq N \mid a_N \sim_\varepsilon a_n$$

D.3 Erdos-Renyi Ensemble Simulations

Definition D.3.1 (Erdos-Renyi Graph)**.** *An Erdos-Renyi graph is a graph $G = (V, E)$ with a set of vertices $V = 1, \ldots, M$ and edges $E = \mathbb{1}_{i,j \in V} \sim Bern(p_{ij})$. It is homogenous if $p_{ij} = p$ is fixed for all i, j.*

Essentially, an Erdos-Renyi graph is a graph whose 'connectedness' is parameterized by a probability p (assuming it's homogenous, which this document will unless otherwise noted). As $p \to 0$, we say that graph becomes more sparse; analogously, as $p \to 1$ the graph becomes more connected.

Recall from probability theory that a sum of i.i.d Bernoulli random variables is a Binomial variable. As such, we may alternatively say that the degree of each vertex v is distributed as $deg(v) \sim Bin(M, p)$. This is helpful to know because the process of simulating graphs becomes much simpler.

D.4 Questions

1. How are the entries of the CRS distributed? Are they normal, and if so, what is its mean?

2. Are the entries of the CRS i.i.d as $t \to \infty$?

3. For an Erdos-Renyi matrix, is the mixing time t dependent on the parameter p?

4. What impact does the running time parameter T have on σ (the variance of the distribution of the CRS entries)?

D.4.1 Cauchy Distributed Ratios

It seems to be the case that the **log-transformed** entries of the CRS are Cauchy distributed about $\log \lambda_1$ where $\lambda_1 = \max(\sigma(Q))$, the largest eigenvalue of Q. That is,

$$r_i \sim \text{Cauchy}(\ln \lambda_1) \text{ for } i = 1, \ldots, M$$

References

Dumitriu, I., & Edelman, A. (2018). Matrix models of beta ensembles. *Journal of Mathematical Physics 43, 5830*, (pp. 1–5).

Gernot Akemann, J. B., & Francesco, P. D. (2015). *Oxford Handbook of Random Matrix Theory*. Oxford University Press.

Hwang, J., & Blitzstein, J. K. (2019). *Introduction to Probability*. CRC Press, Taylor and Francis Group.

Mehta, M. L. (2004). *Random Matrices, 3rd Edition (Vol. 142)*. Academic Press.

Miller, S. J., & Takloo-Bighash, R. (2004). *An Invitation to Modern Number Theory*. Academic Press.

Tao, T. (2012). *Introduction to Random Matrix Theory*. American Mathematical Society.

Wigner, E. (1955). *Characteristic Vectors of Bordered Matrices with Infinite Dimensions*.