

How “Timing Belt” Analysis With Precinct-Preserved Shuffle Tests Reveals Election Rigging

Declaration of Roger W. Fuller

Roger W. Fuller declares, under penalty of perjury, pursuant to 28 U.S.C. § 1746, that all following tabulations and graphs are true and correct to the State Certified Cast Vote Records and drawn to proper scale.

Executed on May 22, 2026.

A handwritten signature in black ink that reads "Roger W. Fuller". The signature is written in a cursive, flowing style.

Most voters vote what we call “**straight ticket**”- meaning a Republican will typically vote for Republican candidates across the ballot – especially in the Presidential, Senate and House races. Democrats also usually vote straight ticket. Historically it has been unusual to see more than a few percent of “split ticket” ballots, and in most cases, the numbers of Republican and Democrat split tickets are similar and cancel out – resulting in a small to zero net effect on outcomes.

In the following, I will use “**ST**” to refer to “**Straight Ticket**”.

When we observe low percentages of ST ballots, we look at the **timing of arrival of ST ballots in the tabulation process** which we call “**Timing Belt Analysis**”.

In Grand Junction in April 2021 we saw 45% ST ballots, with the number of Dem ST about what would be expected but Repub ST about 1/3 what would be expected.

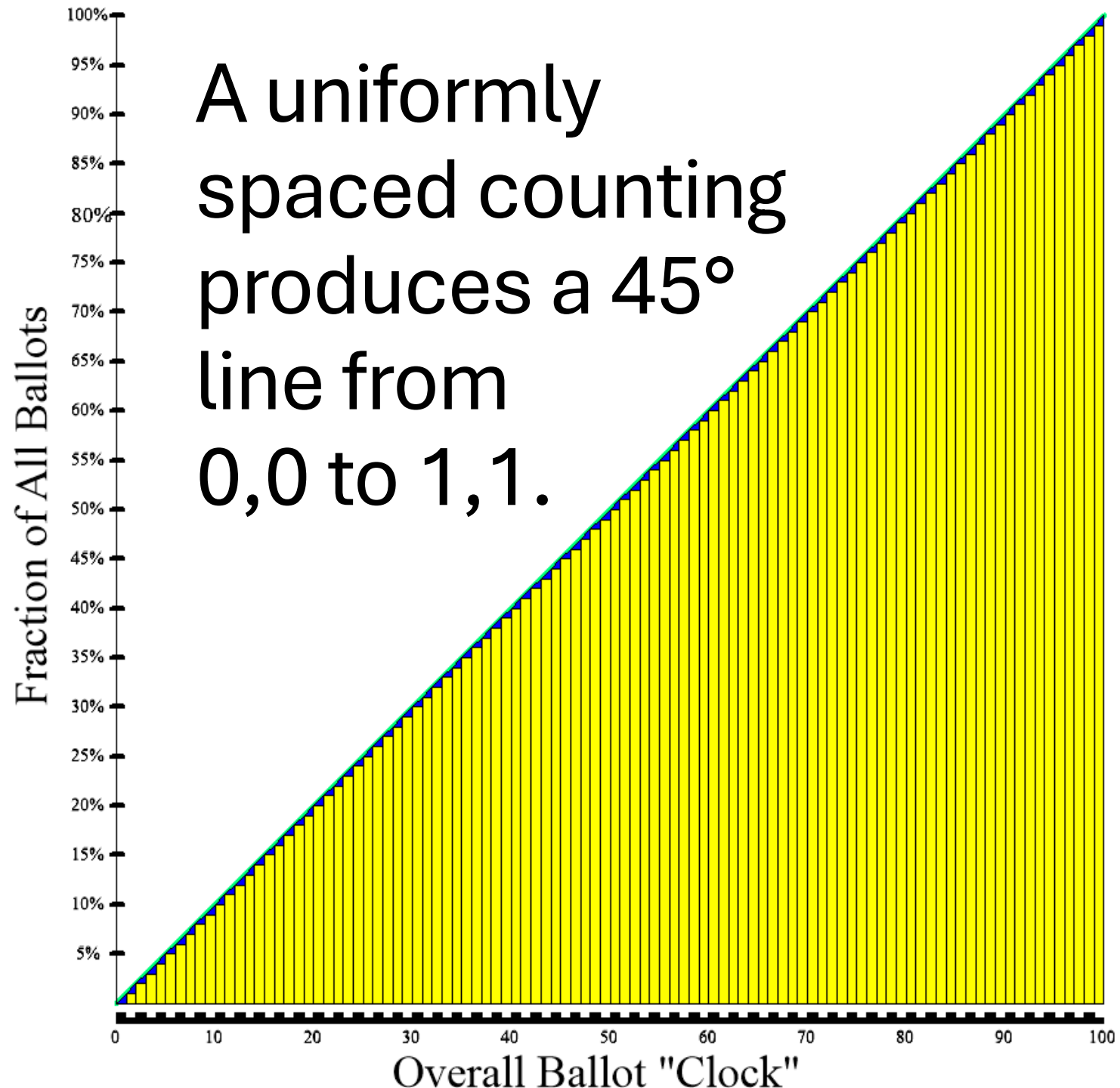
In Bexar Co. TX in 2024, on the three federal contests, Kamala voters were at 73% ST while Trump voters were at 52% ST. The results of analysis on these is shown.

This slideshow will explain how we analyze the ST “Timing Belt” for the original pattern of arrival, and how we follow that with “**Precinct Preserved Shuffle**” testing to precisely measure the chance that the original pattern could occur in an election free of software algorithmic manipulation.

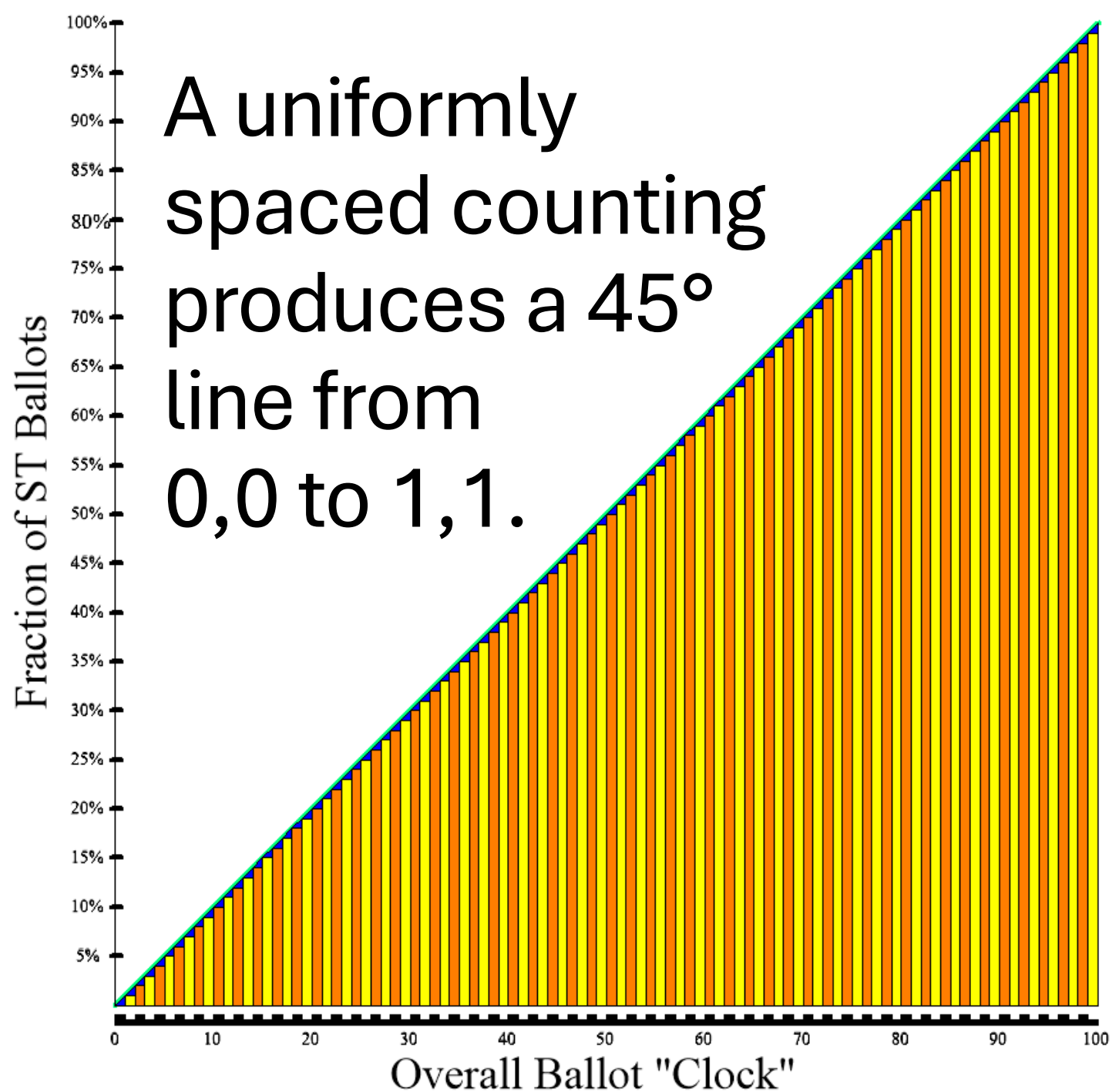
This is an image of a typical automotive Timing Belt. It has evenly spaced notches and “teeth” that engage in pulleys with matching notches and teeth on the crankshaft and camshaft(s) in overhead valve engines. The notches and teeth keep the belt and pulleys in a specific positional relationship so the cams open and close the valves at the right time.



Think of the timing belt cut and unrolled at the bottom of the graph as shown. Each notch space and tooth space represents one ballot as the ballots go through the tabulation process. The total number of ballots is $N = 100$ in these examples. Each ballot passing through is like the next "tick" of the ballot clock. Each ballot occupies an area that includes a rectangle and a triangle. The width of the area is one "tick" which equals $1/N$, in this example $1/100$ or 0.01, 1%. The height of the rectangle is the "time" at which it is processed. The height of each triangle is also $1/N$, 1%. Because the height increases by one unit of "time" per step, the green line showing the fraction of all ballots counted versus time is a 45° line from 0,0 to 1,1. The area under the line is $\frac{1}{2}$.



Now imagine a hypothetical election where exactly half of the ballots are straight-ticket ST ballots. One would expect them to be uniformly distributed in the mix. That would correspond to each tooth on the belt being an ST ballot and every notch being a non-ST ballot. In a real election with many thousands of ballots, there would be a bit of noise. One might see a string of ST ballots together at one point, and gaps where there are fewer for a bit, but the lumps would average out to produce the same “area under the curve” as the 45° right triangle area characteristic as seen here. We observed something unexpected when we started looking at the actual data from the Grand Junction CO April 2021 election. **The plot of the ST ballots forms a “bow” separated from the 45° line.**



Straight-Party Ballots And Timing Analysis

Mesa County 2020 Straight Ticket Numbers

	County		Grand Junction		County		Grand Junction	
	Federal	Federal	Federal	Federal	Fed & BOE & Co. Comm.	Fed & BOE & Co. Comm.	Fed & BOE & Co. Comm.	Fed & BOE & Co. Comm.
	Straight R	Straight D	Straight R	Straight D	Straight R	Straight D	Straight R	Straight D
	51257	26416	26025	16522	44615	20816	22862	13163
Ratio R to D:		1.940377		1.575172		2.143303		1.736838
Total Straight Ticket:		77673		42547		65431		36025
Total Presidential:		90614	Total GJ:	50025	Total Presidential:	90614	Total GJ:	50025
Fraction Straight Ticket:		0.857185		0.850515		0.722085		0.72014

This shows that Most Voters Voted Straight Ticket (both Dems and Repubs) and that Repubs voted almost 2:1 over Dems.

85% in the County and in Grand Junction voted straight ticket for Federal Races President, Senator, and House
72% in the County and in Grand Junction voted straight ticket on the 3 Federal races AND the two county races!

Grand Junction 2021 Straight Ticket Numbers

	Straight-Party	Four Rs	Four Ds	Candidate	Dist A	Dist D	Dist E	At-Large	Measures	2A	2B	2C
		2886	4898	Undervotes	1543	1987	1677	1870	Undervotes	614	348	989
	Total Straight party ballots:		7784									
	Fraction of 17189 total:		0.452848						Selected:	16575	16841	16200

This shows a **COMPLETELY DIFFERENT PICTURE** on straight ticket voting in Grand Junction six months later!

LESS THAN HALF (45.3%) voted straight ticket in all four City Council races – **55% SPLIT TICKETS!**

With 1600-1800 undervotes, approximately 16,500 ballots have votes cast in at least one candidate race.

If Repubs voted at 2:1 over Dems as seen in the 2020 election, one would expect to see about **5,300 Dem** straight tickets and **10,600 Repub** straight tickets. Yes, there are **4898 Dem** BUT ONLY **2886 Repub!**

WHAT HAPPENED TO 7500 REPUBLICAN STRAIGHT TICKETS?

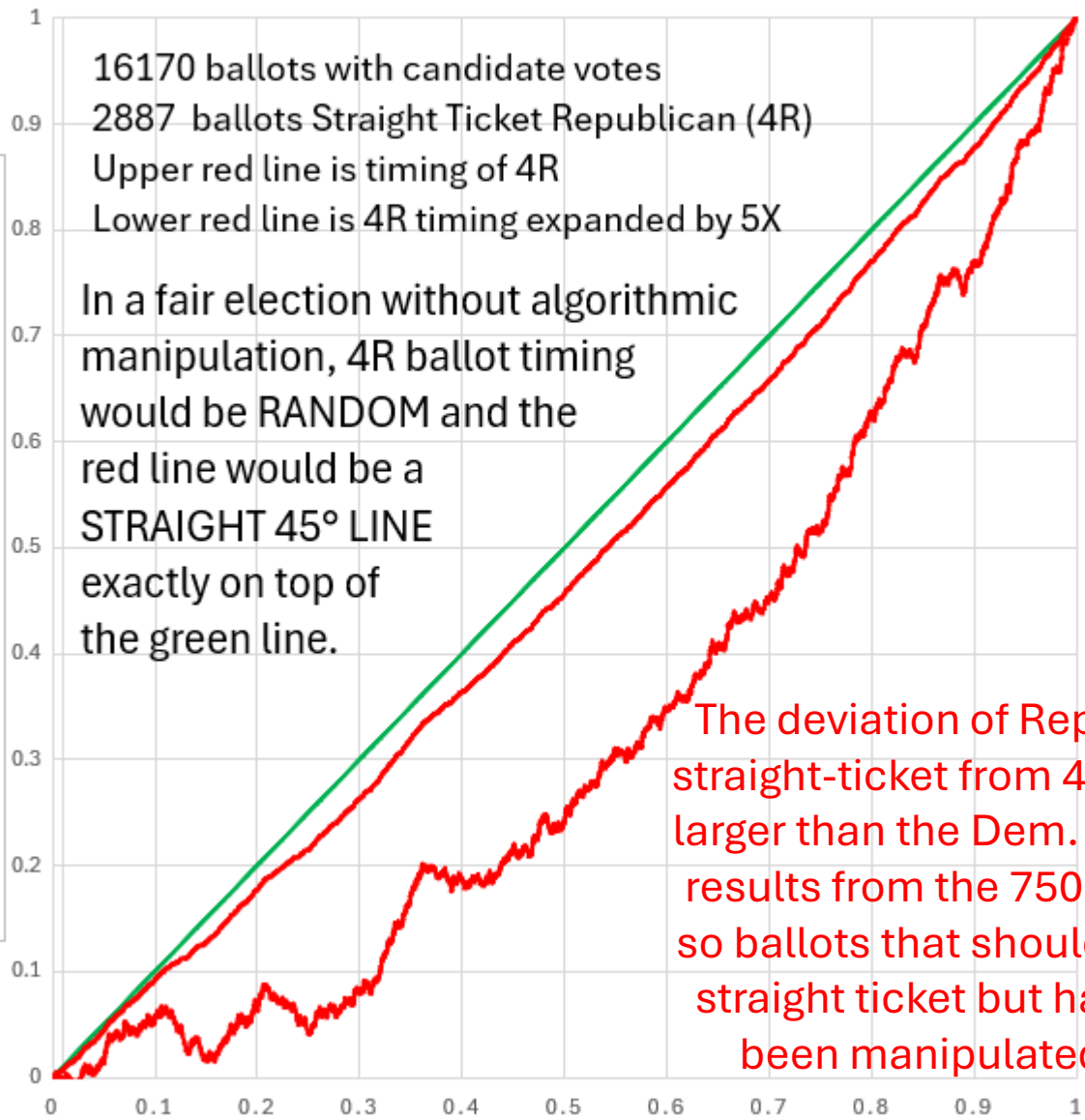
In the process of analyzing Cast Vote Records we see numbers of ST ballots that are very different from what one would expect. A local election in Grand Junction CO in 2021 was very unusual. In the **general election** in 2020, approximately **85% of both Dems and Repubs in Grand Junction voted straight tickets**, producing 42,547 ST ballots out of the total of 50,025 with a vote for Trump or Biden. **The ratio of Trump ST ballots to Biden ST ballots was 1.5 to one**, or 3:2, 60% to 40%. In the **2021 GJ election** there were **about 17,200 total ballots** cast. One would **expect about 14,600 ST** ballots at the 85% of the general 5 months earlier. But there were **only 7,784 ST** ballots. This is **45.3% compared to the 85% of the general**. Even more unusual were the raw numbers. At 1.5 to one seen in the general, one would **EXPECT** to see **8,760 Republican ST** ballots and **5,840 Democrat ST** ballots. But the **OBSERVED** numbers are **4,898 Democrat**, close to expectation, but **ONLY 2,886 Republican!**

WHAT HAPPENED TO THE MISSING 6,000 REPUBLICAN ST BALLOTS?

The “Timing Belts” tell us what happened!

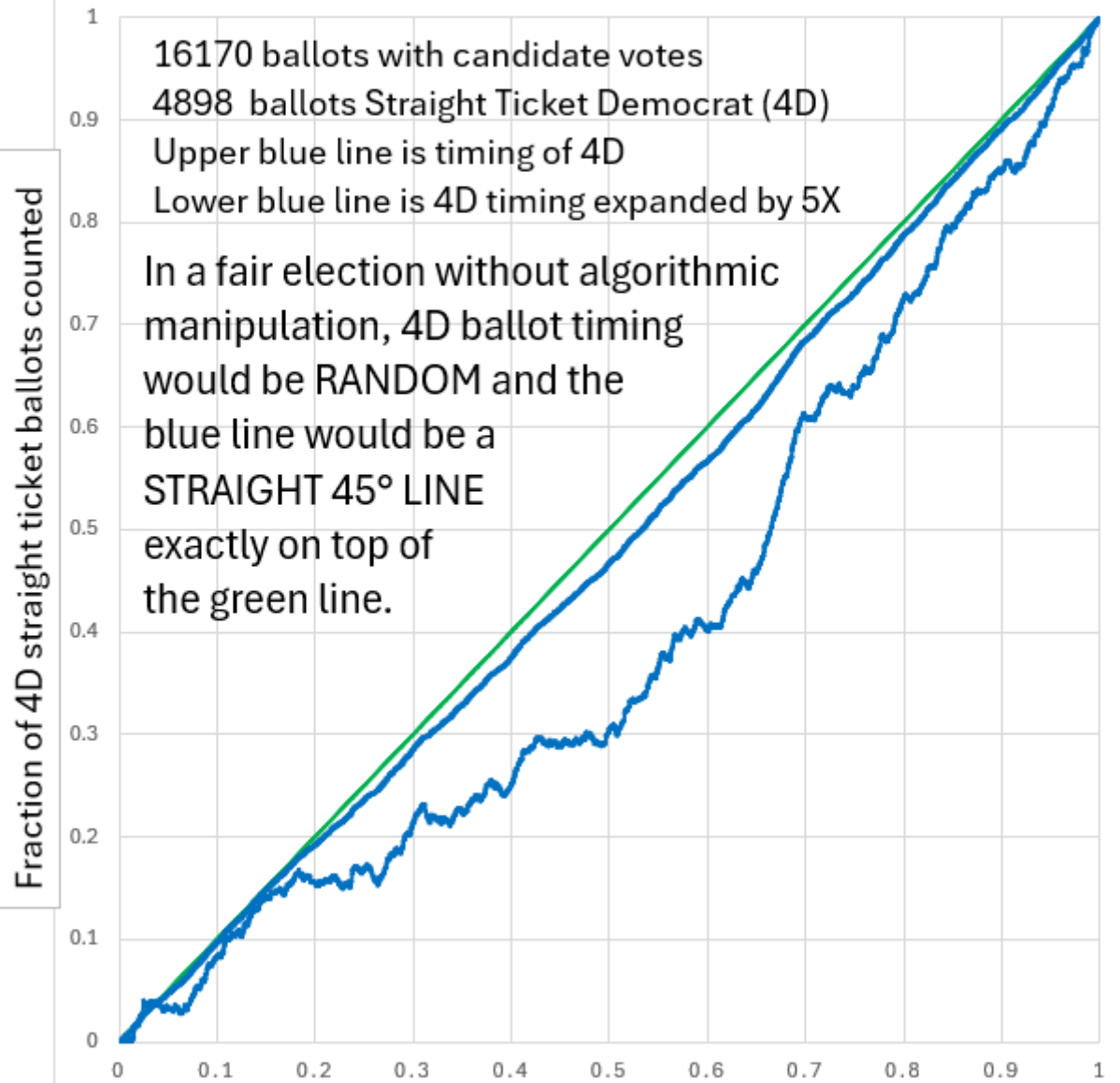
GRAND JUNCTION 2021 TIMING BELT STRAIGHT TICKET REPUBLICAN

45 Degree Line Str R% Str R% EX



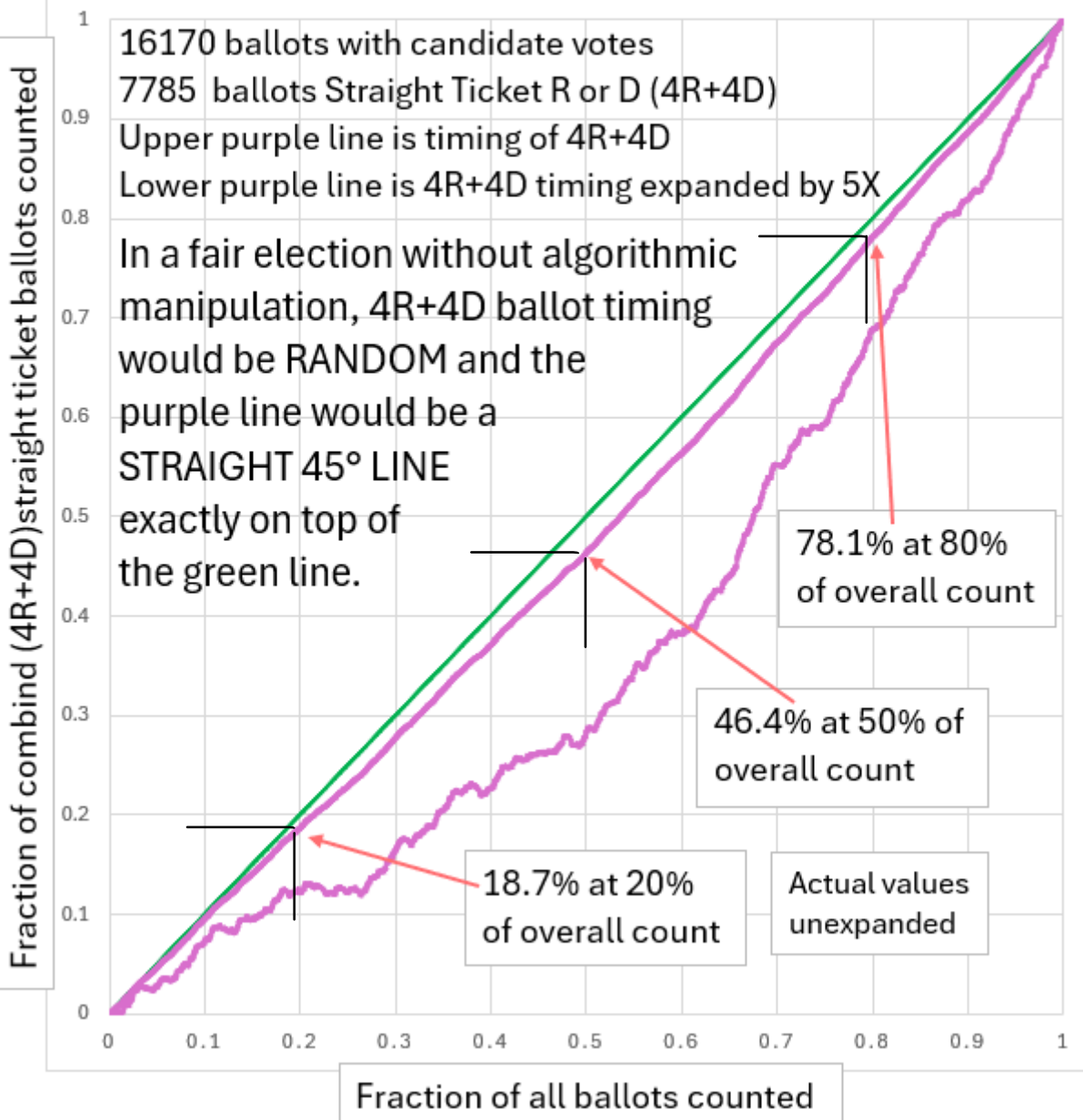
GRAND JUNCTION 2021 TIMING BELT STRAIGHT TICKET DEMOCRAT

45 Degree Line Str D% Str D% EX



GRAND JUNCTION 2021 TIMING BELT STRAIGHT TICKET REPUBLICAN OR DEMOCRAT

— 45 Degree Line — Str R|D% — Str R|D% EX



“Timing Belt” Analysis

This shows the pattern of the timing of the appearance of the straight ticket ballots in the cast vote record during the counting process. Consider the counting process to be like a clock where each tick of the clock is the next ballot counted – this is shown on the x-axis as the fraction of all 16,170 ballots counted from zero to one. Undervotes are excluded: only ballots that have a vote in at least one candidate race are included to make up the 16,170.

The purple lines represent the fraction of the 7,784 combined Repub and Dem straight-ticket ballots that have been counted at the “tick” of the “ballot time” clock of the overall sequence when the next ST ballot appears to be counted. The purple line close to the 45 degree line is the plot of the actual timing while the jagged purple line is expanded to aid visualization.

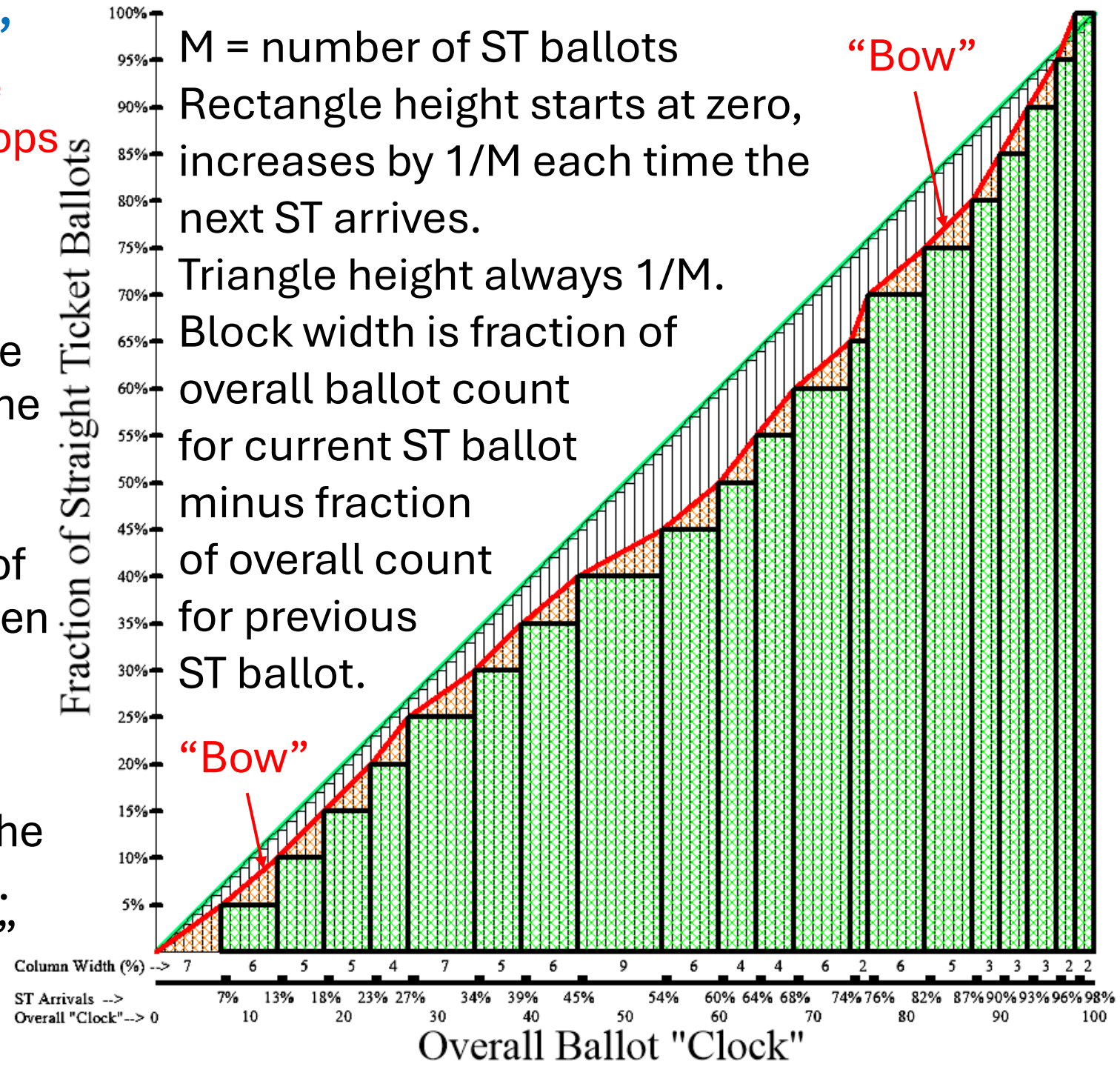
The horizontal and vertical lines show the story: When 20% of all ballots have been counted, only 18.7% of straight-ticket ballots have been counted, at 50% only 46.4% straight-ticket have been counted, and at 80% only 78.1% of the straight-ticket ballots have been counted. The rate of appearance of the straight-ticket ballots picks up at the end so the straight-ticket percentage arrives at 100% with the count of all ballots.

In an election with no manipulation, the purple line would fall on top of the 45 degree line with a bit of random noise.

Simplified ST "Timing Belt"

The objective: add up the area under the "bow" (red line) formed by the triangle tops

The total number of ballots N is 100, with M = 20 ST ballots. The ST ballots are shown by the "teeth" on the belt, empty slots for the non-ST ballots. The block width is the overall position of the current ST ballot minus the overall position of the previous ST ballot. The height of the rectangle is the fraction of ST ballots that have been counted when the current ST arrives. The rectangle height increases by 1/M each time a new ST arrives. The height of each triangle is 1/M. The first ST just adds the triangle and the last just the rectangle. The difference in area under the "bow" and area under the 45° line (0.5) measures the manipulation.



If the ST ballots came in evenly spaced throughout the counting process, the line would be EXACTLY on top of the 45° line, with a bit of noise. We connect all the triangle tops with a line which forma a “bow” shown in red on the diagram. We quantify the difference between the 45° line and the "bow" from the actual data in "Timing Belt" plots by measuring the area under each step in the “bow”. To get this area, we **add up the areas of all the rectangles and triangles under the stepped line**. This is an EXACT area because the line steps with each ballot, forming a triangle and rectangle at each step.

The height of each triangle is 1/(number of ST ballots). For this example, with 20 ST ballots, each triangle height is 0.05. The first ballot only contributes a triangle to the total area and the last ballot only contributes a rectangle to the total area.

The height of each rectangle is the fraction of ST ballots counted at the time the next ST ballot comes in, so the height of the first is 0.05, the second is 0.1, the third 0.15 etc. and the last rectangle has a height of 1.0.

The width of each column is the difference between the fraction of all ballots counted at the current ST ballot and the fraction of all ballots counted at the previous ST ballot.

In this example, the first ST ballot arrives at 7% of the overall count (width 0.07), the second arrives at 13% (width 0.13-0.07=0.06), the third arrives at 18% (width 0.18-0.13=0.05) etc. with the next to last at 96% and the last at 98% (width of the final rectangle 1.0-0.98=0.02). The total area of the triangles is 0.024, the total area of the rectangles is 0.407 and the total area is 0.431. From this we calculate the "Timing Belt Factor" **Tau** which is $\frac{1}{2}(\ln(\text{area}/1-\text{area}))$. For this example, the value of **tau** = $\frac{1}{2}(\ln(0.431/0.569)) = -0.1389$. **If the ST ballots were evenly spaced, the area would be exactly 0.5 and tau would be ZERO**. A negative tau (the "bow" BELOW the 45° line) indicates that the ST ballots came in "slow" relative to the overall count at the beginning and were caught up at the end. A positive value for tau (the "bow" ABOVE the 45° line) indicates that the ST ballots came in "fast" at the beginning and were “slow” at the end.

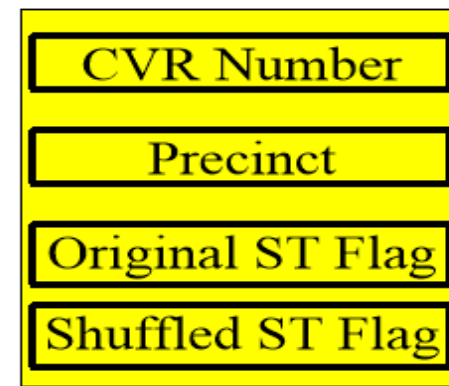
Summary:

1. Add up the area of the triangles and the rectangles
2. Calculate $\text{Tau} = \frac{1}{2}(\ln(\text{area}/(1-\text{area})))$
3. In a fair election, Tau will be small and can be positive or negative.
4. If ST ballots are evenly distributed as they would be in a fair election, the steps would be very close to the 45° line and tau would be VERY small

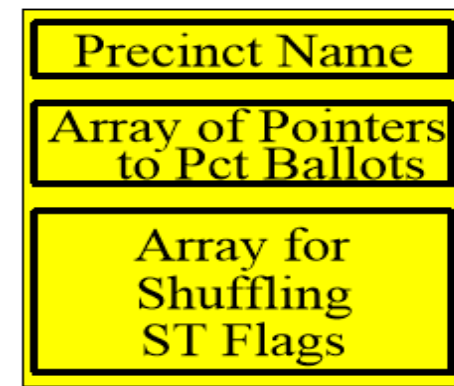
The Tau value calculated from the cast vote record is the “initial” Tau. In a normal election, the arrival timing would be random, there would be no organized “bow”, the tops of the triangles would follow the 45° line with a bit of noise, the resulting area would be $\frac{1}{2}$, and Tau would be zero. **In Grand Junction, the initial Tau was 0.0399.** This shows that the **specific pattern of the timing of ST ballot arrival in the tabulation process is highly unique and likely the result of algorithmic manipulation.** We need to **measure the degree of uniqueness of the pattern** to compare between a specific pattern of arrival of the ST ballots and what one should expect to see if the pattern was random – as if the actual votes were cast by humans. We do this by **performing a “random shuffle” of the ballots within the precincts to preserve the geographical distribution and demographics of the election and recalculating the Tau for the shuffled ballots.** If there is a pattern in the cast vote record, it disappears and the **tau value after shuffling is small.** We **repeat the “random shuffle and recalculate” up to 4,194,304 (two raised to the 22 power) times** to create a **normal distribution of random Tau values.** We can then **calculate a sigma value for the uniqueness of the original CVR pattern.**

The following slides describe the structure and operation of the computer program developed to run a desired number of cycles of calculate tau – shuffle – re-calculate tau

As each ballot is read from the CVR, a “Ballot Object” is created in memory. The object’s CVR number and precinct fields are filled in, both the Original ST flag and Shuffled ST flag fields are set to the original ST value, and the pointer to the object is added to the main pointer array. Setting the shuffled values to the original values upon creation results in the first tau calculation automatically producing the tau value of the original CVR pattern.

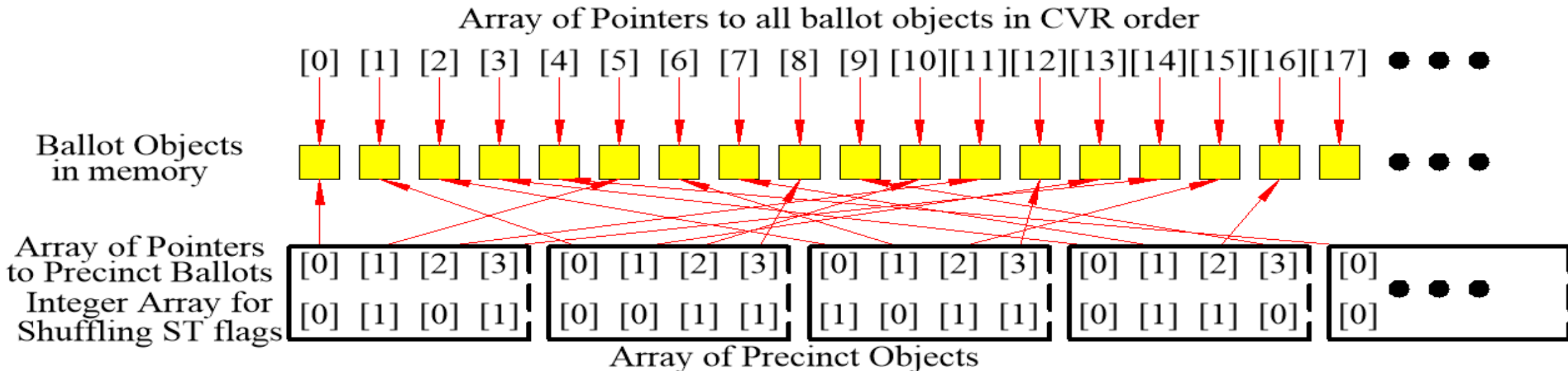


Ballot Object



Precinct Object

When the first ballot is processed, a “Precinct Object” is created and added to the array of precinct objects. The ballot pointer is added to the array of precinct ballot pointers. As subsequent ballots are read, after the ballot is created and the pointer has been added to the main array, the array of precincts is searched for the matching precinct. If the precinct already exists, the incoming ballot is added to it. If not, a new precinct is created and the incoming ballot pointer is added to the new precinct array.



When all the ballots have been read, the program knows the total number of ballots N and calculates the “clock tick” $1/N$. It also knows the total number of ST ballots M and calculates the triangle height and rectangle height increment $1/M$. The program next steps through the main pointer array, using the pointer to read the value of the shuffled ST flag from each ballot in turn. The program counts how many $1/N$ “ticks” from the beginning until it finds the first ballot with a “true” flag. This becomes the width of the first triangle. The area of the first triangle ($1/2$ the width times the height) is added to the total. The first ST ballot only produces a triangle area. The program continues stepping through the ballots and counts the number of $1/N$ ticks until it finds the next ST ballot. The number of ticks between is the width of the rectangle-plus-triangle block for the next ST ballot. The height of the rectangle for the next ST ballot is the top of the previous triangle. The areas of the triangle and the rectangle are added to the cumulative total area and the program continues to step through the main array. The height of each subsequent rectangle always increments by $1/M$. The program keeps track of how many ST ballots have been processed and when it finds the last ST ballot, there is no triangle, the width of the block is the number of “ticks” representing the remaining non-ST ballots and the height is the top of the previous ST ballot $(M-1)/M$. The rectangle area is calculated and added to the total area. The first cycle of calculation measures the area under the “bow” from the original cast vote record pattern. To amplify the difference between the area under the “bow” and the 45° line, we compute the value of tau as $\frac{1}{2}(\ln(\text{area}/(1-\text{area})))$ for each cycle. To perform a shuffle, the original flag values are copied from the precinct ballots to the shuffle array within the precinct. The shuffle arrays are processed using the “std::shuffle” operator provided with the ISO C++20 standard library. After the precinct arrays are shuffled, the flags are copied back to the shuffled ST flag field on the ballots. When the next tau is calculated by running through the overall ballot array array, the values of the ST flags are the result of the shuffle only within the precinct in a “precinct-preserved shuffle”. The tau value for each cycle is output to a spreadsheet. When all the cycles are complete, the mean and standard deviation for all the shuffled cycles is calculated, then the sigma value for the original cast vote record pattern is calculated against that normal distribution. We typically run 2^{22} (4,194,304) shuffles to demonstrate that the original CVR pattern does not occur in the shuffled distribution. This validates the result as being beyond 5 sigma (3.5 million). A 5-sigma finding on a new particle in physics gets a Nobel Prize.

	A	B	C	D	E	F	G	H	I	J	K
1	Group:	4R 4D									
2	Num In Gr	7784									
3	Tau orig:	0.039909									
4	Tau shuf:	0.003371	-0.002568	0.00475	0.008897	-0.00322	-0.00456	-0.00151	-0.00389		
524290		0.001465	-0.00274	0.011789	0.001041	0.005863	-1.6E-05	0.000305	-0.00476		
524291		0.009226	0.000866	0.002035	0.002589	-0.00237	0.006769	-0.00299	-0.00461		
524292											
524293	AVG TAU	0.001147	0.001128	0.001131	0.001132	0.001137	0.001138	0.001125	0.001135		Overall 0.001134
524294	STD DEV	0.004826	0.004821	0.00483	0.004824	0.004826	0.004822	0.004824	0.004825		0.004825
524295	SIGMA	8.031446	8.043799	8.027959	8.039085	8.033942	8.041299	8.040247	8.035304		8.036624
524296											
524297	Tau is 1/2(natural log(a / (1 - a))) where a is the sum of areas of Riemann trapezoids of group time vs overall time.										
524298											
524299	To perform the random shuffling of the ballots within precincts, the program uses functions from										
524300	the <random> standard C++ library provided with Microsoft Visual Studio 2022.										
524301	The following lines are incorporated in the program to implement the random shuffles:										
524302											
524303	#include<random> // provides access to "random" functions in library										
524304	std::random_device rd; // generates a non-deterministic random number										
524305	std::mt19937_64 g(rd); // generates a 64-bit Mersene Twister reordering with a large state (19937)										
524306	std::shuffle(v.begin(), v.end(), g); // performs the shuffle of entries in vector v using the 64-bit Mersenne Twister										
524307											
524308	The program creates a list of all ballots plus a list for each precinct. A vector is created for each precinct that has										
524309	a position for each ballot in the precinct. If the ballot has a straight party vote (4R's or 4D's) the value in the										
524310	vector at that position will be one, otherwise it will be zero. For each shuffle, the values in each precinct vector are										
524311	randomly shuffled within each precinct vector separately using the Mersenne Twister, then the tau value for the overall										
524312	set of ballots is recalculated and output to the spreadsheet. At the end, Excel formulas are inserted to calculate the										
524313	averages, standard deviations, and sigma (relative to the original order from the CVR) for each column and the overall.										

4,194,304 SHUFFLE Tau Analysis Grand Junction CO April 2021

In Grand Junction 2021, the original ordering produces $\tau_0 = 0.039909$.

The FIRST "random shuffle" of ballot data between the ballots **ONLY WITHIN EACH PRECINCT**, in a "Precinct Preserved Shuffle", gives a tau of **0.003371**, more than **10 times smaller** than the original. The program repeats this process **4,194,304 times**, outputting the value for tau after each shuffle. Note that some tau values are positive, some are negative. This is noise as result of the sample size of 16,170 ballots in five precincts. The standard deviation is **0.0048**.

The average tau value is **0.001134**, **THIRTY TIMES smaller than tau of the original order**. Against the standard deviation of 0.0048, the original tau is **8 SIGMA** out on the tail of the distribution!

8 SIGMA translates to a probability of **ONE** chance in **2.1657 QUADRILLION** that the **ORIGINAL SEQUENCE COULD OCCUR NATURALLY!** The annual Federal budget is \$4 trillion. **2.1657 QUADRILLION** is **540 YEARS** of Federal budget! This explains the **low number** of Republican straight-ticket ballots – **VOTES FLIPPED TO DEMS!**

	A	B	C	D	E	F	G	H	I	J	K	L
1	Group:	R+D straight; Races included: 3 of five.										
2	Num In Grp:	478613										
3	Tau orig:	-0.02148										
4	Tau shuf:	-0.00229	-0.00283	-0.00345	-0.00201	-0.00244	-0.00247	-0.00208	-0.0028			
524290		-0.00248	-0.0025	-0.00237	-0.00285	-0.0028	-0.00323	-0.00228	-0.00319			
524291		-0.00244	-0.00283	-0.00253	-0.00331	-0.00276	-0.00276	-0.00301	-0.00278			
524292		Note very small and negative Tau										Overall
524293	AVG TAU	-0.00267	-0.00267	-0.00267	-0.00267	-0.00267	-0.00267	-0.00267	-0.00267	-0.00267	-0.00267	-0.00267
524294	STD DEV	0.000369	0.000369	0.000369	0.000369	0.000369	0.000369	0.000369	0.000369	0.000369	0.000369	0.000369
524295	SIGMA	-50.9753	-50.9785	-51.0129	-50.9563	-51.0116	-50.9386	-50.9898	-51.0157	-50.9848		
524296												
524297	Tau is 1/2(natural log(a / (1 - a)) where a is the sum of areas of Riemann trapezoids of group time vs overall time.											
524298												
524299	To perform the random shuffling of the ballots within precincts, the program uses functions from											
524300	the <random> standard C++ library provided with Microsoft Visual Studio 2022.											
524301	The following lines are incorporated in the program to implement the random shuffles:											
524302												
524303	#include<random> // provides access to "random" functions in library											
524304	std::random_device rd; // generates a non-deterministic random number											
524305	std::mt19937_64 g(rd); // generates a 64-bit Mersene Twister reordering with a large state (19937)											
524306	std::shuffle(v.begin(), v.end(), g); // performs the shuffle of entries in vector v using the 64-bit Mersenne Twister											
524307												
524308	The program creates a list of all ballots plus a list for each precinct. A vector is created for each precinct that has											
524309	a position for each ballot in the precinct. If the ballot has a straight party vote (4R's or 4D's) the value in the											
524310	vector at that position will be one, otherwise it will be zero. For each shuffle, the values in each precinct vector are											
524311	randomly shuffled within each precinct vector separately using the Mersenne Twister, then the tau value for the overall											
524312	set of ballots is recalculated and output to the spreadsheet. At the end, Excel formulas are inserted to calculate the											
524313	averages, standard deviations, and sigma (relative to the original order from the CVR) for each column and the overall.											

4,194,304 SHUFFLE Tau Analysis
Bexar Co. TX 2024 General Election
Kamala 73% ST Trump 52% ST

In Bexar Co. TX in 2024, the original ordering produces $\tau_0 = 0.02148$. The FIRST "random shuffle" of ballot data between the ballots **ONLY WITHIN EACH PRECINCT**, in a "Precinct Preserved Shuffle", gives a tau of **0.002291**, about **8 times smaller** than the original. The program repeats this process **4,194,304 times**, outputting the value for tau after each shuffle. In Bexar, there are no positive values and the **standard deviation** is very small at **0.000369** due to the large sample of 747,382 ballots in 741 precincts.

The average tau value is **0.00267**. Against the standard deviation of 0.000369, The original tau is **51 SIGMA** out on the tail of the distribution!

51 SIGMA entered into a sigma-to-probability calculator returns **INFINITY!** The number is **BEYOND THE RANGE** of the IEEE standard 64-bit double-precision floating-point format used by our computers. This format can express numbers up to 1.0×10^{135} . For comparison, there are 10^{80} **PROTONS** in the **ENTIRE UNIVERSE!** The original ST timing **CANNOT OCCUR NATURALLY!** It could only be the result of algorithmic manipulation!

Conclusion: Timing Belt Analysis Proves Algorithmic Manipulation

In the 2021 Grand Junction Election, common sense would lead to the expectation that in the total of 16,000 total ballots, about 80% to 85% would be straight-ticket ballots. There **SHOULD BE** approximately 5000 Democrat straight-ticket ballots and approximately 10,000 straight-ticket Republican ballots. Yes, there are **4898 Democrat** straight-ticket ballots, but only **2886 Republican** straight ticket ballots, giving a total of 7784 straight ticket ballots overall. Both the Republican number and the overall number are about **7500 SHORT** of what would be expected.

This raises the question **“What happened to the 7500 Republican straight-ticket ballots?”**

In Bexar Co. TX, there are 73% Dem ST ballots but only 53% Repub ST ballots, and the timing belt analysis shows a **MORE-THAN-ASTRONOMICAL 50 SIGMA!**

The Timing Belt analysis shows that the timing of arrival of the straight-ticket ballots is shifted in a way that could only be done with software algorithms during the “counting” process. The only possible conclusion is that ST Democrat ballots were left unchanged while votes from Republican ST ballots were swapped to Democrats (maybe about 7500 in Grand Junction to produce the Democrat sweep?) **If the vote from one or two of the races on a Republican straight-party ballot is swapped, it is no longer a straight-party ticket!** This is the **ONLY WAY** this result could appear!

TINA WAS ALWAYS RIGHT!

THE GRAND JUNCTION 2021 ELECTION WAS COMPLETELY RIGGED!