A Dive into the Tracy-Widom Law via Longest Increasing Subsequences

Daniel K. Rui - June 8, 2020

Abstract

Let S_n be the set of permutations of the numbers $\{1, \dots, n\}$. Define $L(\sigma)$ to be the length of the longest increasing subsequence in σ. Denote σ_n to be a random variable such that $P(\sigma_n = \sigma) = \frac{1}{n!}$ for all $\sigma \in S_n$, and let $\ell_n := \mathbb{E}[L(\sigma_n)]$. The aim of this paper is to give a overview of the methods used to arrive at the asymptotics of ℓ_n and $L(\sigma_n)$ ($\to$ and $\to_p$ $2\sqrt{n}$ resp.), as well as numerical data on the rate of convergence of appropriately scaled $L(\sigma_n)$ to the limiting distribution F_2, the Tracy-Widom distribution, ending with a brief discussion on the universality of F_2.

1 Setup

Let S_n be the set of permutations of the numbers $[n] := \{1, \dots, n\}$ (and so $|S_n| = n!$). Let $\sigma \in S_n$ be one of the $n!$ permutations, and define $L(\sigma) = \max\{k \in [n] : \sigma \text{ has a length } k \text{ increasing subsequence}\}$ to be the length of the longest increasing subsequence in σ. For example, $L((1,2,3)) = 3$, $L((3,2,1)) = 1$, and $L((1,3,2)) = 2$. Similarly, define $D(\sigma)$ to be the length of the longest decreasing subsequence. If we consider σ_n to be a random variable such that $P(\sigma_n = \sigma) = \frac{1}{n!}$ for all $\sigma \in S_n$, then we can define $\ell_n := \mathbb{E}[L(\sigma_n)] = \frac{1}{n!}\sum_{\sigma \in S_n} L(\sigma)$.

Lemma 1.1: Erdös - Szekeres

If we have $n > rs$ for $r, s \in \mathbb{N}$, then for any $\sigma \in S_n$, at least one of the following must be true: $L(\sigma) > r$ or $D(\sigma) > s$.

Proof idea: let $L_i(\sigma)$ and $D_i(\sigma)$ be the length of the longest increasing subsequence of σ ending at the ith number. Then for all $i \in \{1, \dots, n-1\}$, we know that $L_i(\sigma) < L_{i+1}(\sigma)$, or $D_i(\sigma) < D_{i+1}(\sigma)$ (because the $(i+1)$th number either greater or less than the ith number). Thus, the pairs $\{(L_i(\sigma), D_i(\sigma))\}_{i=1}^n$ are all distinct (because going from one pair to the next, one of the two numbers increases). Thus by the pigeonhole principle, it is impossible that $L(\sigma) \leq r$ and $D(\sigma) \leq s$.

Lemma 1.2: First lower bound

For $n \geq 1$, $\ell_n \geq \sqrt{n}$

Proof idea: by Erdös - Szekeres (1.1), we have (for all $\sigma \in S_n$) that $n \leq L(\sigma)D(\sigma)$ (because otherwise either $L(\sigma) > L(\sigma)$, or $D(\sigma) > D(\sigma)$, which is impossible). Then, because L and D are symmetrical and by AM-GM, (recall that σ_n is a random variable),

$$\ell_n = \frac{\mathbb{E}[L(\sigma_n)] + \mathbb{E}[D(\sigma_n)]}{2} = \mathbb{E}\left[\frac{L(\sigma_n) + D(\sigma_n)}{2}\right] \geq \mathbb{E}\left[\sqrt{L(\sigma_n)D(\sigma_n)}\right] \geq \mathbb{E}\left[\sqrt{n}\right] = \sqrt{n}$$

Lemma 1.3: First upper bound

$$\limsup_{n\to\infty} \frac{\ell_n}{\sqrt{n}} \leq e$$

Proof idea: refer to page 9 of Romik [9]

1.1 Hammersley's convergence theorem

We now proceed to our first major theorem:

Theorem 1.1: Hammersley's convergence theorem

The limit $\lim_{n\to\infty} \frac{\ell_n}{\sqrt{n}}$ exists (denote it Λ), and $\frac{L(\sigma_n)}{\sqrt{n}} \to_p \Lambda$ as $n \to \infty$.

Proof idea: refer to pages 10-15 of Romik [9] (perhaps even Exercises 3,4 on page 71).

2 Tableaus

We will now discuss how to to compute $L(\sigma)$; this in turn will lead delightfully into another form of the problem from a more geoemtric perspective, which will in turn be analyzed using some deep results from analysis. The algorithm we'll start with is the patience sorting algorithm (name coined by C.L. Mallows, algorithm due to A.S.C. Ross in the early 1960s [9]), where we take in a permutation and put the numbers in stacks (where the TOPS of the stacks are aligned, and the bottoms allowed to be jagged, looking like $\overline{\downarrow\downarrow\downarrow}$, where the horizontal line is the top of the stacks, and the arrows are pointing to the bottoms of each stack), according to the following rule:

- Given the ith number, say x, from the permutation and the stacks generated by that first $i-1$ numbers from the permutation, place x in a new stack on the right if it is greater than all numbers at the top of all previous stacks, and otherwise, on the top of the leftmost stack for which x is less than the number at the top of the stack (pushing everything in the stack down).

The runnable code for this algorithm is given in the Appendix in Python (just copy-paste into any Python editor/IDE, running Python 3.X); simply change the value of n to whatever you desire:

Lemma 2.1: Length of first row is $L(\sigma)$

The length of the first row of the resulting tableau is $L(\sigma)$ (in the notation of the code above, `len(arr[0])` $= L(\sigma)$)

Proof idea: given a longest increasing subsequence, each number must be in different column (they can't be in the same column because the only way for two numbers to be in the same column is if the second one is smaller than the first, which is not the case when we are considering an increasing subsequence). This proves $\geq$. To prove $\leq$, choose x_s to be the top of the rightmost stack (stack number s), and choose x_{s-1} to be the number at the top of stack number $(s-1)$ at the time we inserted x_s. Continue like this until we get to x_1. Then $\{x_1, \ldots, x_s\}$ forms an increasing subsequence.

2.1 The Robinson-Schensted Algorithm

Unfornately, the patience sorting algorithm as it stands is not useful enough to allow us a way forward in studying the problem. However, the Robinson-Schensted algorithm, which is a recursive variant of the original patience sorting algorithm further arranges the numbers in a form (called the Young tableau) that is more susceptible to attack. Let us first define formally what a Young tableau is:

- A Young diagram is a diagram with n boxes, with r rows (row 1 above row 2 and so on), where the ith row has λ_i boxes, where $\lambda_1 \geq \ldots \geq \lambda_r > 0$ and $\lambda_1 + \ldots + \lambda_r = n$, and c columns (col 1 left of col 2 and so on), where the jth col has λ'_j boxes, and where $\lambda'_1 \geq \ldots \geq \lambda_c > 0$ and $\lambda'_1 + \ldots + \lambda'_c = n$. The shape of a Young diagram is encapsulated by any partition $\lambda = (\lambda_1, \ldots, \lambda_r)$ of n. In other words, $\lambda =\in \mathscr{P}(n)$, where $\mathscr{P}(n)$ is the set of (positive) integer partitions of n.

- A Young tableau is where one fills in the boxes of a Young diagram with the numbers $\{1, \ldots, n\}$ such that the numbers in each row and column are increasing.

The Robinson-Schensted algorithm produces a Young tableau from any $\sigma \in S_n$ as follows:

- Given the ith number, say x, from the permutation and the stacks generated by that first $i-1$ numbers from the permutation, place x in a new stack on the right if it is greater than all numbers at the top of all previous stacks, and otherwise, on the top of the leftmost stack for which x is less than the number at the top of the stack, NOT by pushing everything down, but instead by evicting the number that used to be in the top of that stack.

- With the evicted number a, we perform the above step except instead of looking at row 1, we look at row 2; i.e., if a is greater than all the numbers on row 2, then we place a in the leftmost empty spot on row 2. Otherwise, we find the leftmost number in row 2 that a is less than, and evict that number and put a in its place.

- We continue evicting numbers and moving down rows (creating a new row if need be). Once everything is settled, we move on to the $(i+1)$th number in the permutation and again insert/evict row by row, until we've put all the numbers from the permutation into the structure.

The Python code for this is given in the Appendix. We will not prove that the outputs of the algorithm are Young tableau here, though rest assured it is true.
To allow for this algorithm to have an inverse, we must have some sort of structure to keep track of what changes; that structure being another Young tableau (of the same shape). We call these two tableaux the insertion tableau P (above) and the recording tableau Q.

- To make the recording tableau, simply place the number i in the exact same spot in the recording tableau that the Robinson-Schensted algorithm places the number x_i (the ith number of the permutation) in the insertion tableau.

- Given P and Q generated by some $\sigma \in S_n$, we can find x_n (the last number in the permutation) by noting where n is in Q, finding that spot in P, and backtracking row by row (because once we have a number on say row j that we know was evicted, then we can find the number on row $j-1$ that evicted it), until we get down to row 1. More precise details are in the code.

The Python code for this is given in the Appendix. We will not prove that given any two Young tableaux of size n, this algorithm does in fact give σ that generates such tableaux (though an approach could be by induction). Notice that the first row generated by the R-S algorithm and patience sorting are the same, and so Lemma 2.1 says that $L(\sigma) = \lambda_1$.

2.2 Tableau Shapes and Partitions

Define d_λ (referred to as the "dimension" of λ) as the number of Young tableaus with shape λ. To rephrase the algorithms from the previous section as a theorem, we have

Theorem 2.1: Robinson-Schensted correspondence

As we saw, the Robison-Schensted algorithm maps a $\sigma \in S_n$ to a pair of Young tableaux (P, Q) of the same shape, and the inverse R-S algorithm maps pairs (P, Q) to $\sigma \in S_n$. More importantly, the R-S algorithm and its inverse form a bijection between S_n and the set of pairs of size-n Young tableaux.

Proof idea: from any $\sigma \in S_n$, we can generate exactly one pair of tableaux P, Q (via the Robinson-Schensted algorithm), and for any pair of Young tableaux P, Q, we can generate exactly one permutation σ (by the inverse algorithm), where we furthermore know that σ generates exactly P, Q.

An easy consequence of Theorem 2.1 is that $\sum_{\lambda \in \mathscr{P}(n)} d_\lambda^2 = n!$. Define the random variable $\lambda^{(n)}(\omega)$ to be the shape of the Young tableau generated by $\sigma_n(\omega)$ for all $\omega \in \Omega$, the original probability space of σ_n. The above summation formula tells us that $P(\lambda^{(n)} = \lambda) = \frac{d_\lambda^2}{n!}$ for all $\lambda \in \mathscr{P}(n)$; this probability measure on the set $\mathscr{P}(n)$ is referred to as Plancherel measure (originally discovered by M. Plancherel in the early 1900s arising in the context of representation theory [9])

2.3 The Hook-Length Formula

From before, we've given $\lambda \in \mathscr{P}(n)$ the meaning of "a tuple of positive integers representing a partition of n" and "the shape of a Young tableau". We will now give it another meaning: "the set of all (i, j) such that the $j \in \{1, \dots, \lambda_i\}$" (i.e the set of coordinates for all the boxes in the tableaux). Define the hook $H_\lambda(i, j)$ to be "$\ulcorner$"-shaped region of cells in the Young tableau with shape λ, where the corner of the "$\ulcorner$" is the cell (i, j) (the jth box on the ith row), and where the arms extend as far as they can go (so the horizontal arm goes from column j to column c, and the vertical arm goes from row i to row r), and define $h_\lambda(i, j) = |H_\lambda(i, j)|$ to be the the number of cells in $H_\lambda(i, j)$.

Lemma 2.2: The hook-length formula

For any $\lambda \in \mathscr{P}(n)$,

$$d_\lambda = \frac{n!}{\prod_{(i,j) \in \lambda} h_\lambda(i, j)}$$

Proof idea: refer to pages 26-30 of Romik [9].

3 Limiting Shapes and $\Lambda = 2$

In the past section, we've transformed the problem from asking about the length of the longest increasing subsequence in a random permutation, to the length of the first row of a random Young diagram (chosen according to Plancherel measure). In fact, we will prove a much stronger result than just about the first row; we will determine the limiting shape of the entire Young diagram. However, we only give the barest of outlines for these next sections; the material introduced prior (such as the Robinson-Schensted correspondence) will come up later in the paper in further inquiries beyond the longest increasing subsequence problem (albeit in a tangential way), while the material in the next sections are specific to this problem, and hence not as important to cover in detail. We first define a space on which to study these functions:

Definition 3.1: the space of continual Young diagrams

Let $\mathscr{F}$ be the space of function $f : [0, \infty) \to [0, \infty)$ satisfying:

- f is weakly decreasing (non-increasing)
- $\int_0^\infty f(x)\,dx = 1$ (scaling shapes to look about the same for all n instead of growing bigger)
- f has compact support: $\sup\{x \geq 0 : f(x) > 0\} < \infty$

For a random $\lambda \in \mathscr{P}(n)$ (again distributed according to Plancherel measure), define $\phi_\lambda \in \mathscr{F}$ by

$$\phi_\lambda(x) = \frac{1}{\sqrt{n}} \lambda'_{\lfloor x\sqrt{n} \rfloor + 1}$$

where we scale the horizontal and vertical axes by $1/\sqrt{n}$ to get an area of 1 (recall the usage of λ' from the definition of Young tableau; this style of representing Young tableaux is called the French style, which is the vertical flip of the English style we've been using). Now for any $f \in \mathscr{F}$, as an analogue to the hook-length $h_\lambda(i, j)$ above, define

$$h_f(x, y) = f(x) - y + f^{-1}(y) - x = f(x) - y + (\inf\{x \geq 0 : f(x) \leq y\}) - x$$

Then, we have that

Lemma 3.1: Asymptotic hook-length formula

Uniformly over all $\lambda \in \mathscr{P}(n)$ and as $n \to \infty$,

$$P(\lambda^{(n)} = \lambda) = \frac{d_\lambda^2}{n!} = \exp\left[-n\left(1 + 2I_{\text{hook}}(\phi_\lambda) + \mathscr{O}\left(\frac{\log n}{\sqrt{n}}\right)\right)\right]$$

where

$$I_{\text{hook}}(f) = \int_0^\infty \int_0^{f(x)} \log h_f(x, y)\,dy\,dx$$

Proof idea: refer to pages 36-39 of Romik [9].

The correct thing to do at this point turns out to be to find f such that $I_{\text{hook}}(f)$ is minimized (intuitively, this finds f such that the exponential expression above that relates to $P(\lambda^(n) = \lambda)$ is maximized, something we would want to have because of the heuristic that the behavior of the random model with asymptotically high probability is close to the behavior that is most likely, where the exponential expression above can be thought of as a likelihood "measure" over $\mathscr{F}$ — further discussion of this can be found on pages 39-40 of Romik [9]). We will change the coordinates once more (the inverse of f in the definition of h_f is quite troublesome) by rotating counter-clockwise by 45 degrees (this coordinate system is referred to as the Russian style); i.e. we take

$$u = \frac{x-y}{\sqrt{2}}, v = \frac{x+y}{\sqrt{2}} \quad \text{and} \quad f(x) = y \iff v = g(u) \iff \frac{v-u}{\sqrt{2}} = f\left(\frac{v+u}{\sqrt{2}}\right)$$

where g is the transformed version of f. The transformed version $\mathscr{G}$ of the space $\mathscr{F}$ is defined on page 43 of Romik [9]. Our hook integral formula can then be expressed as

$$I_{\text{hook}}(f) = \frac{1}{2}\iint_{-\infty<t<s<\infty} \log(\sqrt{2}(s-t))(1+g'(t))(1-g'(s))\,dt\,ds$$

Furthermore, defining $h(u) = g(u) - |u|$ (and the corresponding transformed version $\mathscr{H}$ of the space $\mathscr{G}$), we can write

$$2I_{\text{hook}}(f) = \log 2 + Q(h) + L(h) =: J(h)$$

for

$$Q(h) = -\frac{1}{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \log|s-t|h'(t)h'(s)\,dt\,ds \quad \text{and} \quad L(h) = -2\int_{-\infty}^{\infty} h'(u)(u\log|u| - u)\,du$$

Minimizing $I_{\text{hook}}(f)$ on $\mathscr{F}$ is now equivalent to minimizing $J(h)$ on $\mathscr{H}$.

Theorem 3.1: Minimizer of J

The function $h_0 \in \mathscr{H}$:

$$h_0(u) = \begin{cases} \frac{2}{\pi}\left(u\arcsin(\frac{u}{\sqrt{2}}) + \sqrt{2-u^2}\right) - |u| & \text{for } |u| \le \sqrt{2} \\ 0 & \text{otherwise} \end{cases}$$

minimizes J on $\mathscr{H}$. Furthermore, for any $h \in \mathscr{H}$,

$$J(h) \ge J(h_0) + Q(h - h_0)$$

where $Q \ge 0$ on $\mathscr{H}$, with equality iff $h = 0$ (so h_0 is the unique minimizer). Lastly, $J(h_0) = -1$.

Proof idea: refer to pages 47-55 of Romik [9]

For the random partition $\lambda^{(n)}$ (chosen according to Plancherel measure), we already have $\phi_{\lambda^{(n)}} \in \mathscr{F}$, so define $\psi_{\lambda^{(n)}} \in \mathscr{G}$ to be the rotated version of $\phi_{\lambda^{(n)}}$. Define the norm $||h||_Q = \sqrt{Q(h)}$. Then, we have the following very beautiful theorem:

Theorem 3.2: The limiting shape

Define

$$\Omega(u) = \begin{cases} \frac{2}{\pi}\Big(u \arcsin(\frac{u}{\sqrt{2}}) + \sqrt{2-u^2}\Big) & \text{for } |u| \leq \sqrt{2} \\ |u| & \text{otherwise} \end{cases}$$

Then we have $P(||\psi_{\lambda^{(n)}} - \Omega||_Q > \epsilon) \to 0$ as $n \to \infty$. Moreoever, the convergence is not only in the norm d_Q, it is also in the sup-norm $|| \bullet ||_\infty$, i.e. for any $\epsilon > 0$, as $n \to \infty$,

$$P\Big(\sup_{u \in \mathbb{R}} |\psi_n(u) - \Omega(u)| > \epsilon\Big) \to 0$$

Proof idea: for any $\lambda \in \mathscr{P}(n)$, we have $\phi_\lambda \in \mathscr{F}$, so let $g_\lambda \in \mathscr{G}$ to be the transformed ϕ_λ, and let $h_\lambda = g_\lambda - |u|$. Let $\mathscr{M}_n$ be the set of $\lambda \in \mathscr{P}(n)$ satisfying $||g_\lambda - \Omega||_Q = \sqrt{Q(h_\lambda - h_0)} > \epsilon$. From Theorem 3.1, $J(h_\lambda) \geq -1 + Q(h_\lambda - h_0) > -1 + \epsilon^2$. Then from Lemma 3.1,

$$P(\lambda^{(n)} = \lambda) \leq \exp[-\epsilon^2 n + \mathscr{O}(\sqrt{n} \log n)]$$

By a loose bound on the partition function, $|\mathscr{M}_n| \leq |\mathscr{P}(n)| \leq Ce^{C\sqrt{n}}$ for some constant $C > 0$, so

$$P(\lambda^{(n)} \in \mathscr{M}_n) = \sum_{\lambda \in \mathscr{M}_n} P(\lambda^{(n)} = \lambda) \leq C \exp[-\epsilon^2 n + C\sqrt{n} + \mathscr{O}(\sqrt{n} \log n)]$$

which tends to 0 as $n \to \infty$. The extension to the sup norm relies on the fact that $||f||_\infty \leq C(Q(f))^{1/4}$ for a certain class of f; refer to pages 57-60 of Romik [9].

With the limit-shape theorem, we get the following bound for Λ (from all the way back in Theorem 1.1):

Lemma 3.2: $\Lambda \geq 2$

$\Lambda \geq 2$

Proof idea: follows shortly from Theorem 3.2; refer to page 62 of Romik [9] for details.

3.1 $\Lambda \leq 2$

The inequality in the other direction requires a bit more work, involving a new concept called the Plancherel growth process. This is the main result, from which $\Lambda \leq 2$ follows immediately:

Theorem 3.3: ℓ_n growth bound

$$\ell_n - \ell_{n-1} = \mathbb{E}\Big[\lambda_1^{(n)} - \lambda_1^{(n-1)}\Big] \leq \frac{1}{\sqrt{n}}$$

Proof idea: refer to pages 65-68 of Romik [9].

Lemma 3.3: $\Lambda \le 2$

$\Lambda \le 2$

Proof idea: Theorem 3.3 and induction.

Thus concludes the theorem of A. Vershik, S. Kerov, and independently B.F. Logan, L.A. Shepp of 1977. For the history surrounding this problem, take a look at pg. 6 of Romik [9] for the events preceding this theorem, and pg. 80 (with the note on pg. 336) for the events since.

4 Tracy-Widom, an Introduction

The Tracy-Widom distribution was first studied by Tracy and Widom in the context of random matrix theory; they considered the Gaussian Unitary Ensemble (GUE) model of random Hermitian matrices M, where the values on the diagonal $M_{i,i}$ are i.i.d. Normal(0,1) random variables, and the values above the diagonal are $M_{j,i} + iM'_{j,i}$ where the $M_{j,i}, M'_{j,i}$ are all independent (and independent of all the $M_{i,i}$), and distributed Normal$(0, \frac{1}{2})$. For an $n \times n$ GUE matrix, the largest eigenvalue grows like $\sqrt{2n}$, and has standard deviation $\mathcal{O}(n^{-1/6})$ [6], and so it seems sensible to define:

Definition 4.1: the Tracy-Widom distribution

The distribution function F_2 known as the Tracy-Widom distribution can be defined (pg. 7 of [6]) as

$$F_2(x) = \lim_{n\to\infty} P\left(\sqrt{2}n^{1/6}\left(\lambda_{\max} - \sqrt{2n}\right) \le x\right)$$

To see more about where the constants come from (for example, the factor of $\sqrt{2}$ that remains unexplained), it may be instructive to look at a textbook cited in Tracy and Widom's 1994 paper [10], Mehta's "Random Matrices", 2nd edition, published in 1990 [8], which already had hints of the Airy kernel (defined on the next page) as relating to the max eigenvalue problem; see in particular the scaling/shifting transformation done to the joint p.d.f. of the eigenvalues of the GUE matrix on page 64 (explained in more detail on page 59) for some ideas regarding the motivation behind scaling $\lambda_{\max}$ the way we did in the definition of F_2; and chapter 18 (pages 372 to 376) for the hints about the Airy kernel (for ease of reference, one should know that the notation σ_N is defined on page 72).

For more explicit development of the hints Mehta provided in his 2nd edition, one can take a look at chapter 24 of his 3rd edtion [7], published 2004, where he includes the results of Tracy and Widom from later that decade, a natural extension of where he left off in chapter 18 of the 2nd edition. In fact, this translation of the question about max eigenvalues to Fredholm determinants of the Airy kernel is "natural" enough (i.e. it seems that the Airy approach has fewer "arbitrary" constants) to consider defining the Tracy-Widom distribution using the Airy kernel. Romik does exactly this in his book [9]:

Definition 4.2: the Tracy-Widom distribution, Airy version

Let $\mathrm{Ai}(x)$ denote the Airy function

$$\mathrm{Ai}(x) = \frac{1}{\pi}\int_0^\infty \cos\left(\tfrac{1}{3}t^3 + xt\right) dt,$$

otherwise known as the unique (up to scalar multiple) solution of the differential equation $y'' = xy$ satisfying $y \to 0$ as $x \to \infty$. Define the Airy kernel $\mathbf{A} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ to be

$$\mathbf{A}(x,y) = \begin{cases} \frac{\mathrm{Ai}(x)\,\mathrm{Ai}'(y) - \mathrm{Ai}'(x)\,\mathrm{Ai}(y)}{x-y} & \text{if } x \neq y \\ \mathrm{Ai}'(x)^2 - x\,\mathrm{Ai}(x)^2 & \text{if } x = y \end{cases}$$

The Tracy-Widom distribution function F_2 can be defined (pg. 81 of Romik [9]) as

$$F_2(t) = 1 + \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} \int_t^\infty \dots \int_t^\infty \det_{i,j=1}^{n}\Big(\mathbf{A}(x_i, x_j)\Big)\, dx_1 \dots dx_n =: \det(\mathbf{I} - \mathbf{A})$$

The $\det(\mathbf{I} - \mathbf{A})$ refers to the Fredholm determinant, which Romik motivates in section 2.4 [9] (titled "Discrete Determinantal Point Processes") by considering determinants involving kernels $\mathbf{K} : \Omega \times \Omega \to \mathbb{R}$, key pages being end-of-page 92 to 93 (for the case where Ω is finite), end-of-page 96 to 97, end-of-page 98 to 99 (for the case where Ω is countable), and page 144 (for the case where Ω is uncountable, like in the case of the Airy kernel $\mathbf{A}$).

The other major definition of the Tracy-Widom distribution (also discovered by Tracy and Widom in their 1994 paper [10]) is that $F_2(t) = \exp(-\int_t^\infty (x-t)q^2(x)\,dx)$ where $q : \mathbb{R} \to \mathbb{R}$ is the unique solution to $q''(t) = tq(t) + 2q^3(t)$ (a special case of the Painléve II differential equation) also satisfying $q(t) \sim \mathrm{Ai}(t)$ as $t \to \infty$ [9].

Other discoveries about the Tracy-Widom distribution include that its p.d.f. is log-concave for $x \geq 0$ (proof, credited to P. Deift, appearing in a paper by M. Bóna, M. Lackner, and B. E. Sagan published in 2017 [3]), and that it is not infinitely divisible (proof by J. A. Domínguez-Molina also in 2017 [5]).

5 Baik-Deift-Johansson

In 1998, two decades after Logan-Shepp and Vershik-Kerov proved that $\mathbb{E}[L(\sigma_n)] \sim 2\sqrt{n}$, and about five years after Tracy-Widom published their results on the properties of F_2, J. Baik, P. Deift, and K. Johansson found the precise limiting distribution of the fluctuations of $L(\sigma_n)$ around its mean:

Theorem 5.1: the Baik-Deift-Johansson theorem

As $n \to \infty$,

$$P\left(\frac{L(\sigma_n) - 2\sqrt{n}}{n^{1/6}} \leq x\right) \to F_2(x),$$

Within a year, this result was strengthened (with three different proofs [9], by Okounkov, Borodin-Olshanski-Okounkov, and Johansson) to not only deal with $L(\sigma_n)$ (the length of the first row of the Young tableau), but also the lengths of all the other rows. To state it, we first have to define several things:

- A random point process on $\mathbb{R}$ is any random locally finite subset X of $\mathbb{R}$ (i.e. a locally finite collection of points in $\mathbb{R}$). Romik admits that this is not very formal, but he suggests thinking of X as some kind of random variable with values as sets of points in $\mathbb{R}$, for which one can begin asking questions like "what's the probability that exactly $k \in \mathbb{Z}_{\geq 0}$ points of X fall in some interval I".
- For such an X, define for every $n \in \mathbb{N}$ its n-point correlation function $\rho_X^{(n)} : \mathbb{R}^n \to [0, \infty)$ as

$$\rho_X^{(n)}(x_1, \ldots, x_n) = \lim_{\epsilon \searrow 0} \frac{P\Big(\Big[\#\{X \cap [x_1 - \epsilon, x_1 + \epsilon]\} = 1\Big] \cap \ldots \cap \Big[\#\{X \cap [x_n - \epsilon, x_n + \epsilon]\} = 1\Big]\Big)}{(2\epsilon)^n}$$

- Such a process X is determinantal if there is some correlation kernel $\mathbf{K} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ such that for all $n \in \mathbb{N}$,
$$\rho_X^{(n)}(x_1, \ldots, x_n) = \det_{i,j=1}^{n} \Big(\mathbf{K}(x_i, x_j)\Big)$$

The Airy ensemble X_{Airy} is the determinantal process with correlation kernel $\mathbf{A}(x, y)$ (the Airy kernel); that is to say,
$$\rho_{X_{\text{Airy}}}^{(n)}(x_1, \ldots, x_n) = \det_{i,j=1}^{n} \Big(\mathbf{A}(x_i, x_j)\Big)$$
We can label the random elements of X in decreasing order, so have that

$$X_{\text{Airy}} = \{\zeta_1, \zeta_2, \ldots : \zeta_1 > \zeta_2 > \ldots\}$$

(the inequalities are almost always strict). Then, we have the following theorem:

Theorem 5.2: the Borodin-Okounkov-Olshanski-Johnasson theorem

For all $n \in \mathbb{N}$, let $\lambda^{(n)}$ denote a random partition of n, chosen according to Plancherel measure of order n, and let $\lambda_j^{(n)}$ denote the jth number in the partition (when in decreasing order), or equivalently the length of its jth row (when thinking in Young tableux). Denote $\bar{\lambda}_j^{(n)} = n^{-1/6}(\lambda_j^{(n)} - 2\sqrt{n})$. Then for all $k \in \mathbb{N}$, we have that as $n \to \infty$,

$$(\bar{\lambda}_1^{(n)}, \ldots, \bar{\lambda}_k^{(n)}) \to_d (\zeta_1, \ldots, \zeta_k)$$

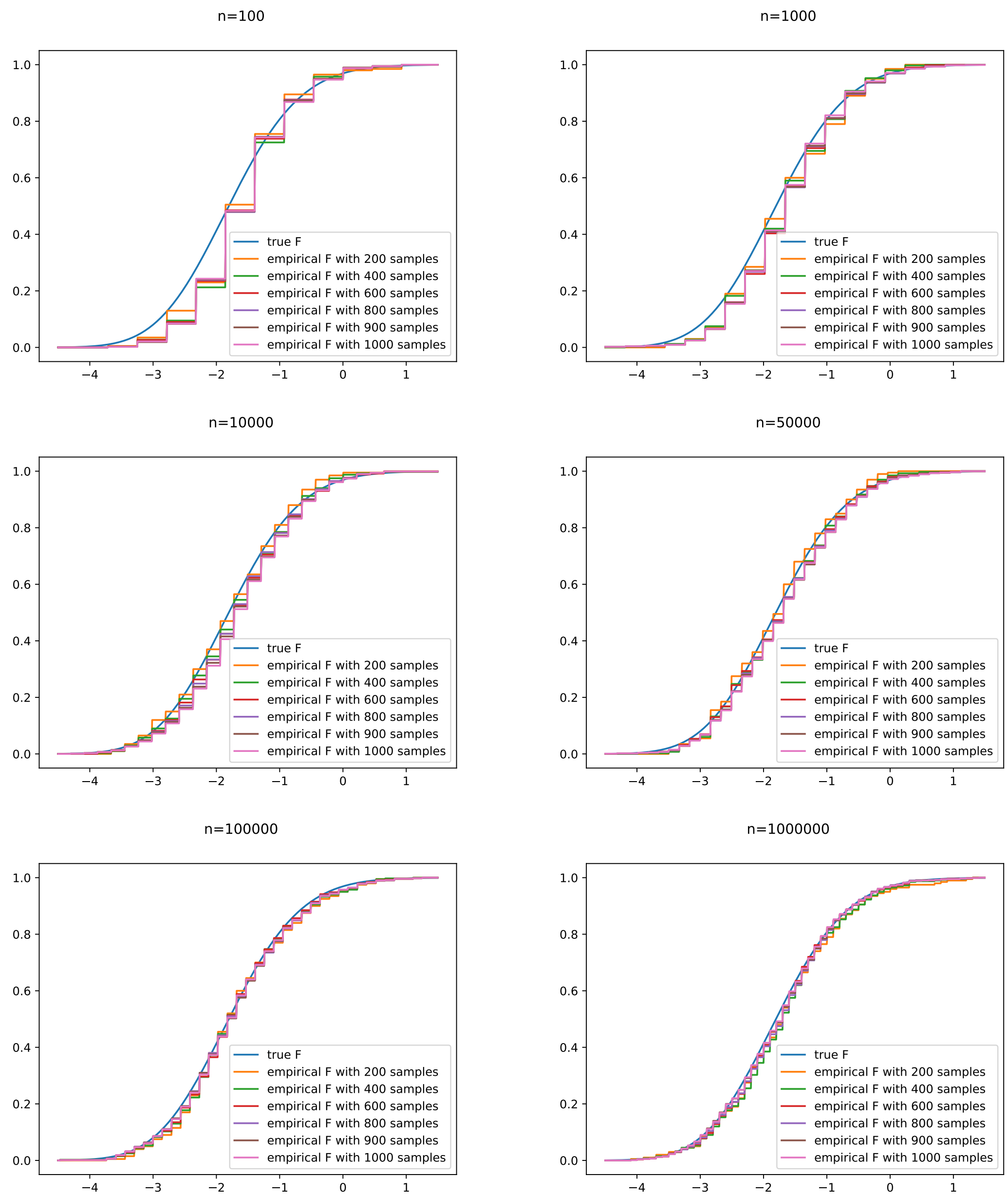
Proof idea: refer to Chapter 2 of Romik [9]

6 Rate of Convergence to Tracy-Widom

We now know that lengths of the longest increasing subsequence in a random permutation (appropriately scaled) converge to the Tracy-Widom distribution. But how fast does it converge? Unfortunately, I could not find any work regarding this question online, either numerical or analytic, so I wrote a little program (see the Appendix) to get some numerical data, generated as follows: for a selected n, we have 1000 random permutations $\sigma_{n,1}, \ldots, \sigma_{n,1000}$, and we calculate the empirical distribution function with m samples as:

$$\hat{F}_2^{(n,m)}(t) = \frac{1}{m} \sum_{i=1}^{m} 1_{(-\infty,t]} \left(\frac{L(\sigma_{n,i}) - 2\sqrt{n}}{n^{1/6}} \right)$$

where 2 is chosen as the subscript because of the relation with the Tracy-Widom distribution F_2. The plots for selected n and $m \in \{200, 400, 600, 800, 900, 1000\}$ are shown below:



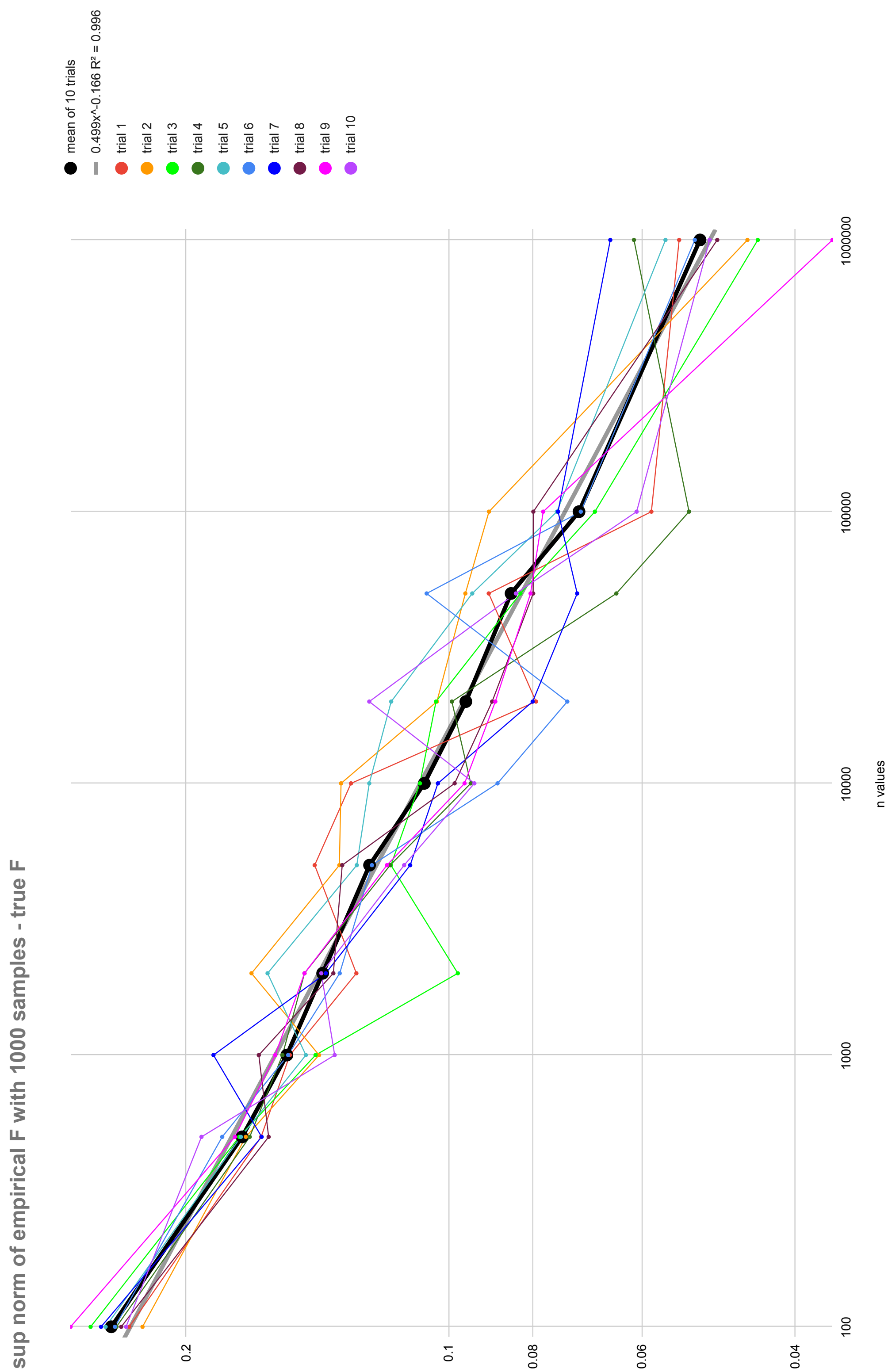
For any one of the selected n and $m = 1000$, we calculate the maximum deviation, $||\hat{F}_2^{(n,1000)} - F_2||_\infty$ and store it in a table. The tables below show the data from 10 trials for selected n:

n values	trial 1	trial 2	trial 3	trial 4	trial 5
100	0.232492	0.224492	0.257492	0.240492	0.247492
500	0.164099	0.1708562685	0.1738562685	0.169099	0.173099
1000	0.152097	0.140792	0.142097	0.155097	0.145792
2000	0.127573	0.168315	0.097573	0.146315	0.161315
5000	0.1424330859	0.1334330859	0.1164330859	0.116665	0.1274330859
10000	0.1293925005	0.132847	0.107847	0.094296	0.123296
20000	0.079363	0.103115	0.103363	0.099115	0.116363
50000	0.08991884911	0.09564399275	0.08268884694	0.064172	0.09391884911
100000	0.0585	0.089862	0.06791884911	0.05296129954	0.075097
1000000	0.054363	0.045388	0.044157	0.061256	0.056363

n values	trial 6	trial 7	trial 8	trial 9	trial 10
100	0.241492	0.250492	0.237492	0.271492	0.234492
500	0.1818562685	0.1638562685	0.1608562685	0.1758562685	0.192099
1000	0.152792	0.186097	0.165097	0.158097	0.135097
2000	0.133315	0.138315	0.135573	0.146315	0.1399541541
5000	0.1224330859	0.110665	0.1324330859	0.117665	0.1124330859
10000	0.087847	0.102847	0.09839250052	0.095847	0.09339250052
20000	0.0730574721	0.0800574721	0.089115	0.088363	0.123296
50000	0.10595	0.071172	0.07995104406	0.080513	0.08368884694
100000	0.0705	0.074862	0.07991884911	0.077862	0.06082937754
1000000	0.052157	0.065211	0.049157	0.036256	0.050157

The table below shows the mean maximum deviation over the 10 trials for each n. The graph on the next page plots the data from all 10 trials (in varying colors), along with the mean data (in bold black) in a log-log plot, along with the trendline for the mean data (in bold gray).

n values	mean of 10 trials
100	0.243792
500	0.1725533611
1000	0.1533055
2000	0.1394563154
5000	0.1232026601
10000	0.1066004502
20000	0.09552079442
50000	0.08476174289
100000	0.07083113753
1000000	0.0514465


sup norm of empirical F with 1000 samples - true F
mean of 10 trials
0.499x^-0.166 R² = 0.996
trial 1
trial 2
trial 3
trial 4
trial 5
trial 6
trial 7
trial 8
trial 9
trial 10
0.2
0.1
0.08
0.06
0.04
100
1000
10000
100000
1000000
n values

Although the data for each trial varies widely (especially as n increases), the data averaged together shows a strong linear (in the log-log scaling) correlation between the n values and the maximum deviation (the correlation coefficient between the mean data and its trendline is quite high). The line of best fit (in the log-log scaling) is very close to $\frac{1}{2}n^{1/6}$.

A reasonable conjecture would be to say that the true distribution function for a given n, $F_2^{(n)}$ (i.e. calculated using all $n!$ permutations instead of just m samples of random permutations $\in S_n$) satisfies $||F_2^{(n)} - F_2||_\infty \sim \frac{1}{2}n^{1/6}$, or at the very least $||F_2^{(n)} - F_2||_\infty \approx \frac{1}{2}n^{1/6}$ or $||F_2^{(n)} - F_2||_\infty = \mathcal{O}(n^{1/6})$. However, one should keep in mind that these are simply guesses based upon the data up to $n = 10^6$, which is not a very big number at all.

7 Universality of Tracy-Widom

Although the longest increasing subsequence problem is interesting and its relation to F_2 deep, the most important thing about the Tracy-Widom distribution is its yet-fully-understood ubiquitous nature. Deift [4] gives seven problems (the first of which is a physics problem, so I won't be talking about it), whose solutions end up involving the Tracy-Widom distribution in some form:

- Assuming the Riemann hypothesis, H. Montgomery in the early 1970's discovered that given the imaginary parts $\gamma_1 \leq \gamma_2 \leq \dots$ of the zeroes $\{\frac{1}{2} + i\gamma_k\}$ of the Riemann zeta function above the real axis, one can rescale them:

$$\gamma_k \mapsto \frac{\gamma_k \log \gamma_k}{2\pi} =: \tilde{\gamma}_k$$

so that the mean spacing approaches 1:

$$\lim_{T\to\infty} \frac{\#\{k \geq 1 : \tilde{\gamma}_k \leq T\}}{T} = 1$$

and get that

$$\lim_{N\to\infty} \sum_{1\leq i\neq j\leq N} f(\tilde{\gamma}_i - \tilde{\gamma}_j) = \int_{\mathbb{R}} f(r)\left(1 - \left(\frac{\sin(\pi r)}{\pi r}\right)^2\right) dr$$

for any rapidly decaying function f whose Fourier transform $\hat{f}(\xi)$ is supported in the interval $|\xi| < 2$. Moreover, if one could prove this for all smooth rapidly decaying functions f, and define the two-point correlation function

$$R(a,b) = \lim_{N\to\infty} \frac{1}{N}\#\{(k_1,k_2) : k_1 \neq k_2, 1 \leq k_1, k_2 \leq N, \tilde{\gamma}_{k_1} - \tilde{\gamma}_{k_2} \in (a,b)\}$$

(where (k_1, k_2) is an ordered pair while (a, b) is an interval), then one would have

$$R(a,b) = \int_a^b 1 - \left(\frac{\sin(\pi r)}{\pi r}\right)^2 dr$$

which is the exactly formula of the two-point correlation function of the eigenvalues of a random GUE matrix.

- The longest increasing subsequence problem we've been talking about, plus variants of it. For example, consider $S_n^{(\mathrm{inv})}$, the set of involutions in S_n, which has a corresponding set of Young tableaux (via the Robinson-Schensted correspondence), and therefore a new corresponding measure on Young diagrams (not Plancherel measure, which was for Young diagrams corresponding to all of S_n and not just the involutions), denoted $P^{(\mathrm{inv})}$. Taking Ω to be the set of all $\lambda = (\lambda_1, \lambda_2, \dots) \in \mathscr{P}(n)$ (where again $\lambda_1 \geq \lambda_2 \geq \dots$), and using the measure $P^{(\mathrm{inv})}$, J. Baik and E. M. Rains proved in 2001 that

$$\lim_{n\to\infty} P^{(\mathrm{inv})}\left(\frac{\lambda_1 - 2\sqrt{n}}{n^{1/6}} \leq x\right) \to F_1(x) \quad \text{and} \quad \lim_{n\to\infty} P^{(\mathrm{inv})}\left(\frac{\lambda_2 - 2\sqrt{n}}{n^{1/6}} \leq x\right) \to F_4(x)$$

where F_1 and F_4 are Tracy-Widom distributions for GOE and GSE matrices (different ensembles of random matrices; GOE is like GUE but everything is real and the matrix is symmetric, and GSE is like GUE but with quaternions).

- If one models n buses as n independent rate 1 Poisson processes going from the bus depot at time $t = 0$ to the terminal at time $t = T$, conditioned not to intersect for any $t \in [0, T]$, then at any observation point x along the route of length $N > n$, the probability distribution for the rescaled arrival times of the buses is exactly the same as the eigenvalue distribution for the Jacobi Unitary Ensemble, and thereby the GUE in the appropriate scaling limit.
  - On a slightly related note from [6], taking n independent 1-dimensional Brownian motions with time in $[0, 1]$ conditioned so that all paths start and end at the same point and do not intersect in $(0, 1)$, at any time $t_0 \in (0, 1)$, the positions of the paths (scaled appropriately) have the same distribution as the eigenvalues of an $n \times n$ GUE matrix.

- Imagine at $t = 0$ that we have walkers located at 0, 1, 2, and so on on the real line. At each tick of the clock, exactly one walker moves left one unit. However, no two walkers can be in the same spot. For any N, define d_N to be the distance traveled by the walker starting at 0 within $t = 0$ to $t = N$. Clearly, for any N, there are only finitely many possible walks of duration time N; suppose we now that we have probability measure P that makes all such walks equally likely (for any fixed N). Then,

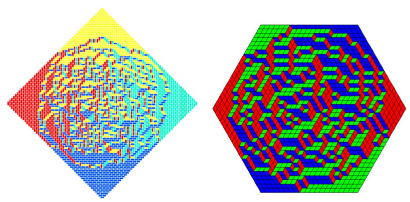
$$\lim_{N\to\infty} P\left(\frac{d_N - 2\sqrt{N}}{N^{1/6}} \leq t\right) = F_1(t)$$

To get F_2 as the limiting distribution, consider the same situation, but at $t = N$, we start choosing walkers to move right 1 unit, so that at $t = 2N$, everyone is back where they started. Again, for each fixed N, there are only finitely many walks, so make a new probability measure P' to make each one of these walks equally likely, and denote d'_N to be the farthest negative number at which the walker at 0 gets to. Then,

$$\lim_{N\to\infty} P'\left(\frac{d'_N - 2\sqrt{N}}{N^{1/6}} \leq t\right) = F_2(t)$$

These discoveries were made by J. Baik, E. M. Rains, and P.J. Forrester around 1999.

- Consider the rotated square $\{(x, y) : |x| + |y| \leq n + 1\}$ and consider tilings of the interior by horizontal and vertical 1×2 dominoes. Clearly for any fixed n, the number of tilings is finite, so let P_n be a probability measure such that each tiling is equally probable to occur. To make things simpler, scale everything so that the square is now $S = \{(u, v) : |u| + |v| \leq 1\}$ and the dominoes are $\frac{2}{n+1} \times \frac{1}{n+1}$. Define the inscribed circle C to be $\{(u, v) : u^2 + v^2 = \frac{1}{2}\}$. Then, it turns out that most of the tilings are "frozen" in $S \setminus C$ but unpredictable inside C (discovered by W. Jockusch, J. Propp, P. Shor in 1995), as shown in the left image below. K. Johansson in 2000 showed that fluctuations of the boundary of the "unpredictable" zone along the line $u + v = \alpha$ about the points of intersection between $u + v = \alpha$ and C were related to F_2.



Images taken from http://faculty.uml.edu/jpropp/tiling/www/.

  - On a slightly related note from [6], similar results hold for tilings of rhombi on hexagons (see right image above). Again, Tracy-Widom is involved in the transition between the "frozen" regions in the corners and the unpredictable center.

- One can make a simple model of plane boarding, where the seats are numbered 1 through N, and there is a permutation σ_n representing the order in which people are lined up outside the plane. For example, consider $\sigma_n = (3, 4, 1, 5, 6, 2)$. Person 3 goes to seat 3 and start loading bags, but 4 is blocked. Person 1 can go though and start loading bags. At $t = 1$, Person 3 and Person 1 sit, and now Person 4 and 2 can start loading bags. Person 5 and 6 are still blocked (by Person 4). At $t = 2$, Person 5 can start loading, and at $t = 3$, Person 6 can finally start loading. Thus everyone is seated by $t = 4$. Let $b(\sigma_n)$ be the boarding time given a permutation $\sigma_n \in S_n$. In 2005, E. Bachmat, D. Berend, L. Sapir, S. Skiena, and N. Stolyarov proved that

$$\lim_{n\to\infty} P\left(\frac{b(\sigma_n) - 2\sqrt{n}}{n^{1/6}} \leq t\right) = F_2(t)$$

  using the Robinson-Schensted correspondence on a different type of diagram, called Viennot diagrams.

After his series of examples, Deift ends his paper with a challenge — a challenge to prove a theorem like the central limit theorem, but with the Tracy-Widom distribution as the end result/limiting distribution instead of the Gaussian. He suggests two papers which begin to make progress into this question:

J. Baik and T. M. Suidan in 2004 [1] proved the following result:

Theorem 7.1: Baik-Suidan

Consider a family $\{X_{i,j}\}_{i,j=1}^{\infty}$ of i.i.d. mean-0 and variance-1 random variables with finite fourth moment $\mathbb{E}\big[|X_{i,j}|^4\big] < \infty$. Define

$$L(N,k) = \sup_{0=i_0 \le i_1 \le \ldots \le i_k = N} \sum_{j=1}^{k} \sum_{i=i_{j-1}+1}^{i_j} X_{i,j}$$

and

$$R(N,k) = \inf_{0=i_0 \le i_1 \le \ldots \le i_k = N} \sum_{j=1}^{k} \sum_{i=i_{j-1}+1}^{i_j} X_{i,j}$$

If $k, N \to \infty$ and $k = o(N^{\alpha})$ for $\alpha < \frac{3}{14}$, then

$$P\left(k^{1/6}\left(\frac{L(N,k)}{\sqrt{N}} - 2\sqrt{k}\right) \le t\right) \to F_2(t)$$

and

$$P\left(k^{1/6}\left(-\frac{R(N,k)}{\sqrt{N}} - 2\sqrt{k}\right) \le t\right) \to F_2(t)$$

If we further specify that the $X_{i,j}$ are all i.i.d. Gaussian random variables, then the above convergence to F_2 holds if $k = o(N^{\alpha})$ for $\alpha < \frac{3}{7}$.

and similarly also in 2004, T. Bodineau and J. B. Martin [2] proved the following result:

Theorem 7.2: Bodineau-Martin

Let $\omega_{i,r}$, $i \ge 0, r \ge 1$ be a family of i.i.d. random variables. Define $\Pi(n,k)$ to be the set of directed paths from $(0,1)$ to (n,k) where each step we increase one of the coordinates by 1. For $n \ge 0$ and $k \ge 1$, the last passage time to the point (n,k) is defined as

$$T(n,k) = \max_{\pi \in \Pi(n,k)} \sum_{(i,r) \in \pi} \omega_{i,r}$$

Supposing that $\mathbb{E}[|\omega_{i,r}|^p] < \infty$ for some $p > 2$, with $\mu = \mathbb{E}[\omega_{i,r}]$ and $\sigma^2 = \mathrm{Var}[\omega_{i,r}]$, then for all $a < \frac{6}{7}(\frac{1}{2} - \frac{1}{p})$,

$$P\left(\frac{T(n, \lfloor n^a \rfloor) - n\mu - 2\sigma n^{\frac{1+a}{2}}}{\sigma n^{\frac{1}{2} - \frac{a}{6}}} \le t\right) \to F_2(t)$$

If furthermore all the $\omega_{i,r}$ has finite moments of all orders, then the above convergence to F_2 holds for all $a < \frac{3}{7}$.

It is clear from this brief overview that the Tracy-Widom distribution is intertwined with many different fields of mathematics, from random matrix theory to combinatorics to analytic number theory, and that there are still many very interesting and exciting open questions to pursue.

8 Appendix for Code

Patience sorting

```
import numpy as np

def insert_pushDown(val, arr, col):
  # if no more room, make new row
  if (arr[arr.shape[0]-1][col] != 0):
    newRow = np.zeros((1,arr.shape[1]))
    arr = np.vstack((arr, newRow))
  # push everything down
  for i in range(arr.shape[0]-1,0,-1):
    arr[i][col] = arr[i-1][col]
  # insert value into opened space
  arr[0][col] = val
  return arr

def patienceSorting(permutation):
  # initialize with first value of permutation
  arr = np.full((1,1), permutation[0])
  for i in range(1,len(permutation)):
    inserted = 0
    for j in range(0,arr.shape[1]):
      # if less than number in col, insert and push down
      if (permutation[i] < arr[0][j]):
        arr = insert_pushDown(permutation[i], arr, j)
        inserted = 1
        break
    # greater than all previous top level numbers
    # need to create new column
    if (inserted == 0):
      newCol = np.zeros((arr.shape[0],1))
      newCol[0][0] = permutation[i]
      arr = np.hstack((arr, newCol))
  return arr

n = 70
tableau = patienceSorting(np.random.permutation(n)+1)
print(tableau)
```

Robinson-Schensted Algorithm

```
import numpy as np

def RSalgo_helper(val, arr, row):
  # if the row doesn't exist yet, add new row with val in left most spot
  if (row >= arr.shape[0]):
    newRow = np.zeros((1,arr.shape[1]))
    newRow[0][0] = val
    arr = np.vstack((arr, newRow))
    return arr

  inserted = 0
  for j in range(0,arr.shape[1]):
    # if arr[row][j] is 0 (empty), then we can insert
    if (arr[row][j] == 0):
      arr[row][j] = val
      inserted = 1
```

```
      break
    # if less than number in col, insert and bump
    if (val < arr[row][j]):
      oldVal = arr[row][j]
      arr[row][j] = val
      arr = RSalgo_helper(oldVal, arr, row+1)
      inserted = 1
      break
  # greater than all previous top level numbers => need to create new column
  # row == 0 only, because for row > 0, we will always have room
  if (row == 0 and inserted == 0):
    newCol = np.zeros((arr.shape[0],1))
    newCol[0][0] = val
    arr = np.hstack((arr, newCol))

  return arr

def RSalgo(permutation):
  # initialize with first value of permutation
  arr = np.full((1,1), permutation[0])
  for i in range(1,len(permutation)):
    arr = RSalgo_helper(permutation[i], arr, 0)
  return arr

n = 100
tableau = RSalgo(np.random.permutation(n-1)+1)
print(tableau)
```

RS Algorithm with Inverse

```
import numpy as np

def RSalgo_helper(valnum, val, arr, record, row):
  # if the row doesn't exist yet, add new row with val in left most spot
  if (row >= arr.shape[0]):
    newRow = np.zeros((1,arr.shape[1]))
    newRow[0][0] = val
    arr = np.vstack((arr, newRow))
    # ADDITIONS FOR RECORDING TABLEAU
    newRowRec = np.zeros((1,arr.shape[1]))
    newRowRec[0][0] = valnum
    record = np.vstack((record, newRowRec))
    return arr, record

  inserted = 0
  for j in range(0,arr.shape[1]):
    # if arr[row][j] is 0 (empty), then we can insert
    if (arr[row][j] == 0):
      arr[row][j] = val
      record[row][j] = valnum # ADDITION FOR RECORDING TABLEAU
      inserted = 1
      break
    # if less than number in col, insert and bump
    if (val < arr[row][j]):
      oldVal = arr[row][j]
      arr[row][j] = val
      arr, record = RSalgo_helper(valnum, oldVal, arr, record, row+1)
      inserted = 1
      break
  # greater than all previous top level numbers => need to create new column
  # row == 0 only, because for row > 0, we will always have roomm
```

```
  if (row == 0 and inserted == 0):
    newCol = np.zeros((arr.shape[0],1))
    newCol[0][0] = val
    arr = np.hstack((arr, newCol))
    # ADDITIONS FOR RECORDING TABLEAU
    newColRec = np.zeros((arr.shape[0],1))
    newColRec[0][0] = valnum
    record = np.hstack((record, newColRec))

  return arr, record

def RSalgo(permutation):
  # initialize with first value of permutation
  arr = np.full((1,1), permutation[0])
  record = np.full((1,1), 1)
  for i in range(1,len(permutation)):
    arr, record = RSalgo_helper(i+1, permutation[i], arr, record, 0)
  return arr, record

def cleanPrint(tableau):
  # replace all zeroes with space and print version that can be copy/pasted into spreadsheet
  tableau = tableau.astype('str')
  tableau[tableau == '0.0'] = '\t'
  for i in range(0,tableau.shape[0]):
    print(*tableau[i], sep =' ')

n = 100
# permut = np.random.permutation(n-1)+1
permut = [4,1,2,7,6,5,8,9,3]
print(permut)
insTableau, recTableau = RSalgo(permut)
print(insTableau)
print(recTableau)
# cleanPrint(insTableau)

def inverse_helper(ins, rec):
  row,col = np.unravel_index(np.argmax(rec), rec.shape)
  val = ins[row][col]
  ins[row][col] = 0
  rec[row][col] = 0
  row -= 1

  while row >= 0:
    # find index of max newval in array (ins[row] < val and != 0)
    noise= np.array(range(len(ins[row]))) * 1e-15
    # add increasing noise to array so argmax will find the rightmost occurence of 'True'
    col = np.argmax(np.logical_and(ins[row] < val, ins[row] != 0) + noise)
    newVal = ins[row][col]
    ins[row][col] = val
    val = newVal
    row -= 1

  return ins, rec, val

def inverse(ins, rec):
  permut = []
  n = np.max(rec)
  for i in range(0,int(n)):
    ins, rec, val = inverse_helper(ins,rec)
    permut.insert(0, val)
  return permut

```

```
permut = inverse(insTableau.copy(), recTableau.copy())
print(permut)

# for fun:
permut = inverse(recTableau, insTableau)
print(permut)
```

Rate of Convergence to TW

Remark: one must have a file `tracy_widom.py` with contents copied from https://gist.github.com/yymao/7282002 for a source to look up values of the Tracy-Widom distribution in order for this code to execute.

```
import numpy as np
import math
import tracy_widom
import matplotlib
import matplotlib.pyplot as plt

# further optmization patience sorting algorithm
def lengthOfLIS_PS_opt(permutation):
    # initialize with first value of permutation
    n = len(permutation)
    arr = np.zeros(int(3 * np.sqrt(n)))
    arr[0] = permutation[0]
    filledLength = 1
    for i in range(1,n):
        inserted = 0
        # maybe this will speed things up a little bit?
        tempVar = permutation[i]
        j = np.argmax(arr > tempVar)
        if (j > 0 or tempVar < arr[0]):
            arr[j] = tempVar
        else:
            arr[filledLength] = tempVar
            filledLength += 1

        # need to create new column
        if (filledLength >= len(arr) - 5):
            extraRoom = np.empty(np.sqrt(n))
            np.concatenate((arr, extraRoom), axis=1)

    return filledLength

tw = tracy_widom.TracyWidom()

nList = [100, 500, 1000, 2000, 5000, 10000, 20000, 50000, 100000, 1000000]
numSamples = 1000
sampleMilestones = [199, 399, 599, 799, 899, 999]
xGrid = np.arange(-4.5, 1.5, 0.005)

def empiricalF(xGrid, valArr):
    total = len(valArr)
    empiricalFArr = [1.0 * np.count_nonzero(valArr <= x)/total for x in xGrid]
    return empiricalFArr

maxDeviation = []
FArr = np.asarray([tw.cdf(x) for x in xGrid])
outF = open("./figures/maxDeviation.txt", "w")

for n in nList:
```

```
    plt.figure()
    plt.suptitle("n=" + str(n), fontsize=12)
    plt.plot(xGrid, FArr, label="true F")

    valArr = []
    for i in range(numSamples):
        perm = np.random.permutation(n) + 1
        permLIS = lengthOfLIS_PS_opt(perm)
        print(i, "lis is", permLIS)
        # calculate (LIS - 2sqrt(pi))/n^1/6
        val = (permLIS - 2 * math.sqrt(n)) / (n ** (1/6))
        valArr.append(val)

        if i in sampleMilestones:
            string = "empirical F with " + str(i+1) + " samples"
            plt.plot(xGrid, empiricalF(xGrid, valArr), label=string)

    plt.legend()
    string = "./figures/n=" + str(n) + ".pdf"
    plt.savefig(string)

    M = np.max(np.absolute(np.asarray(empiricalF(xGrid, valArr)) - FArr))
    print(str(n) + "\t" + str(M))
    print(str(n) + "\t" + str(M), file=outF)
    maxDeviation.append(M)

outF.close()
print(nList)
print(maxDeviation)
```

References

[1] Jinho Baik and Toufic M. Suidan. "A GUE central limit theorem and universality of directed first andlast passage site percolation". In: *International Mathematics Research Notices* 2005.6 (Jan. 2005), pp. 325–337. ISSN: 1073-7928. DOI: 10.1155/IMRN.2005.325. eprint: arXiv:math/0410042. URL: https://doi.org/10.1155/IMRN.2005.325.

[2] Thierry Bodineau and James Martin. "A Universality Property for Last-Passage Percolation Paths Close to the Axis". In: *Electron. Commun. Probab.* 10 (2005), pp. 105–112. DOI: 10.1214/ECP.v10-1139. URL: https://doi.org/10.1214/ECP.v10-1139.

[3] Miklós Bóna, Marie-Louise Lackner, and Bruce E. Sagan. "Longest Increasing Subsequences and Log Concavity". In: *Annals of Combinatorics* 21.4 (Dec. 2017), pp. 535–549. ISSN: 0219-3094. DOI: 10.1007/s00026-017-0365-x. eprint: https://arxiv.org/pdf/1511.08653.pdf. URL: https://doi.org/10.1007/s00026-017-0365-x.

[4] Percy Deift. *Universality for Mathematical and Physical Systems*. 2006. DOI: 10.4171/022. eprint: arXiv:math-ph/0603038. URL: https://www.semanticscholar.org/paper/Universality-for-mathematical-and-physical-systems-Deift/95b5709395919f426762ddb5c4da333b8209c568.

[5] J. Armando Domínguez-Molina. "The Tracy–Widom distribution is not infinitely divisible". In: *Statistics & Probability Letters* 123 (2017), pp. 56–60. ISSN: 0167-7152. DOI: https://doi.org/10.1016/j.spl.2016.11.029. eprint: https://arxiv.org/pdf/1601.02898.pdf. URL: http://www.sciencedirect.com/science/article/pii/S0167715216302826.

[6] Arno Kuijlaars. "Universality of distribution functions in random matrix theory". Workshop on Stochastic Eigen-Analysis and its Applications. Jul. 2006. URL: http://web.mit.edu/sea06/agenda/talks/Kuijlaars.pdf.

[7] Madan Lal Mehta. *Pure and Applied Mathematics*. 3rd edition. Institute of Mathematical Statistics Textbooks. Academic Press, 2004, p. 706. ISBN: 9780120884094; 9780080474113.

[8] Madan Lal Mehta. *Random Matrices*. 2nd edition. Pure and Applied Mathematics. Academic Press, 1990, p. 562. ISBN: 9781483295954.

[9] Dan Romik. *The surprising mathematics of longest increasing subsequences*. Vol. 4. Institute of Mathematical Statistics Textbooks. Cambridge University Press, New York, 2015, pp. xi+353. ISBN: 978-1-107-42882-9; 978-1-107-07583-2.

[10] Craig A. Tracy and Harold Widom. "Level-spacing distributions and the Airy kernel". In: *Comm. Math. Phys.* 159.1 (1994), pp. 151–174. URL: https://projecteuclid.org:443/euclid.cmp/1104254495.