Analysis of Michigan Precinct Timeseries Voting Data:

Table of Contents:

Table of Contents (cont.):

Table of Contents (cont.):

Abstract:

This paper documents the use of a quintic polynomial to assign Trump's percentage of the vote in each precinct in each county of Michigan. The initial quintic starts with two conjugate roots, with the other three roots at zero.

These qunitics are used to impersonate a natural distribution of the ratios (percentages) for Trump in each precinct.

The polynomial is then rotated via a simple rotation matrix on each coordinate generated by the polynomial, where x is the precinct number, and y is the percentage corresponding to the quintic.

The negative rotation of the initial polynomial induces a large subset of the precincts that are assigned to the second half the initial polynomial (the portion that is concave up) to now have lower percentage values for Trump for an equal sized subset of precincts assigned to the first half of the polynomial (the portion is concave down).

The entire polynomial is then vertically translated down the y-axis to subtract a flat percentage from Trump in every precinct.

However, to cover their tracks, the engineers of this algorithm do not permit these two subsets of the precincts to swap their positions on either side of the mean percentage, as such an obvious map between the domains of either polynomial would make an easy case for fraud. Rather, they simulated Brownian Motion inside of a trigonometrically shaped container (two cosine waves), such that the map between the domains of either polynomial appears as a scatter plot bounded between two standard bell curves.

Brownian motion - Wikipedia
https://en.wikipedia.org/wiki/Brownian_motion

Section I, Timestamp Set, Precinct Set and Ratio Set Definitions:

Let **T** be the set of all unique timestamps, let $|\mathbf{T}| = \alpha$; $\alpha = 141$
Let **P** be the set of all unique precincts, let $|\mathbf{P}| = \beta$; $\beta = 2499$

Definition of Cardinality:
https://brilliant.org/wiki/cardinality/#:~:text=The%20cardinality%20of%20a%20set,elements%20that%20are%20in%20it.

For all $\mathbf{p_d}$ in **P**:
Let $\mathbf{Z_d}$ be set of all total votes for each precinct on each timestamp, $|\mathbf{Z_d}| = \alpha$; $\alpha = 141$
Let **Y** be set of all $\mathbf{Z_d}$, $|\mathbf{Y}| = \beta$; $\beta = 2499$

By definition, **Y** is a dynamic matrix, of dimension β by α.

For all $\mathbf{p_d}$ in **P**:
Let $\mathbf{X_d}$ be set of all Trump votes for each precinct on each timestamp, $|\mathbf{X_d}| = \alpha$; $\alpha = 141$
Let **W** be set of all $\mathbf{X_d}$, $|\mathbf{W}| = \beta$; $\beta = 2499$

By definition, **W** is a dynamic matrix, of dimension β by α.

Let **H** be the set of all quotients for each precinct, on each timestamp, where the quotient is Trump's vote divided by the Total Vote.

Let **R** be the Ratio Set, where all quotients in **H** are reduced to their simplest form, such that there exists a surjection from **H** onto **R**.

Let $|\mathbf{R}| = \Psi$; $\Psi = 4789$

Section II, Definition of the Tensor Sets

Specific Definition of the Second Tensor Set.

Let **R** be the set of all ratios (Trump's total to the Grand Total for each precinct, over the entire timeline), and |**R**| = Ψ, and let **R** be ordered from least to greatest.

Let $\mathbf{T_2}$ be the Second Tensor, such that $\mathbf{T_2}$ contains all consecutive pairings of **R**.

Then let $\mathbf{T_{2,1}}$ be the first partition of the Second Tensor, such that $\mathbf{T_{2,1}}$ contains all pairing of **R**, leading with the odd element of **R**; likewise, let $\mathbf{T_{2,2}}$ be the second partition of the Second Tensor, such that $\mathbf{T_{2,2}}$ contains all pairings of the **R**, leading with the even element. All such tensors are ordered from least to greatest, in respect to their leading elements from **R**.

In general, let $\mathbf{T_{2,m}}$ denote the m^{th} partition of the Tensor.

Then let $\mathbf{T_{2,m,k}}$ denote the k^{th} pairing, which is the k^{th} element of $\mathbf{T_{2,m}}$.

Let **X** be the set of timestamps for which a ratio, r_i, in $\mathbf{T_{2,m,k}}$, is reported, and let |**X**| = Ω.

Then let $\mathbf{T_{2,m,k,x}}$ denote the x^{th} timestamp for which r_i is reported, such that $1 \leq x \leq \Omega$, containing all precincts reporting r_i at time x.

Finally let **Y** be the set of precincts that simultaneously report r_i at time x; and let |**Y**|= λ.
Now let $\mathbf{T_{2,m,k,x,y}}$ denote the y^{th} precinct that reports r_i, within the timestamp x, such that $1 \leq y \leq \lambda$.

General Definition of the Tensor Set.

Let **R** be the set of all ratios (Trump's total to the Grand Total for each precinct, over the entire timeline), and |**R**| = Ψ, and let **R** be ordered from least to greatest.

Let T_g be the g^{th} tensor, where **G** is the set of all tensors, |**G**| = Γ , such that T_g contains all groupings of consecutive ratios of length *g* in respect to **R**.

Let $\mathbf{T}_{\mathbf{g,m}}$ denote the m^{th} partition of the Tensor, such that:

$\mathbf{T}_{\mathbf{g,m}}$ is the m^{th} partition, containing all groupings of consecutive ratios such that the difference in the indices of the leading and the trailing ratio in the group is equal to *g*, with the index leading ratio of the group, r_i, being congruent to *m* mod *g*, and the index of the trailing ratio of the group being congruent to *(m+g-1)* mod *g*.

Then let $\mathbf{T}_{\mathbf{g,m,k}}$ denote the k^{th} grouping, which is the k^{th} element of $\mathbf{T}_{\mathbf{g,m}}$.

Let **X** be the set of timestamps for which a ratio, r_i, in $\mathbf{T}_{\mathbf{g,m,k}}$, is reported, and let |**X**| = Ω.

Then let $\mathbf{T}_{\mathbf{g,m,k,x}}$ denote the x^{th} timestamp for which r_i is reported, such that $1 \leq x \leq \Omega$, containing all precincts reporting r_i at time *x*.

Finally let **Y** be the set of precincts that simultaneously report r_i at time *x*; and let |**Y**|= λ.
Now let $\mathbf{T}_{\mathbf{g,m,k,x,y}}$ denote the y^{th} precinct that reports r_i, within the timestamp *x*, such that $1 \leq y \leq \lambda$.

Specific Definition of the First Tensor Set.

Let **R** be the set of all ratios (Trump's total to the Grand Total for each precinct, over the entire timeline), and |**R**| = Ψ, and let **R** be ordered from least to greatest.

Let T_1 be the *1*st tensor.

Let $\mathbf{T_{1,1}}$ denote the only partition of the First Tensor, such that:

$\mathbf{T_{1,1,k}}$ denotes the *k*th ratio, since each "*group*" contains a single ratio.

Let **X** be the set of timestamps for which a ratio, r_i, in $\mathbf{T_{1,1,k}}$, is reported, and let |**X**| = Ω.

Then let $\mathbf{T_{1,1,k,x}}$ denote the *x*th timestamp for which r_i is reported, such that $1 \leq x \leq \Omega$, containing all precincts reporting r_i at time *x*.

Finally let **Y** be the set of precincts that simultaneously report r_i at time *x*; and let |**Y**|= λ.
Now let $\mathbf{T_{1,1,k,x,y}}$ denote the *y*th precinct that reports r_i, within the timestamp *x*, such that $1 \leq y \leq \lambda$.

Section III, Definitions of Stasis Sets:

Let **T** be the set of all unique timestamps, let $|\mathbf{T}| = \alpha$; $\alpha = 141$
Let **P** be the set of all unique precincts, let $|\mathbf{P}| = \beta$; $\beta = 2499$

Definition of Cardinality:
https://brilliant.org/wiki/cardinality/#:~:text=The%20cardinality%20of%20a%20set,elements%20that%20are%20in%20it.

For all $\mathbf{p}_d$ in **P**:
Let $\mathbf{Z}_d$ be set of all total votes for each precinct on each timestamp, $|\mathbf{Z}_d| = \alpha$; $\alpha = 141$
Let **Y** be set of all $\mathbf{Z}_d$, $|\mathbf{Y}| = \beta$; $\beta = 2499$

By definition, **Y** is a dynamic matrix, of dimension β by α.

For all $\mathbf{p}_d$ in **P**:
Let $\mathbf{X}_d$ be set of all Trump votes for each precinct on each timestamp, $|\mathbf{X}_d| = \alpha$; $\alpha = 141$
Let **W** be set of all $\mathbf{X}_d$, $|\mathbf{W}| = \beta$; $\beta = 2499$

By definition, **W** is a dynamic matrix, of dimension β by α.

Let **H** be the set of all quotients for each precinct, on each timestamp, where the quotient is Trump's vote divided by the Total Vote.

Let **R** be the Ratio Set, where all quotients in **H** are reduced to their simplest form, such that there exists a surjection from **H** onto **R**.

Let $|\mathbf{R}| = \Psi$; $\Psi = 4789$

For each precinct, p_i, in **P,** let $\mathbf{S}_i$ be the set of timestamps from **T,** such that the precinct updated its vote tabulations, with those timestamps ordered from least to greatest, $|\mathbf{S}_i| = \eta \leq \alpha$.

$\forall\ p_i \in \mathbf{P}$, let $\mathbf{S}_i \subset \mathbf{T}$, such that the total numbers of votes, z_{j-1} , at time t_{j-1} is not equal to z_j , at time t_j . The timestamp t_j is the stamp that is injected into $\mathbf{S}_i$.

However, $\mathbf{S}_i$ shall start with the first timestamp that this precinct recorded information in its tabulations.

Set $\mathbf{S}_i$ shall be known as the Statis Set of p_i.

Let **V** be the set of all $\mathbf{S}_i$, $|\mathbf{V}| = \beta$; by definition **V** is a dynamic two-dimensional array.

Definition of Present Stasis:

For each timestamp, t_j , for any precinct p_i, the first timestamp in $\mathbf{S}_i$ that is less than or equal to t_j, is known as the Present Stasis, since that precinct has yet to update its tabulations.
This value is known as $s_{i,f}$

This will often be referred to as the "Time of Release."

Definition of Prior Stasis:

The Prior Stasis, for each timestamp, t_j, for any p_i, is the timestamp preceding $s_{i,f}$, since this was the first before last timestamp at which the precinct had updated its tabulations.
This value is simply $s_{i,f-1}$. If f is equal to 1, then this value is zero.

This will often be referred to as the "Time of Seizure."

Definition of Penultimate Stasis:

The Penultimate Stasis, for each timestamp, t_j, for any p_i, is the timestamp preceding $s_{i,f-1}$, since this was the second before last timestamp at which the precinct had updated its tabulations.
This value is simply $s_{i,f-2}$. If *f* is equal to 1, then this value is equal to -10,000, to denote a null entry.

This will often be referred to as the "Last Potential Free State."

Definition of Future Stasis:

The Future Stasis, for each timestamp, t_j, for any p_i, is the timestamp succeeding $s_{i,f}$, since it is the next timestamp at which the precinct will have updated its tabulations.

This value is simply $s_{i,f+1}$. If *f* is equal to η, then this value is equal to +10,000, to denote a null entry.

This will always be referred to as the Future Stasis.

Chapter One; Polynomial Regression of Each County.

Section I, General Definitions,

Regression of the Largest Percentage Drop for Trump in each Precinct

For each precinct, p_i, in **P,** let $\mathbf{C_j}$ be the subset of **P**, such that all for p_i in $\mathbf{C_j}$, p_i is from the same county.

Let **D** be the set of all $\mathbf{C_j}$, where |**D**| is the number of counties in the Fathersheet, |**D**| = 20.

For each precinct, $c_k \in \mathbf{C_j}$, let $\mathbf{E_j}$ be the set containing the largest negative drop (in absolute value) in Trump's percentage for each precinct, c_k, in that County. If the precinct has no negative drop on record, then it is removed from $\mathbf{C_j}$, and it's least positive increase is removed from $\mathbf{E_j}$, maintaining the bijection between the sets.

Let $\mathbf{E_j}$ be ordered from least to greatest, and remove outliers from this set, where the difference between two consecutive indexes of $\mathbf{E_j}$ expands rapidly in respect to the prior differences between the consecutive indexes of $\mathbf{E_j}$; such that $|\mathbf{E_j}| = e$ (let it be noted that even with the inclusions of the few remote outliers, the regressions remain strong degree 10 polynomials).

Let the number of outliers removed from $\mathbf{E_j}$ be equal to ε; let $|\mathbf{E_{j-\varepsilon}}| = e - \varepsilon$.

Let $\mathbf{F_{j-\varepsilon}}$ be the precincts corresponding to each index in $\mathbf{E_{j-\varepsilon}}$; let $|\mathbf{F_{j-\varepsilon}}| = e - \varepsilon$.

Let each index of $\mathbf{F_{j-\varepsilon}}$ be graphed as the x-axis input for the polynomial regression.
Let each index of $\mathbf{E_{j-\varepsilon}}$ be graphed as the y-axis input for the polynomial regression.

There exists a bijection between both sets.

We will use the following calculators for these regressions:
Polynomial Regression Calculator (stats.blue)
https://arachnoid.com/polysolve/

After each regression is determined, when then calculate the amount of votes that were deleted from Trump, by adding the Percentage on the algorithmically seized timestamp to the net loss that occurred, and then multiplying the natural percentage against the Total Number of Votes on the seized timestamp.

Random Number Generation of Maximum Negative Drops

Let **T** be the set of all unique timestamps (in this simulation), let $|\mathbf{T}| = \boldsymbol{\alpha}$; $\boldsymbol{\alpha} = 100$
Let **P** be the set of all unique precincts (in this simulation), let $|\mathbf{P}| = \boldsymbol{\beta}$; $\boldsymbol{\beta} = 100$

Let $\mathbf{A}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ be the array of the total number of votes added to each Candidate **A** for each timestamp in each precinct.

Let $\mathbf{B}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ be the array of the total number of votes added to each Candidate **B** for each timestamp in each precinct.

$\forall\ t_{\mathrm{i,j}} \in \mathbf{T}$, let $a_{\mathrm{i,j}}$ and $b_{\mathrm{i,j}}$ be a random number bounded between m and n.

Let $\mathbf{C}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ be the total number of votes added to each precinct on each timestamp, such that $c_{\mathrm{i,j}} = a_{\mathrm{i,j}} + b_{\mathrm{i,j}}$.

Let $\mathbf{F}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ be the cumulative number of votes for Candidate **A** for each timestamp (in each precinct):.
Let $\mathbf{G}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ be the cumulative number of votes for Candidate **B** for each timestamp (in each precinct):

Let $\mathbf{H}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ be the cumulative number of votes for each precinct on each timestamp, such that $h_{\mathrm{i,j}} = f_{\mathrm{i,j}} + g_{\mathrm{i,j}}$.

Let $\mathbf{R}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ be the ratio $(f_{\mathrm{i,j}}) / (h_{\mathrm{i,j}})$.

Let $\square_{\boldsymbol{\alpha}\text{-}\mathbf{1},\boldsymbol{\beta}}$ be the difference between $(r_{\mathrm{i,j}}) - (r_{\mathrm{i-1,\,j}})$.

Let $\boldsymbol{\Delta}_{\boldsymbol{\beta}}$ be the one dimensional array containing the least value of $\square_{\mathbf{x}}$ for each $\square_{\mathrm{y}}$.

Now let $\boldsymbol{\Theta}_{\boldsymbol{\beta}}$ be $\boldsymbol{\Delta}_{\boldsymbol{\beta}}$ ordered from greatest to least, now let $\boldsymbol{\Gamma}_{\boldsymbol{\beta}}$ be the absolute value of each entry in $\boldsymbol{\Theta}_{\boldsymbol{\beta}}$.
This will be the response array in the regression.

Now Let $\mathbf{P}_{\boldsymbol{\beta}}'$ be the reordered set of the precincts corresponding the entries in $\boldsymbol{\Gamma}_{\boldsymbol{\beta}}$.
This will be the explanatory array in the regression.

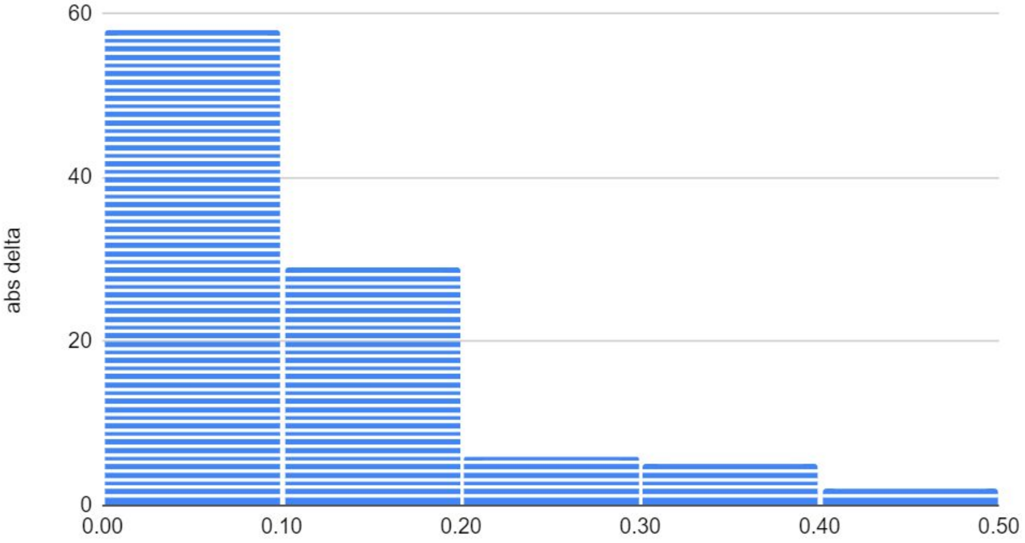
In order to allow for a dynamic number of precincts and timestamps, we used the following formula for the Precincts against the Timestamps.

Make a list from 1 to (β)($\boldsymbol{\alpha}$).
=CONCATENATE("Precinct #",MOD(A2,100)+1)
Or more generally:
=CONCATENATE("Precinct #",MOD(A2, β)+1)

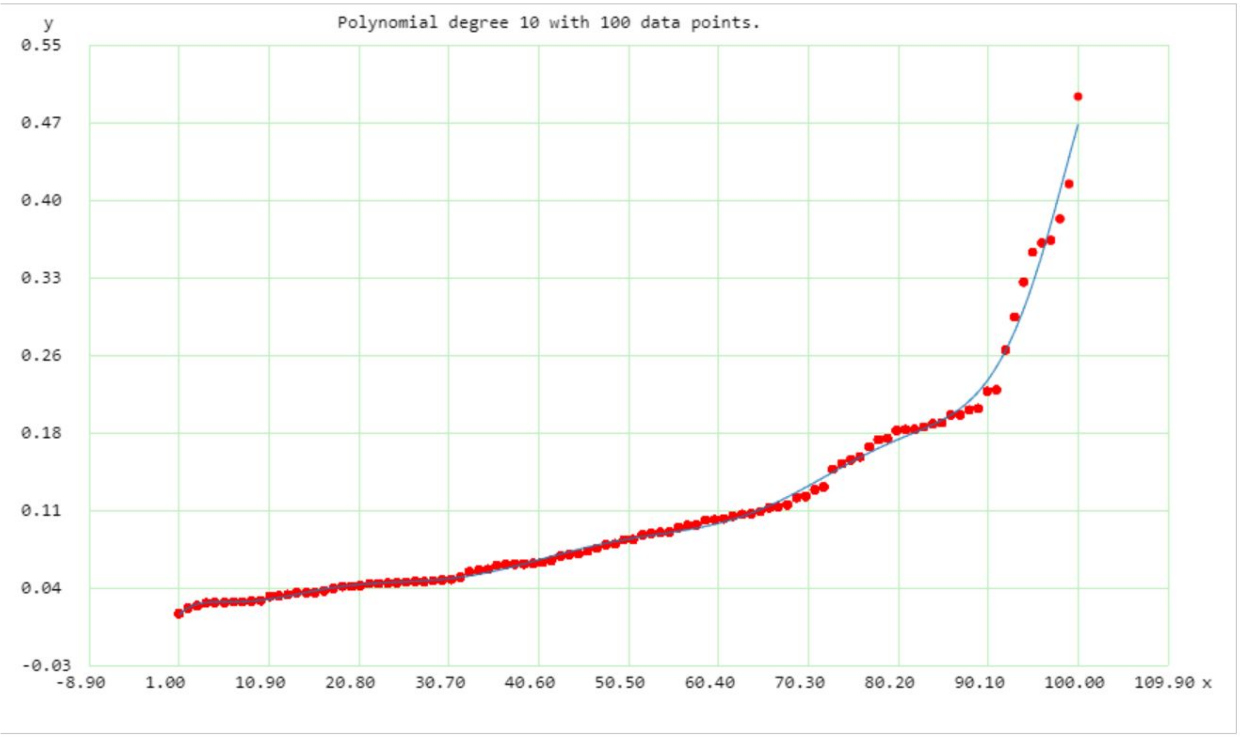
Where each precinct, p_i, starts on timestamp p_i, and then each successive timestamp for each precinct is equal to:
$p_i+k\beta$, for $\boldsymbol{\alpha}$ iterations, $0\leq k \leq \boldsymbol{\alpha}$, thus, the last timestamp for each precinct is equal to $p_i+(\boldsymbol{\alpha}-1)(\beta)$.

Then resort both the timestamp and precinct columns by Precinct Ascending only.
Now all consecutive timestamps congruent to p_i mod β correspond to each precinct's series of timestamps.

The resulting histogram, showing the distribution of these maximum deltas, per precinct, is as follows:



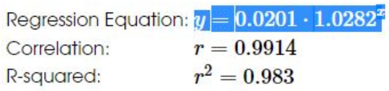
The resulting natural graph of the regression of the largest difference in each precinct, is as follows:



Polynomial degree 10, 100 x,y data pairs; https://arachnoid.com/polysolve/
Correlation coefficient = 0.9938251306192952
Standard error = 0.0077356827443010685

6.8307183817297931e-004,	
1.4405486445013022e-002,	x
-3.3889699870534545e-003,	x^2
3.9577071263138569e-004,	x^3
-2.5076336931664835e-005,	x^4
9.3844728545769911e-007,	x^5
-2.1636827802877174e-008,	x^6
3.1024318250517910e-010,	x^7
-2.6916604801642070e-012,	x^8
1.2920500959517656e-014,	x^9
-2.6310766438204999e-017,	x^10

Here is the exponential regression; Exponential Regression Calculator (stats.blue)



Regression Equation: $y = 0.0201 \cdot 1.0282^x$
Correlation: $r = 0.9914$
R-squared: $r^2 = 0.983$

.

Pure Random Number Generator, for a single Candidate

Let **T** be the set of all unique timestamps (in this simulation), let $|\mathbf{T}| = \boldsymbol{\alpha}$; $\boldsymbol{\alpha} = 30$
Let **P** be the set of all unique precincts (in this simulation), let $|\mathbf{P}| = \boldsymbol{\beta}$; $\boldsymbol{\beta} = 378$

Let $\mathbf{A}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ be the array of the total number of votes for each Candidate **A** for each timestamp in each precinct.

$\forall\ t_{i,j} \in \mathbf{T}$, let $a_{i,j}$ be a random number bounded between *m* and *n* (this simulation is not cumulative, as the prior simulation was).

Let $\square_{\boldsymbol{\alpha}\text{-}1,\boldsymbol{\beta}}$ be the difference between $(a_{i,j}) - (a_{i-1,j})$.

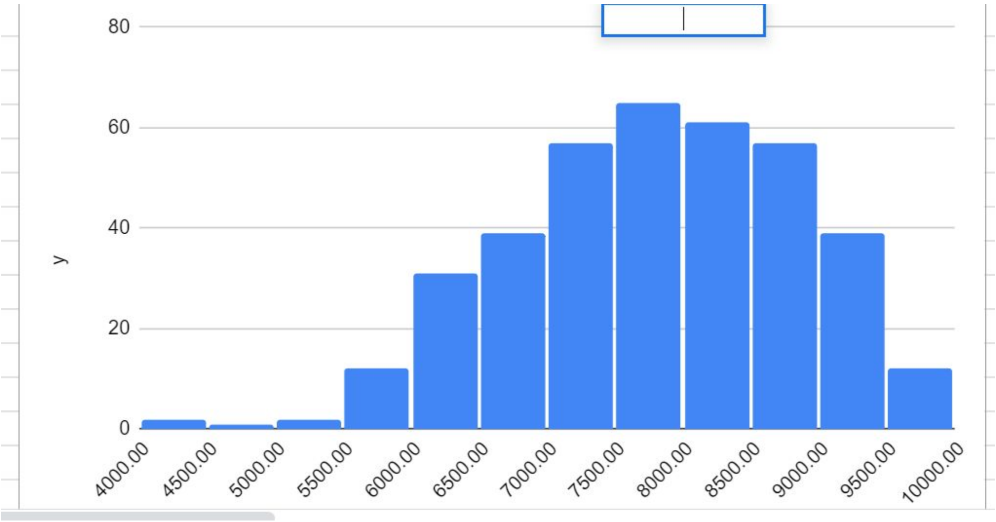
Let $\boldsymbol{\Delta}_{\boldsymbol{\beta}}$ be the one dimensional array containing the least value of $\square_{x}$ for each $\square_{y}$.

Now let $\boldsymbol{\Theta}_{\boldsymbol{\beta}}$ be $\boldsymbol{\Delta}_{\boldsymbol{\beta}}$ ordered from greatest to least, now let $\boldsymbol{\Gamma}_{\boldsymbol{\beta}}$ be the absolute value of each entry in $\boldsymbol{\Theta}_{\boldsymbol{\beta}}$. This will be the response array in the regression.

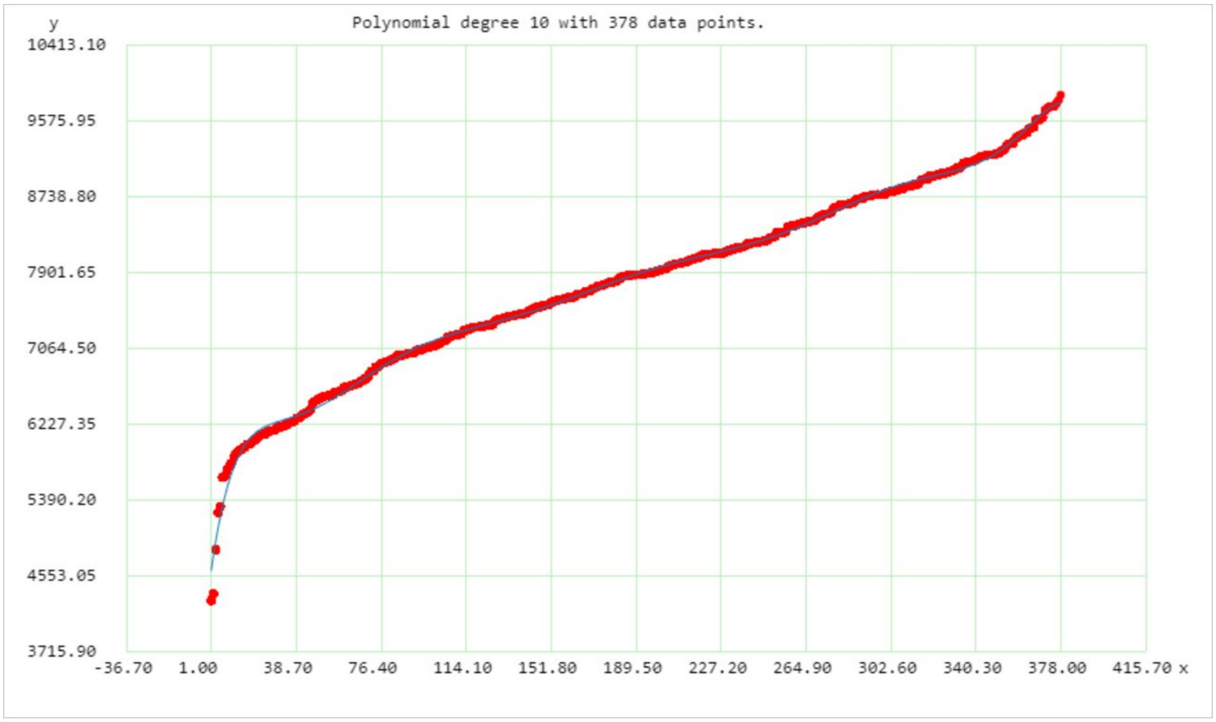
Now Let $\mathbf{P}_{\boldsymbol{\beta}}'$ be the reordered set of the precincts corresponding the entries in $\boldsymbol{\Gamma}_{\boldsymbol{\beta}}$.
This will be the explanatory array in the regression.

In order to allow for a dynamic number of precincts and timestamps, we simply start with matrix of size: $(\beta)(\boldsymbol{\alpha})$.

The resulting histogram, showing the distribution of these maximum deltas, per precinct, is as follows:



The resulting natural graph of the regression of the largest difference in each precinct, is as follows (take note that this graph has a strong linear regression, and the decic regression has nine changes in concavity).



Polynomial degree 10, 100 x,y data pairs; https://arachnoid.com/polysolve/
Correlation coefficient = 0.9938251306192952
Standard error = 0.0077356827443010685

Polynomial degree 10, 378 x,y data pairs.
Correlation coefficient = 0.9977351615620487
Standard error = 50.60873417534612

4.4192244077516043e+003,
1.9473592971901033e+002,
-8.7586429683474236e+000,
2.1208772793318764e-001,
-2.9362067059432945e-003,
2.4904101349234941e-005,
-1.3387703247752521e-007,
4.5758033453077935e-010,
-9.6298631140299514e-013,
1.1371162181874423e-015,
-5.7623398299408060e-019

Difference Between Two Naturally Distributed Sets on consecutives timestamps (relative to each precinct):

Let **T** be the set of all unique timestamps (in this simulation), let $|\mathbf{T}| = \boldsymbol{\alpha}$; $\boldsymbol{\alpha} = 2$
Let **P** be the set of all unique precincts (in this simulation), let $|\mathbf{P}| = \boldsymbol{\beta}$; $\boldsymbol{\beta} = 10{,}000$

Let the standard deviation, $\boldsymbol{\sigma}$, be equal to 1; and let p_{5016} be the mean at zero, $\boldsymbol{\mu} = 0$.

Let $\mathbf{R}_0$ be the set of all ratios (in this simulation), let $|\mathbf{R}| = \boldsymbol{\beta}$; $\boldsymbol{\beta} = 10{,}000$, such that the ratio of each precinct corresponds to its natural distance from p_{5016}.

Now transform $\boldsymbol{\sigma}$ to 10%, applying the formula $(10r_k/100)$;
Now transform $\boldsymbol{\mu}$ to 65%, applying the formula $0.65+(10r_k/100)$.
Let this transformation be equal to $\mathbf{R}_1$.

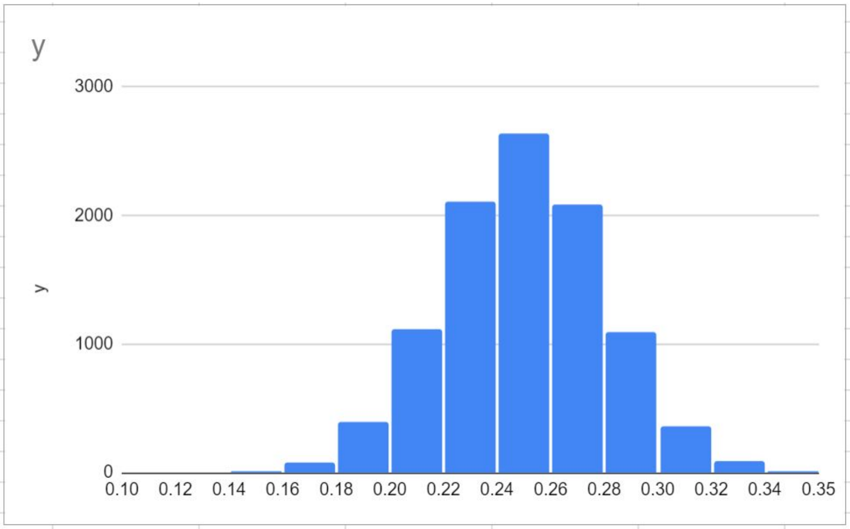
These are the ratios for the first timestamp that each precinct began to report its tabulations.

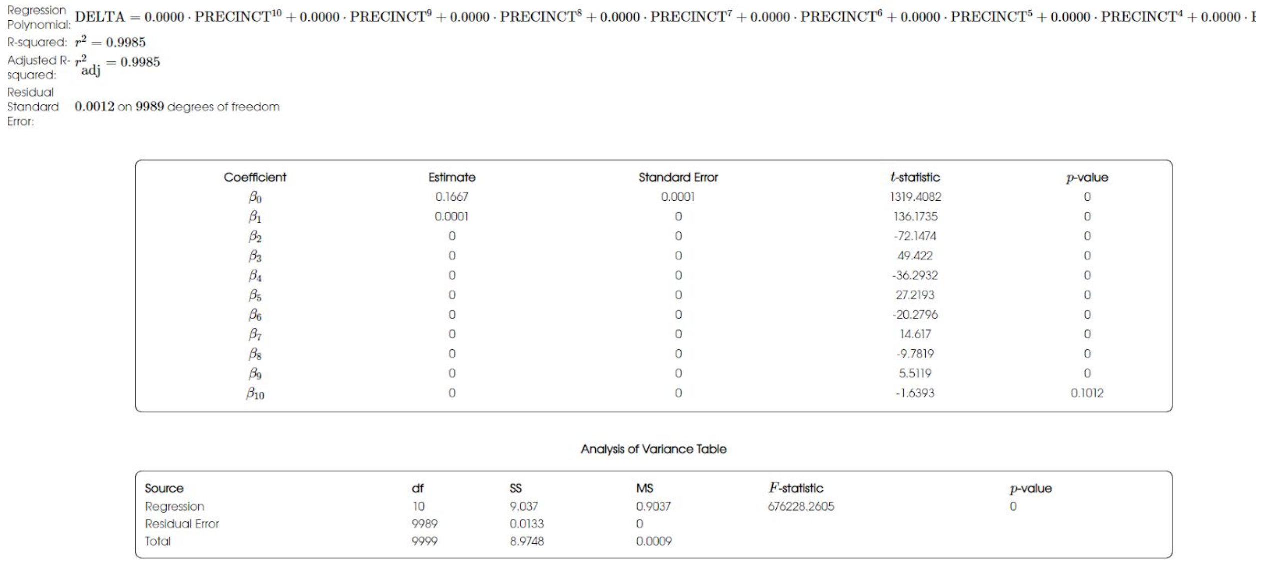
Now transform $\boldsymbol{\sigma}$ to 7%, applying the formula $(7r_k/100)$;
Now transform $\boldsymbol{\mu}$ to 40%, applying the formula $0.40+(10r_k/100)$.
Let this transformation be equal to $\mathbf{R}_2$.

These are the ratios for the second timestamp that each precinct updated its tabulations.

Now let **D** be the set of differences, $r_{2,k} - r_{1,k}$, $\forall k \leq \boldsymbol{\beta}$, and then convert **D** to absolute value.

Set **D** will be the response of the regression, and set **P** will be the explanatory axis of the regression.
Histogram Distribution of **D**.





Regression Polynomial: $\mathrm{DELTA} = 0.0000 \cdot \mathrm{PRECINCT}^{10} + 0.0000 \cdot \mathrm{PRECINCT}^{9} + 0.0000 \cdot \mathrm{PRECINCT}^{8} + 0.0000 \cdot \mathrm{PRECINCT}^{7} + 0.0000 \cdot \mathrm{PRECINCT}^{6} + 0.0000 \cdot \mathrm{PRECINCT}^{5} + 0.0000 \cdot \mathrm{PRECINCT}^{4} + 0.0000 \cdot \mathrm{I}$

R-squared: $r^2 = 0.9985$

Adjusted R-squared: $r^2_{\mathrm{adj}} = 0.9985$

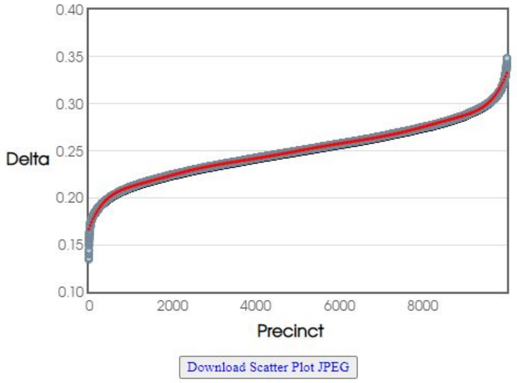
Residual Standard Error: **0.0012** on **9989** degrees of freedom

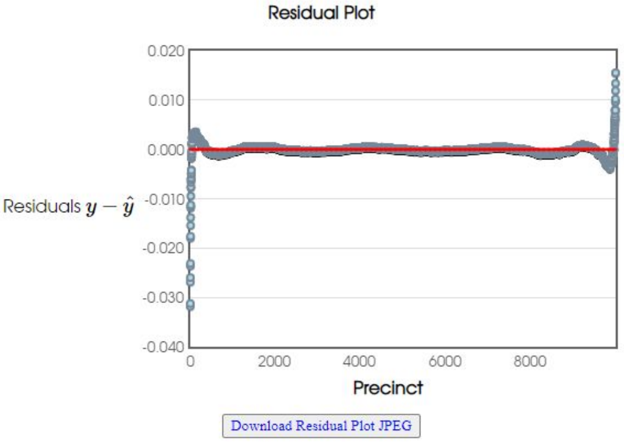
Coefficient	Estimate	Standard Error	t-statistic	p-value
β_0	0.1667	0.0001	1319.4082	0
β_1	0.0001	0	136.1735	0
β_2	0	0	-72.1474	0
β_3	0	0	49.422	0
β_4	0	0	-36.2932	0
β_5	0	0	27.2193	0
β_6	0	0	-20.2796	0
β_7	0	0	14.617	0
β_8	0	0	-9.7819	0
β_9	0	0	5.5119	0
β_{10}	0	0	-1.6393	0.1012

Analysis of Variance Table

Source	df	SS	MS	F-statistic	p-value
Regression	10	9.037	0.9037	676228.2605	0
Residual Error	9989	0.0133	0		
Total	9999	8.9748	0.0009		

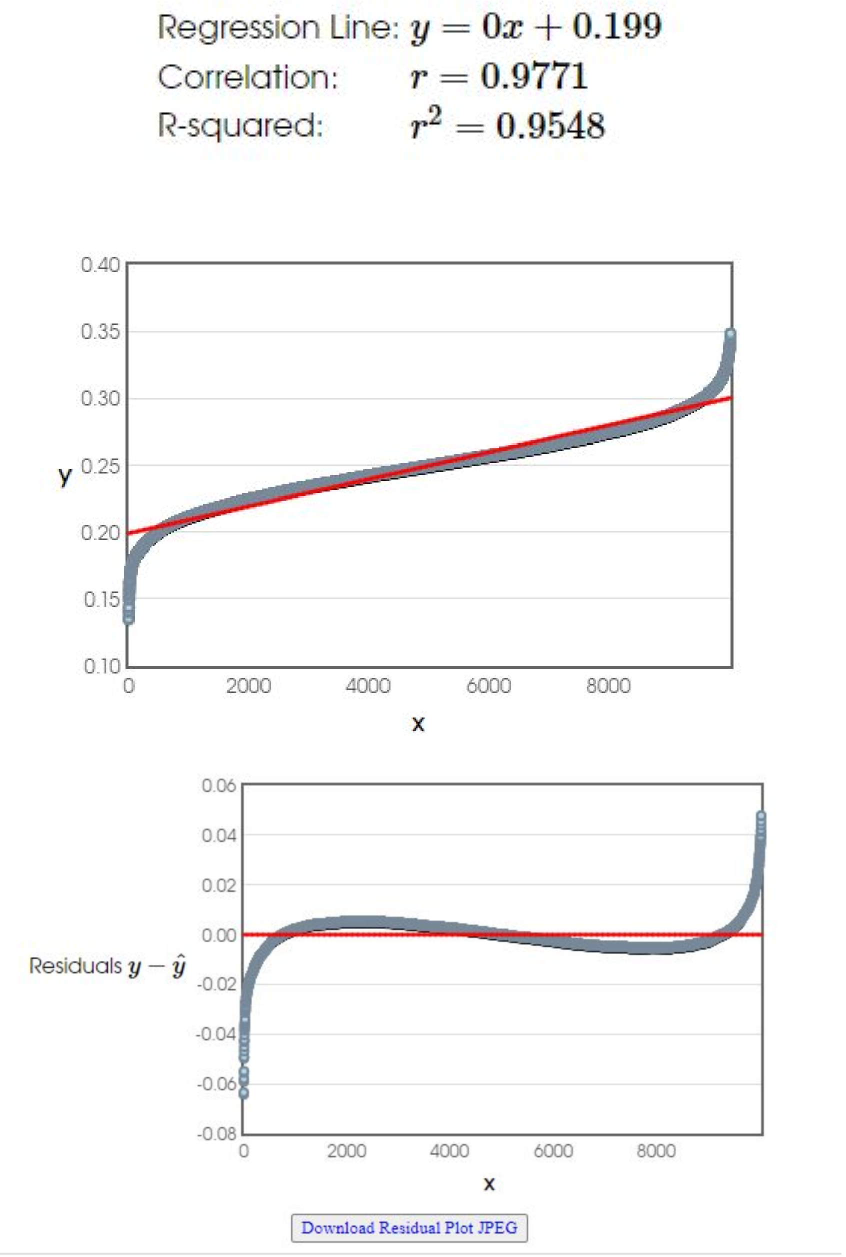
The resulting graph of the difference between these two naturally distributed sets appears as follows:





The resulting graph of the difference between these two naturally distributed sets appears as follows;

Using Linear Regression:



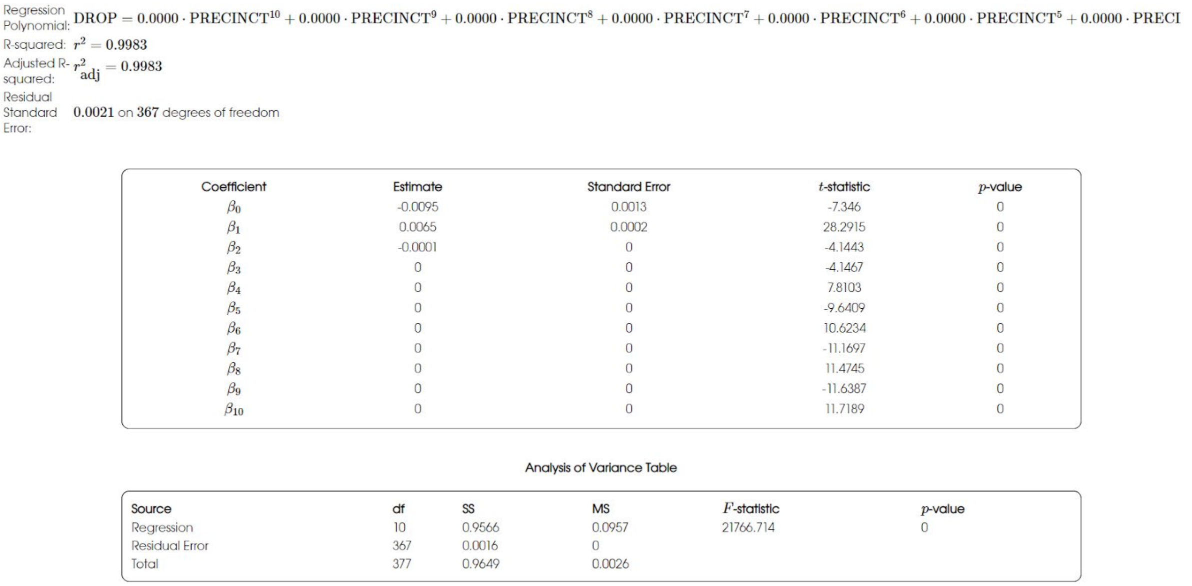
Regression Line: $y = 0x + 0.199$

Correlation: $r = 0.9771$

R-squared: $r^2 = 0.9548$

Oakland County Regression, 010Evo

After the sets $\mathbf{E}_{\mathbf{j-\varepsilon}}$ and $\mathbf{F}_{\mathbf{j-\varepsilon}}$ were determined for Oakland County, the result of the polynomial regression returned a 99.05% R^2 value for degree five (quintic regression) and a 99.83% R^2 for degree ten (Decic Regression). All successive degrees fail catastrophically. Trump suffers a net loss of 112,044 votes from this act. Actual polynomial on next page:



Regression Polynomial: $\mathrm{DROP} = 0.0000 \cdot \mathrm{PRECINCT}^{10} + 0.0000 \cdot \mathrm{PRECINCT}^{9} + 0.0000 \cdot \mathrm{PRECINCT}^{8} + 0.0000 \cdot \mathrm{PRECINCT}^{7} + 0.0000 \cdot \mathrm{PRECINCT}^{6} + 0.0000 \cdot \mathrm{PRECINCT}^{5} + 0.0000 \cdot \mathrm{PRECI}$

R-squared: $r^2 = 0.9983$

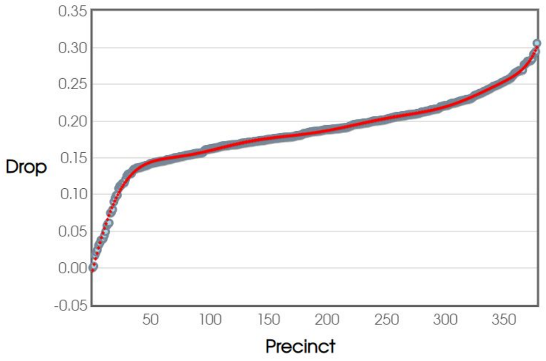
Adjusted R-squared: $r^2_{\mathrm{adj}} = 0.9983$

Residual Standard Error: **0.0021** on **367** degrees of freedom

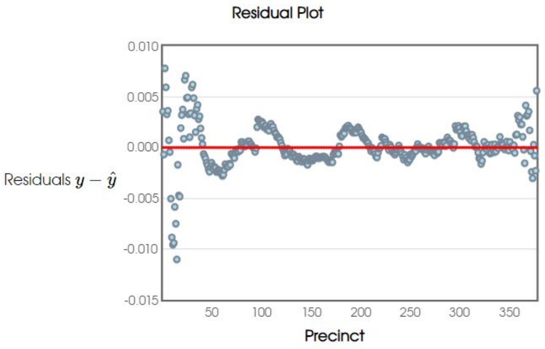
Coefficient	Estimate	Standard Error	t-statistic	p-value
β_0	-0.0095	0.0013	-7.346	0
β_1	0.0065	0.0002	28.2915	0
β_2	-0.0001	0	-4.1443	0
β_3	0	0	-4.1467	0
β_4	0	0	7.8103	0
β_5	0	0	-9.6409	0
β_6	0	0	10.6234	0
β_7	0	0	-11.1697	0
β_8	0	0	11.4745	0
β_9	0	0	-11.6387	0
β_{10}	0	0	11.7189	0

Analysis of Variance Table

Source	df	SS	MS	F-statistic	p-value
Regression	10	0.9566	0.0957	21766.714	0
Residual Error	367	0.0016	0		
Total	377	0.9649	0.0026		



Download Scatter Plot JPEG



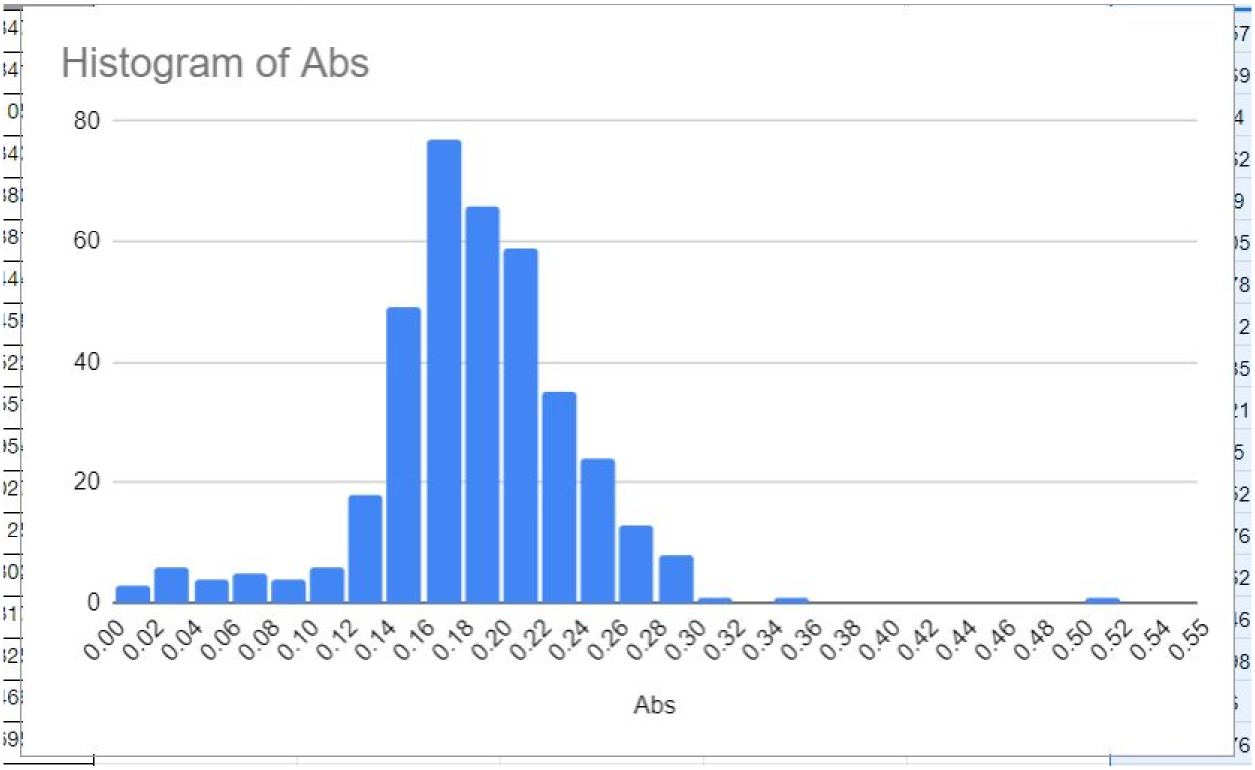
Download Residual Plot JPEG

Oakland County, Actual Decic Polynomial:
Mode: normal x,y analysis
Polynomial degree 10, 378 x,y data pairs.
Correlation coefficient = 0.9983586620981255
Standard error = 0.002052365687023779

Output form: mathematical function:

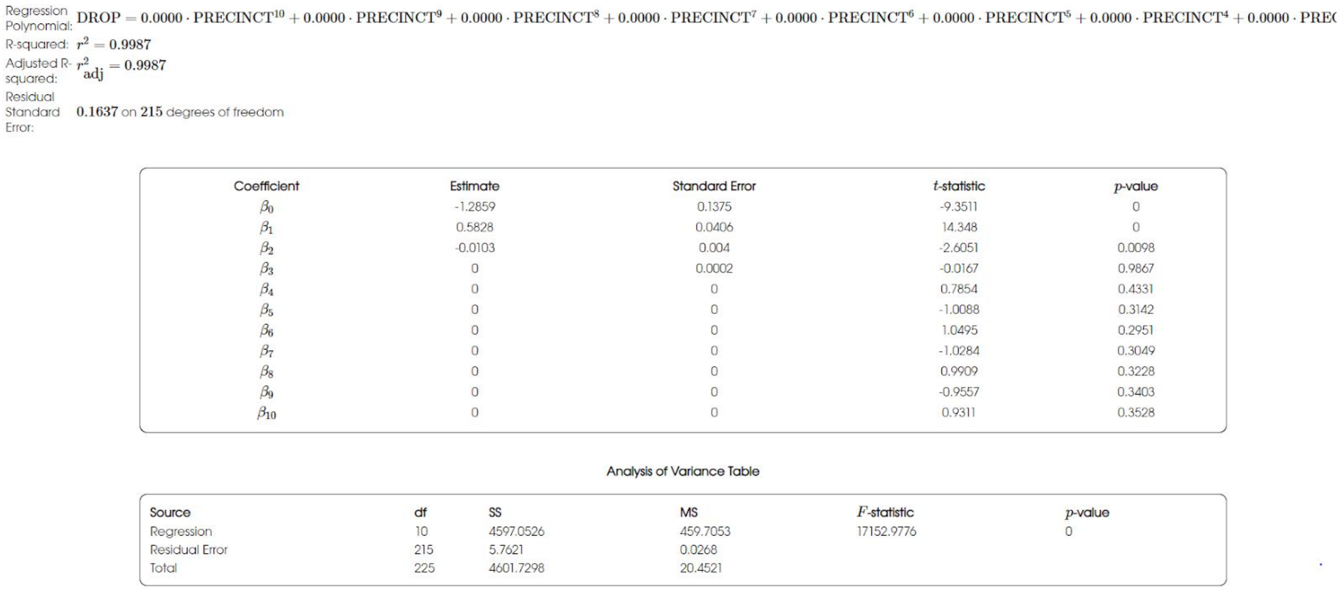
```
f(x) = -9.4560878626671480e-003 * x^0
    +  6.5312388128317758e-003 * x^1
    + -5.6432191533824423e-005 * x^2
    + -1.5510230293790736e-006 * x^3
    +  4.4410490812850710e-008 * x^4
    + -5.0090131092300033e-010 * x^5
    +  3.1570075939277991e-012 * x^6
    + -1.1966602890131484e-014 * x^7
    +  2.7118155733875802e-017 * x^8
    + -3.3892101121244086e-020 * x^9
    +  1.7985920499718028e-023 * x^10
```

Oakland Hist, same bucket size at 2%.



Kent County Regression, 010Evo

After the sets $\mathbf{E}_{\mathbf{j}-\varepsilon}$ and $\mathbf{F}_{\mathbf{j}-\varepsilon}$ were determined for Kent County, the result of the polynomial regression returned a 99.81% R^2 value for degree five (quintic regression) and a 99.87% R^2 for degree ten (Decic Regression). All successive degrees fail catastrophically. Trump then suffers a net loss of 44,053 votes from this act.



Regression Polynomial: $\mathrm{DROP} = 0.0000 \cdot \mathrm{PRECINCT}^{10} + 0.0000 \cdot \mathrm{PRECINCT}^{9} + 0.0000 \cdot \mathrm{PRECINCT}^{8} + 0.0000 \cdot \mathrm{PRECINCT}^{7} + 0.0000 \cdot \mathrm{PRECINCT}^{6} + 0.0000 \cdot \mathrm{PRECINCT}^{5} + 0.0000 \cdot \mathrm{PRECINCT}^{4} + 0.0000 \cdot \mathrm{PREC}$

R-squared: $r^2 = 0.9987$

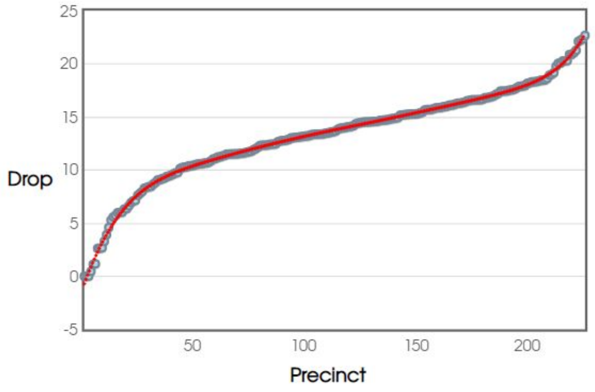
Adjusted R-squared: $r^2_{\mathrm{adj}} = 0.9987$

Residual Standard Error: 0.1637 on 215 degrees of freedom

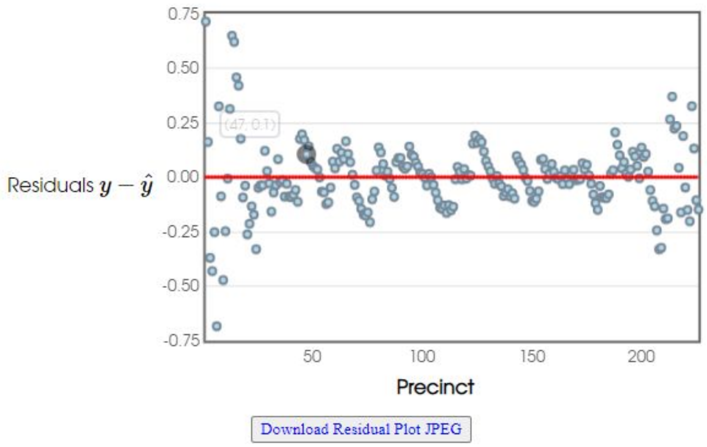
Coefficient	Estimate	Standard Error	t-statistic	p-value
β_0	-1.2859	0.1375	-9.3511	0
β_1	0.5828	0.0406	14.348	0
β_2	-0.0103	0.004	-2.6051	0.0098
β_3	0	0.0002	-0.0167	0.9867
β_4	0	0	0.7854	0.4331
β_5	0	0	-1.0088	0.3142
β_6	0	0	1.0495	0.2951
β_7	0	0	-1.0284	0.3049
β_8	0	0	0.9909	0.3228
β_9	0	0	-0.9557	0.3403
β_{10}	0	0	0.9311	0.3528

Analysis of Variance Table

Source	df	SS	MS	F-statistic	p-value
Regression	10	4597.0526	459.7053	17152.9776	0
Residual Error	215	5.7621	0.0268		
Total	225	4601.7298	20.4521		



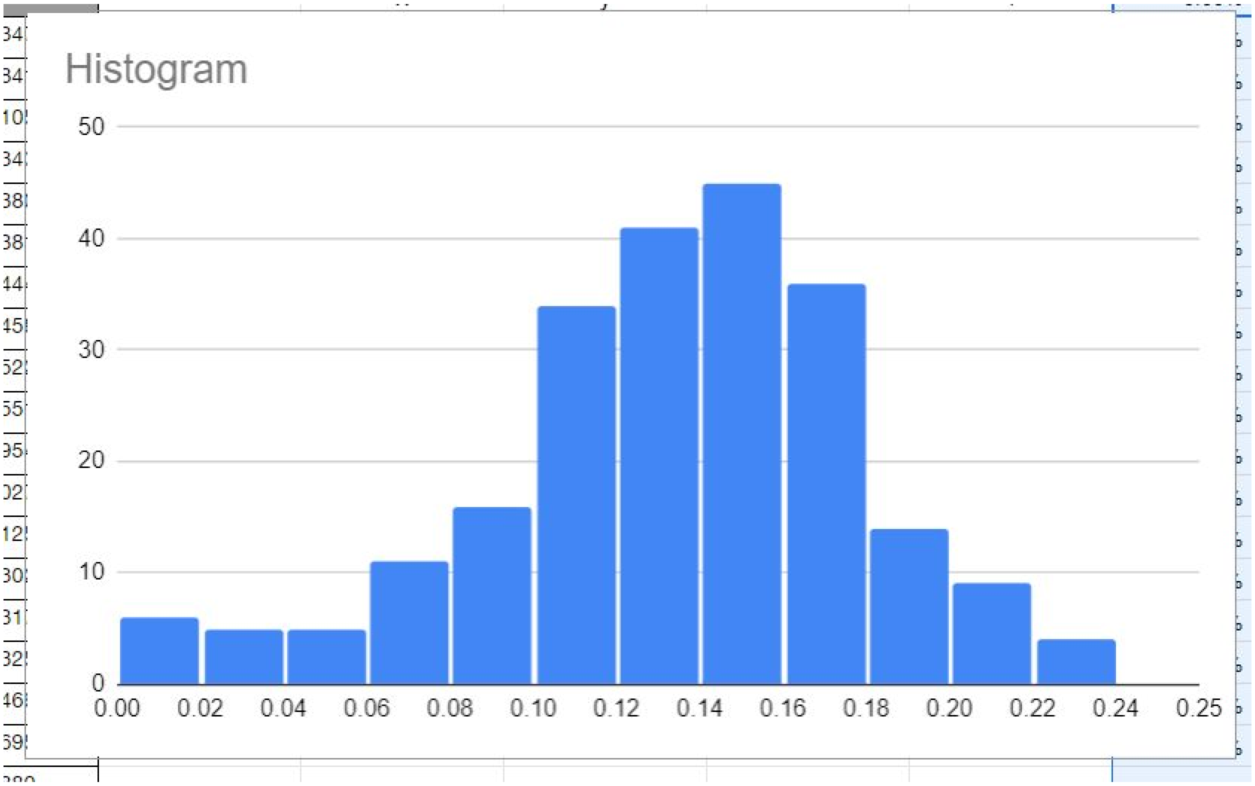
Residual Plot



Kent County, Actual Polynomial

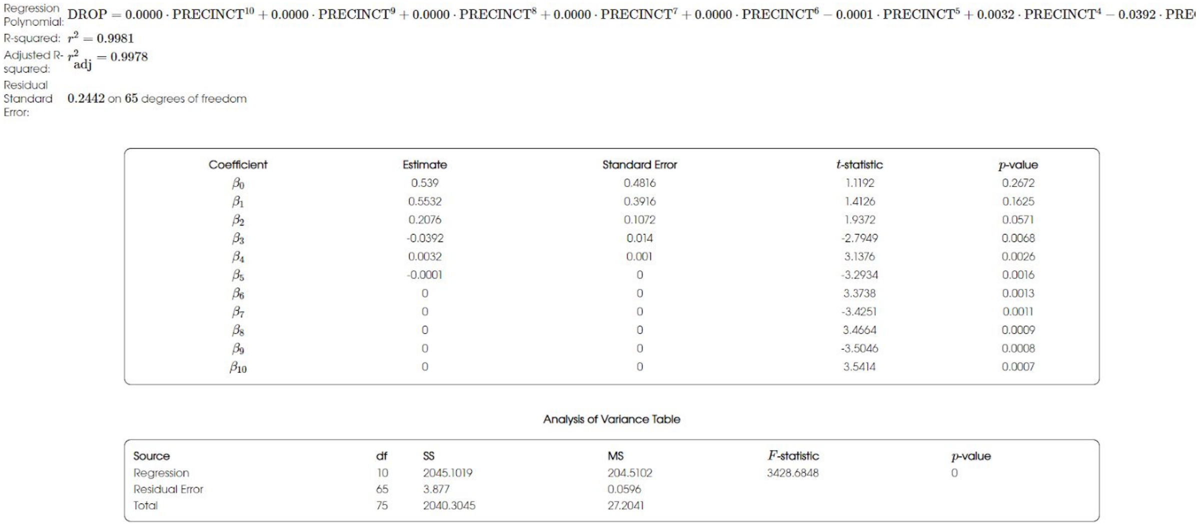
```
Mode: normal x,y analysis
Polynomial degree 10, 223 x,y data pairs.
Correlation coefficient = 0.9989403255368532
Standard error = 0.13965184055614888

Output form: mathematical function:

f(x) = -2.8769607098756977e+000 * x^0
  +  9.5081814995777381e-001 * x^1
  + -3.9802883319774607e-002 * x^2
  +  1.1602085303810540e-003 * x^3
  + -2.2588609052030097e-005 * x^4
  +  2.9192396240136943e-007 * x^5
  + -2.4881463880642809e-009 * x^6
  +  1.3750964209010889e-011 * x^7
  + -4.7236551004857299e-014 * x^8
  +  9.1403793373321801e-017 * x^9
  + -7.5946088963928345e-020 * x^10
```

Kent Hist, same bucket size at 2%.



Kalamazoo County Regression, 010Evo

After the sets $\mathbf{E}_{j-\varepsilon}$ and $\mathbf{F}_{j-\varepsilon}$ were determined for Kent County, the result of the polynomial regression returned a 99.51% R^2 value for degree five and a 99.81% R^2 for degree ten. All successive degrees fail catastrophically (except 11, which has a lower R^2 value than 10). Trump then suffers a net loss of 15,910 votes from this act.



Regression Polynomial: $\mathrm{DROP} = 0.0000 \cdot \mathrm{PRECINCT}^{10} + 0.0000 \cdot \mathrm{PRECINCT}^{9} + 0.0000 \cdot \mathrm{PRECINCT}^{8} + 0.0000 \cdot \mathrm{PRECINCT}^{7} + 0.0000 \cdot \mathrm{PRECINCT}^{6} - 0.0001 \cdot \mathrm{PRECINCT}^{5} + 0.0032 \cdot \mathrm{PRECINCT}^{4} - 0.0392 \cdot \mathrm{PRE}$

R-squared: $r^2 = 0.9981$

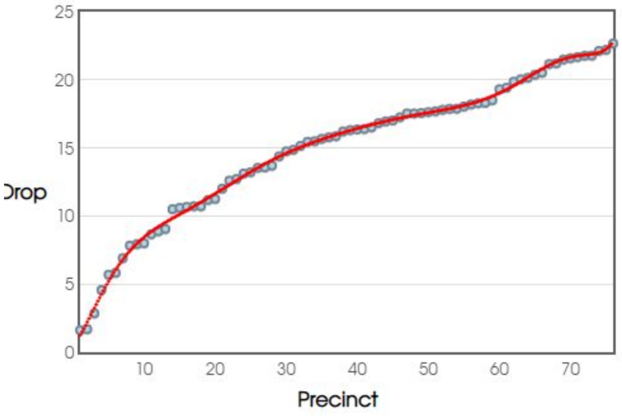
Adjusted R-squared: $r^2_{\mathrm{adj}} = 0.9978$

Residual Standard Error: 0.2442 on 65 degrees of freedom

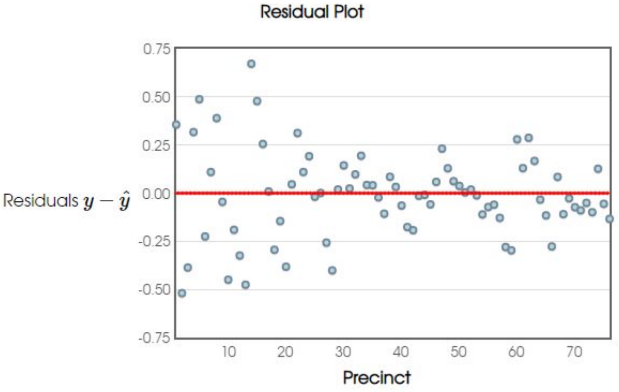
Coefficient	Estimate	Standard Error	t-statistic	p-value
β_0	0.539	0.4816	1.1192	0.2672
β_1	0.5532	0.3916	1.4126	0.1625
β_2	0.2076	0.1072	1.9372	0.0571
β_3	-0.0392	0.014	-2.7949	0.0068
β_4	0.0032	0.001	3.1376	0.0026
β_5	-0.0001	0	-3.2934	0.0016
β_6	0	0	3.3738	0.0013
β_7	0	0	-3.4251	0.0011
β_8	0	0	3.4664	0.0009
β_9	0	0	-3.5046	0.0008
β_{10}	0	0	3.5414	0.0007

Analysis of Variance Table

Source	df	SS	MS	F-statistic	p-value
Regression	10	2045.1019	204.5102	3428.6848	0
Residual Error	65	3.877	0.0596		
Total	75	2040.3045	27.2041		



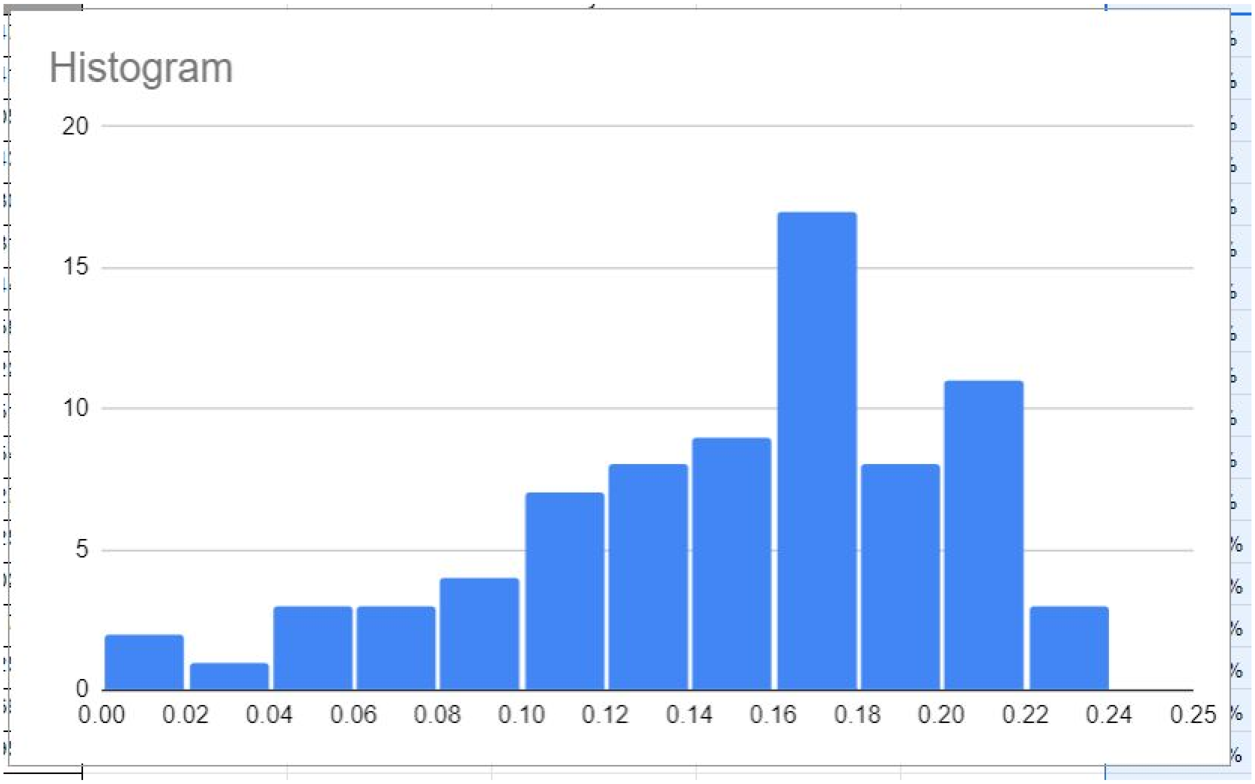
Download Scatter Plot JPEG



Download Residual Plot JPEG

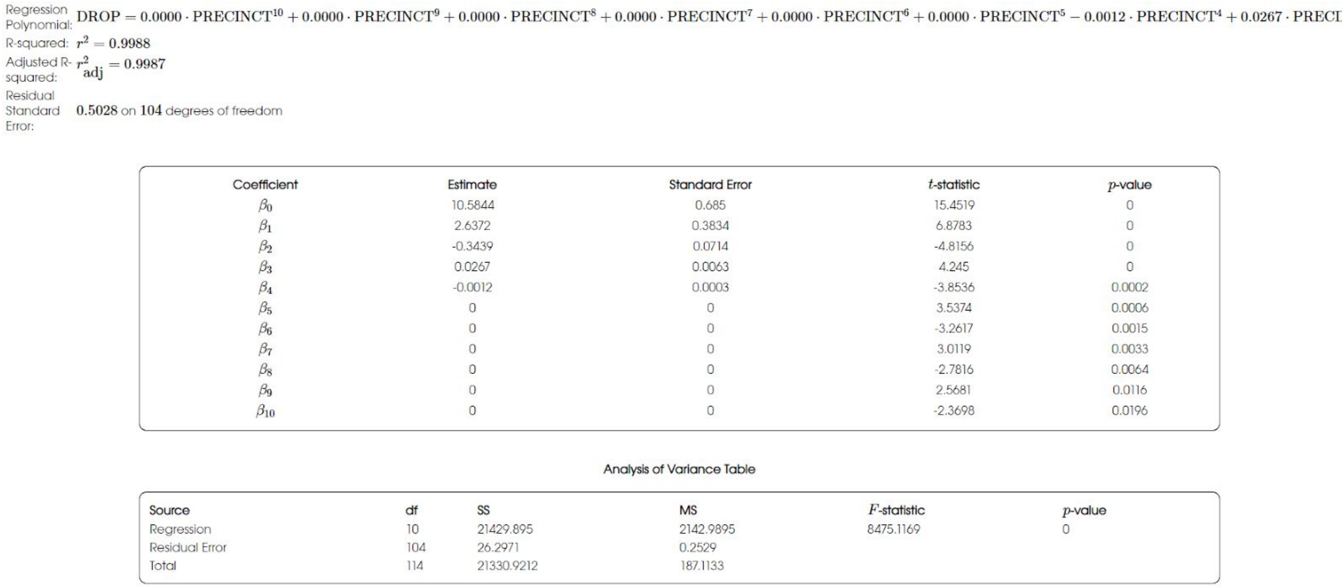
```
Kalamazoo, Actual Polynomial
Mode: normal x,y analysis
Polynomial degree 10, 76 x,y data pairs.
Correlation coefficient = 0.9981131581434618
Standard error = 0.22808639511639908

Output form: mathematical function:

f(x) =  5.4197236758015255e-001 * x^0
    +  5.5013562909189462e-001 * x^1
    +  2.0851297904859911e-001 * x^2
    + -3.9306532679476022e-002 * x^3
    +  3.2285582520283494e-003 * x^4
    + -1.5029807718279015e-004 * x^5
    +  4.2965793755929391e-006 * x^6
    + -7.6982371277358507e-008 * x^7
    +  8.4338504776218692e-010 * x^8
    + -5.1646565785570038e-012 * x^9
    +  1.3538808051149861e-014 * x^10
```

Kalam Hist, same bucket size at 2%.



Ingham County Regression, 010Evo (No Delta Data, Reflexive)

Since Ingham County had no prior differences to measure for the regression, we simply took the least percentage recorded by each precinct, the result of the polynomial regression returned a 99.84% R^2 value for degree five and a 99.88% R^2 for degree ten. All successive degrees fail catastrophically (except 11, which has a lower R^2 value than 10). We do not have the prior data from Ingham County to know the amount of votes stolen from President Donald J. Trump. We will use this data in Section II of this Chapter.



Regression Polynomial: $\mathrm{DROP} = 0.0000 \cdot \mathrm{PRECINCT}^{10} + 0.0000 \cdot \mathrm{PRECINCT}^{9} + 0.0000 \cdot \mathrm{PRECINCT}^{8} + 0.0000 \cdot \mathrm{PRECINCT}^{7} + 0.0000 \cdot \mathrm{PRECINCT}^{6} + 0.0000 \cdot \mathrm{PRECINCT}^{5} - 0.0012 \cdot \mathrm{PRECINCT}^{4} + 0.0267 \cdot \mathrm{PRECI}$

R-squared: $r^2 = 0.9988$

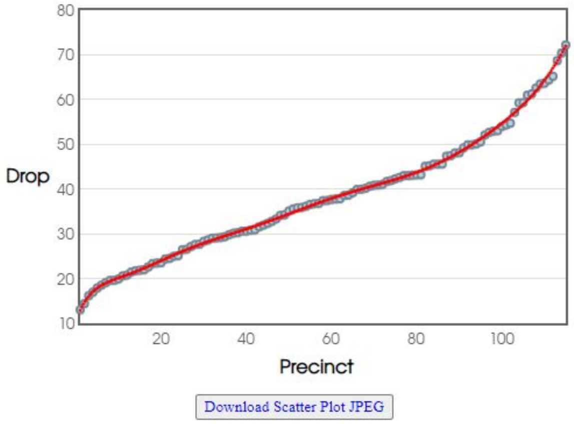
Adjusted R-squared: $r^2_{\text{adj}} = 0.9987$

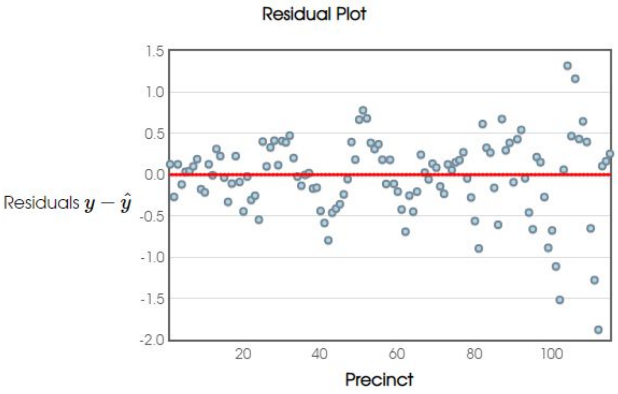
Residual Standard Error: 0.5028 on 104 degrees of freedom

Coefficient	Estimate	Standard Error	t-statistic	p-value
β_0	10.5844	0.685	15.4519	0
β_1	2.6372	0.3834	6.8783	0
β_2	-0.3439	0.0714	-4.8156	0
β_3	0.0267	0.0063	4.245	0
β_4	-0.0012	0.0003	-3.8536	0.0002
β_5	0	0	3.5374	0.0006
β_6	0	0	-3.2617	0.0015
β_7	0	0	3.0119	0.0033
β_8	0	0	-2.7816	0.0064
β_9	0	0	2.5681	0.0116
β_{10}	0	0	-2.3698	0.0196

Analysis of Variance Table

Source	df	SS	MS	F-statistic	p-value
Regression	10	21429.895	2142.9895	8475.1169	0
Residual Error	104	26.2971	0.2529		
Total	114	21330.9212	187.1133		



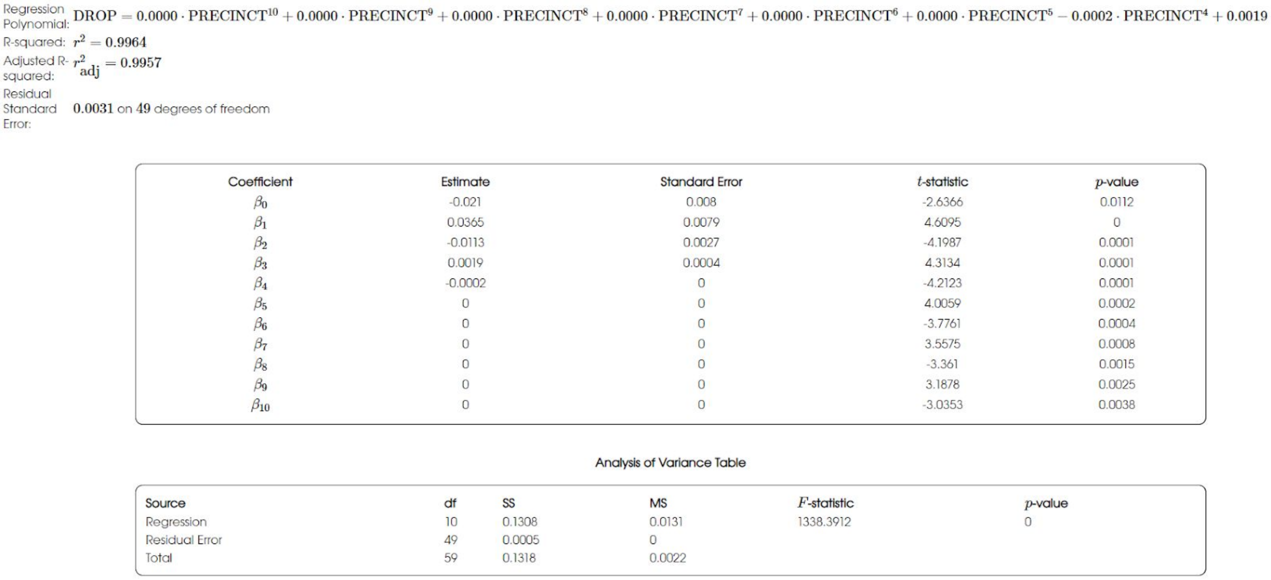


THIS IS NOT DELTA REGRESSION, WE WILL RETURN TO THIS IN SECTION II.

```
Ingham, Actual Polynomial
Mode: normal x,y analysis
Polynomial degree 10, 115 x,y data pairs.
Correlation coefficient = 0.9987829002399025
Standard error = 0.47932344591084103

Output form: mathematical function:

f(x) =  1.0587210078193035e+001 * x^0
    +  2.6351121234279176e+000 * x^1
    + -3.4343935874537213e-001 * x^2
    +  2.6686733515223368e-002 * x^3
    + -1.1879535482091071e-003 * x^4
    +  3.2296673007515566e-005 * x^5
    + -5.5369140829377692e-007 * x^6
    +  6.0031965336045307e-009 * x^7
    + -3.9885136059869065e-011 * x^8
    +  1.4810994529954557e-013 * x^9
    + -2.3530020548186278e-016 * x^10
```

Saginaw County Regression, Part 1 010Evo

Since many precincts in Saginaw County had no prior differences to measure for the regression, we first applied the same logic as we did to the counties prior to Ingham, only considering the precincts with recorded negative changes. The regression returned a 99.01% R^2 value for degree five and a 99.64% R^2 for degree ten. All successive degrees fail catastrophically (except 11, which has a lower R^2 value than 10) .Trump then suffers a net of 10,286 votes from this act.



Regression Polynomial: $\mathrm{DROP} = 0.0000 \cdot \mathrm{PRECINCT}^{10} + 0.0000 \cdot \mathrm{PRECINCT}^{9} + 0.0000 \cdot \mathrm{PRECINCT}^{8} + 0.0000 \cdot \mathrm{PRECINCT}^{7} + 0.0000 \cdot \mathrm{PRECINCT}^{6} + 0.0000 \cdot \mathrm{PRECINCT}^{5} - 0.0002 \cdot \mathrm{PRECINCT}^{4} + 0.0019$

R-squared: $r^2 = 0.9964$

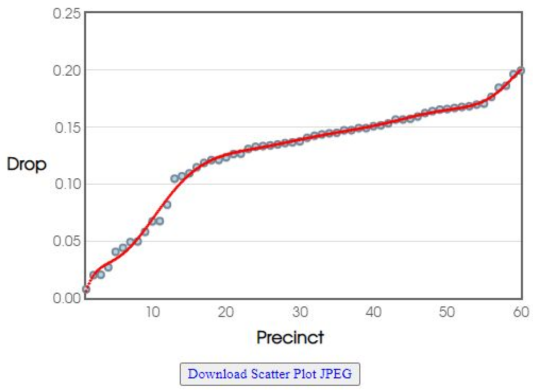
Adjusted R-squared: $r^2_{\mathrm{adj}} = 0.9957$

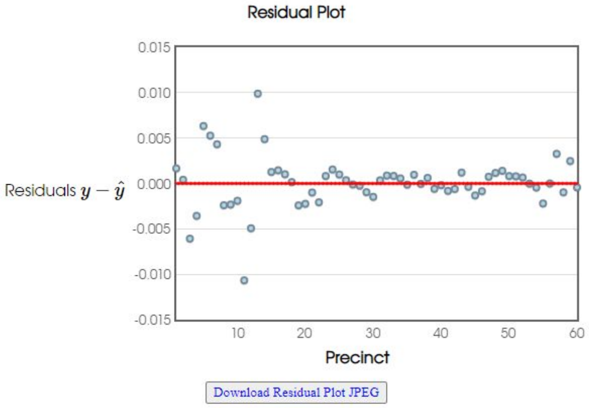
Residual Standard Error: **0.0031** on **49** degrees of freedom

Coefficient	Estimate	Standard Error	t-statistic	p-value
β_0	-0.021	0.008	-2.6366	0.0112
β_1	0.0365	0.0079	4.6095	0
β_2	-0.0113	0.0027	-4.1987	0.0001
β_3	0.0019	0.0004	4.3134	0.0001
β_4	-0.0002	0	-4.2123	0.0001
β_5	0	0	4.0059	0.0002
β_6	0	0	-3.7761	0.0004
β_7	0	0	3.5575	0.0008
β_8	0	0	-3.361	0.0015
β_9	0	0	3.1878	0.0025
β_{10}	0	0	-3.0353	0.0038

Analysis of Variance Table

Source	df	SS	MS	F-statistic	p-value
Regression	10	0.1308	0.0131	1338.3912	0
Residual Error	49	0.0005	0		
Total	59	0.1318	0.0022		

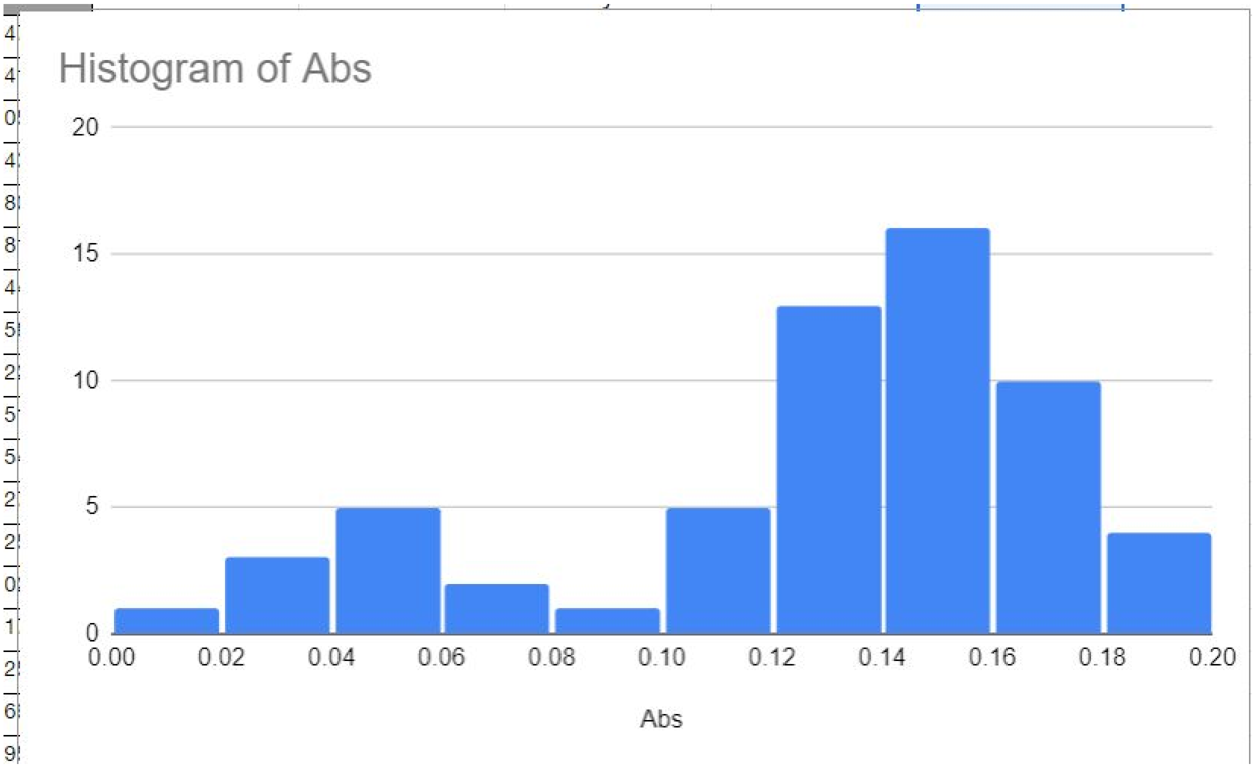




Saginaw 1, Actual Polynomial

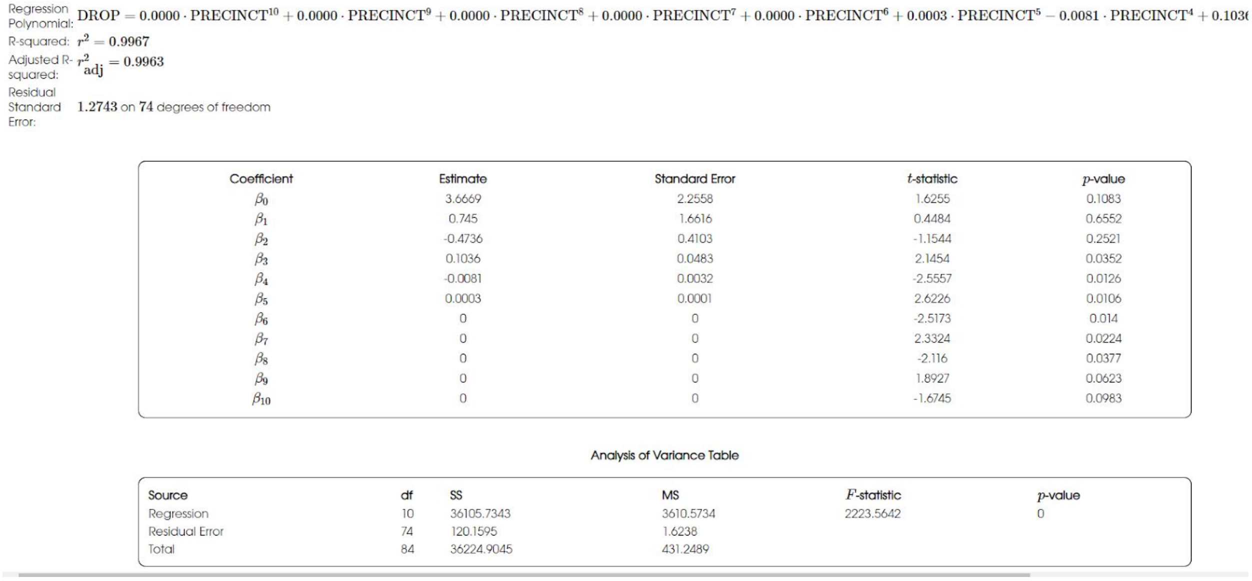
```
Mode: normal x,y analysis
Polynomial degree 10, 60 x,y data pairs.
Correlation coefficient = 0.9963822077812268
Standard error = 0.0028675907999551496

Output form: mathematical function:

f(x) = -2.0979080217749423e-002 * x^0
   +  3.6548363825861283e-002 * x^1
   + -1.1273408652485651e-002 * x^2
   +  1.8891853630980885e-003 * x^3
   + -1.6912091064194697e-004 * x^4
   +  8.9599930581750883e-006 * x^5
   + -2.9658213881205975e-007 * x^6
   +  6.2130474729839530e-009 * x^7
   + -8.0128996181236346e-011 * x^8
   +  5.8085080465768596e-013 * x^9
   + -1.8112577500212840e-015 * x^10
```

Saginaw Partial, same bucket size at 2%.



Saginaw County Regression, Part 2 010Evo

Since Saginaw County had many precincts with prior differences to measure for the regression,we then applied the same method as we did in Ingham. The result of the polynomial regression returned a 99.84% R^2 value for degree five and a 99.67% R^2 for degree ten. All successive degrees fail catastrophically (except 11, which has a slightly increased R^2 value than 10, at 99.67%). We do not have the data to determine the total amount of votes stolen in Saginaw, the current minimum stands at 10,286.



Regression Polynomial: $\mathrm{DROP} = 0.0000 \cdot \mathrm{PRECINCT}^{10} + 0.0000 \cdot \mathrm{PRECINCT}^{9} + 0.0000 \cdot \mathrm{PRECINCT}^{8} + 0.0000 \cdot \mathrm{PRECINCT}^{7} + 0.0000 \cdot \mathrm{PRECINCT}^{6} + 0.0003 \cdot \mathrm{PRECINCT}^{5} - 0.0081 \cdot \mathrm{PRECINCT}^{4} + 0.103$

R-squared: $r^2 = 0.9967$

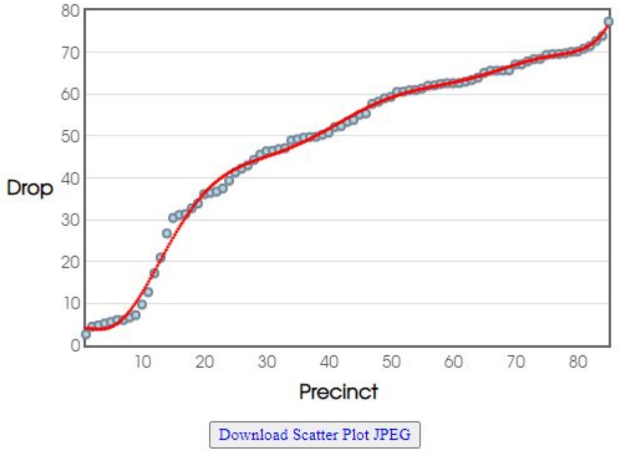
Adjusted R-squared: $r^2_{\mathrm{adj}} = 0.9963$

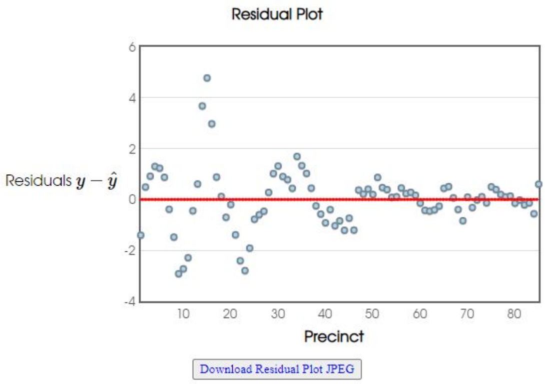
Residual Standard Error: **1.2743** on **74** degrees of freedom

Coefficient	Estimate	Standard Error	t-statistic	p-value
β_0	3.6669	2.2558	1.6255	0.1083
β_1	0.745	1.6616	0.4484	0.6552
β_2	-0.4736	0.4103	-1.1544	0.2521
β_3	0.1036	0.0483	2.1454	0.0352
β_4	-0.0081	0.0032	-2.5557	0.0126
β_5	0.0003	0.0001	2.6226	0.0106
β_6	0	0	-2.5173	0.014
β_7	0	0	2.3324	0.0224
β_8	0	0	-2.116	0.0377
β_9	0	0	1.8927	0.0623
β_{10}	0	0	-1.6745	0.0983

Analysis of Variance Table

Source	df	SS	MS	F-statistic	p-value
Regression	10	36105.7343	3610.5734	2223.5642	0
Residual Error	74	120.1595	1.6238		
Total	84	36224.9045	431.2489		



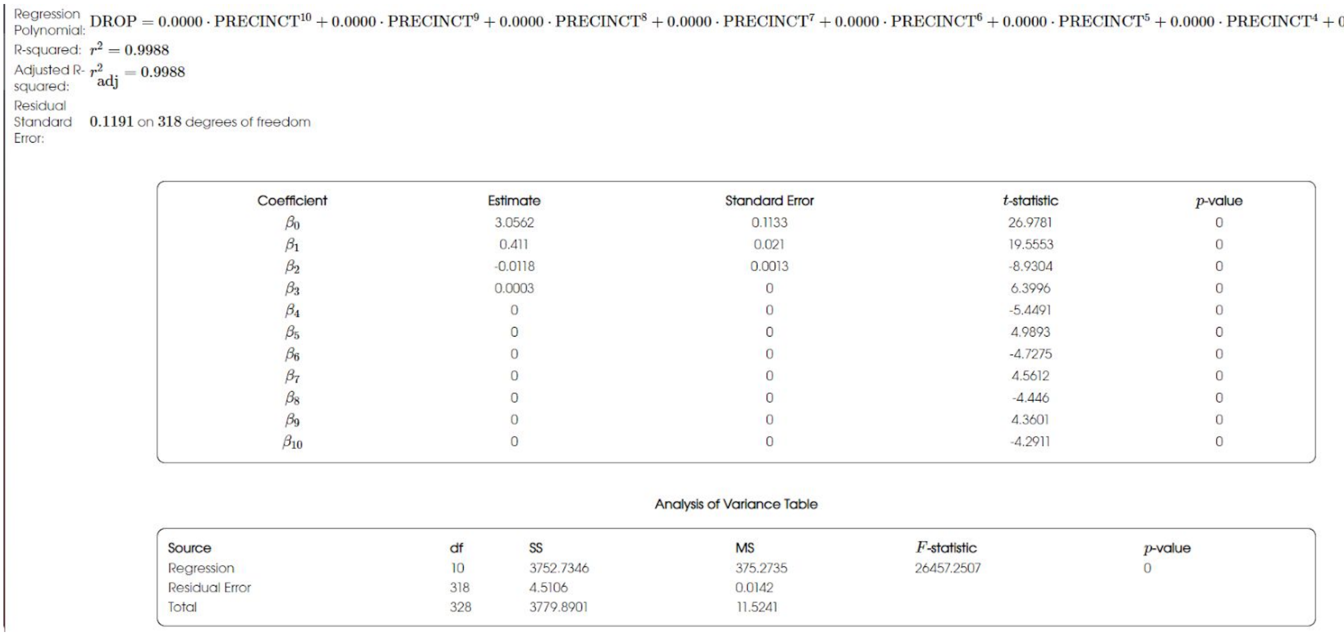


THIS IS NOT DELTA REGRESSION, WE WILL RETURN TO THIS IN SECTION II.

```
Saginaw 2, Actual Polynomial
Mode: normal x,y analysis
Polynomial degree 10, 85 x,y data pairs.
Correlation coefficient = 0.9966829978813773
Standard error = 1.2031989744703895

Output form: mathematical function:

f(x) =  3.6587382390260843e+000 * x^0
    +  7.5292894393938159e-001 * x^1
    + -4.7585174302399635e-001 * x^2
    +  1.0388361177825606e-001 * x^3
    + -8.1310151074834008e-003 * x^4
    +  3.3200524251345302e-004 * x^5
    + -7.9733633077362391e-006 * x^6
    +  1.1685947491276487e-007 * x^7
    + -1.0283384631706999e-009 * x^8
    +  4.9910188831513645e-012 * x^9
    + -1.0260745718821363e-014 * x^10
```

Macomb County Regression, 010Evo

After the sets $\mathbf{E}_{\mathbf{j}\text{ - }\varepsilon}$ and $\mathbf{F}_{\mathbf{j}\text{ - }\varepsilon}$ were determined for Macomb County, the result of the polynomial regression returned a 99.82% R^2 value for degree five (quintic regression) and a 99.88% R^2 for degree ten (Decic Regression). All successive degrees fail catastrophically (except degree 11, which has a lower R^2 value). Trump suffers a net loss of 70,944 votes from this act.



Regression Polynomial: $\mathrm{DROP} = 0.0000 \cdot \mathrm{PRECINCT}^{10} + 0.0000 \cdot \mathrm{PRECINCT}^{9} + 0.0000 \cdot \mathrm{PRECINCT}^{8} + 0.0000 \cdot \mathrm{PRECINCT}^{7} + 0.0000 \cdot \mathrm{PRECINCT}^{6} + 0.0000 \cdot \mathrm{PRECINCT}^{5} + 0.0000 \cdot \mathrm{PRECINCT}^{4} + 0$

R-squared: $r^2 = 0.9988$

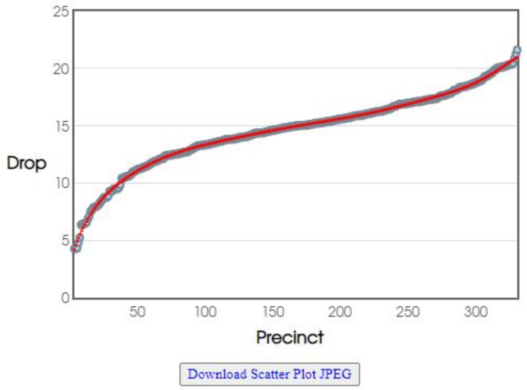
Adjusted R-squared: $r^2_{\mathrm{adj}} = 0.9988$

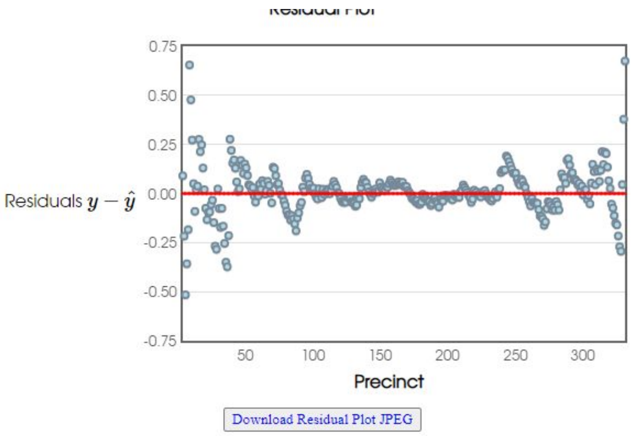
Residual Standard Error: 0.1191 on 318 degrees of freedom

Coefficient	Estimate	Standard Error	t-statistic	p-value
β_0	3.0562	0.1133	26.9781	0
β_1	0.411	0.021	19.5553	0
β_2	-0.0118	0.0013	-8.9304	0
β_3	0.0003	0	6.3996	0
β_4	0	0	-5.4491	0
β_5	0	0	4.9893	0
β_6	0	0	-4.7275	0
β_7	0	0	4.5612	0
β_8	0	0	-4.446	0
β_9	0	0	4.3601	0
β_{10}	0	0	-4.2911	0

Analysis of Variance Table

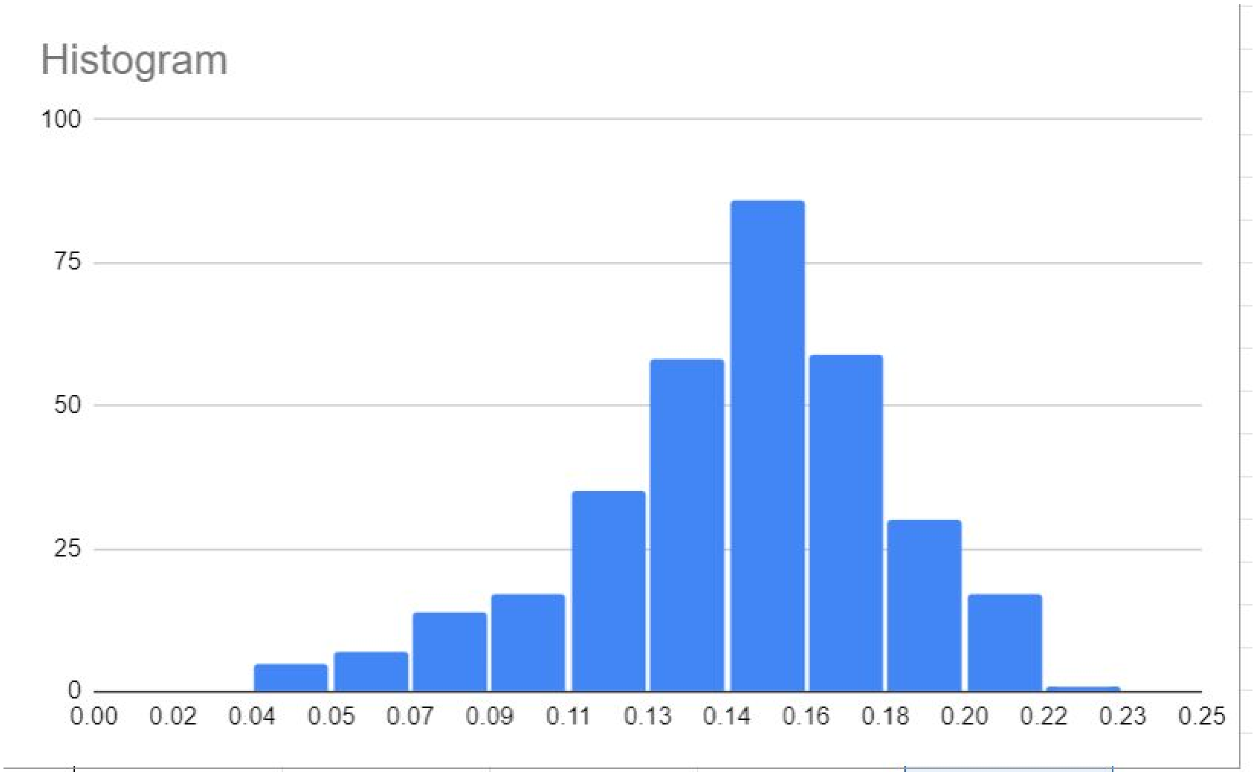
Source	df	SS	MS	F-statistic	p-value
Regression	10	3752.7346	375.2735	26457.2507	0
Residual Error	318	4.5106	0.0142		
Total	328	3779.8901	11.5241		





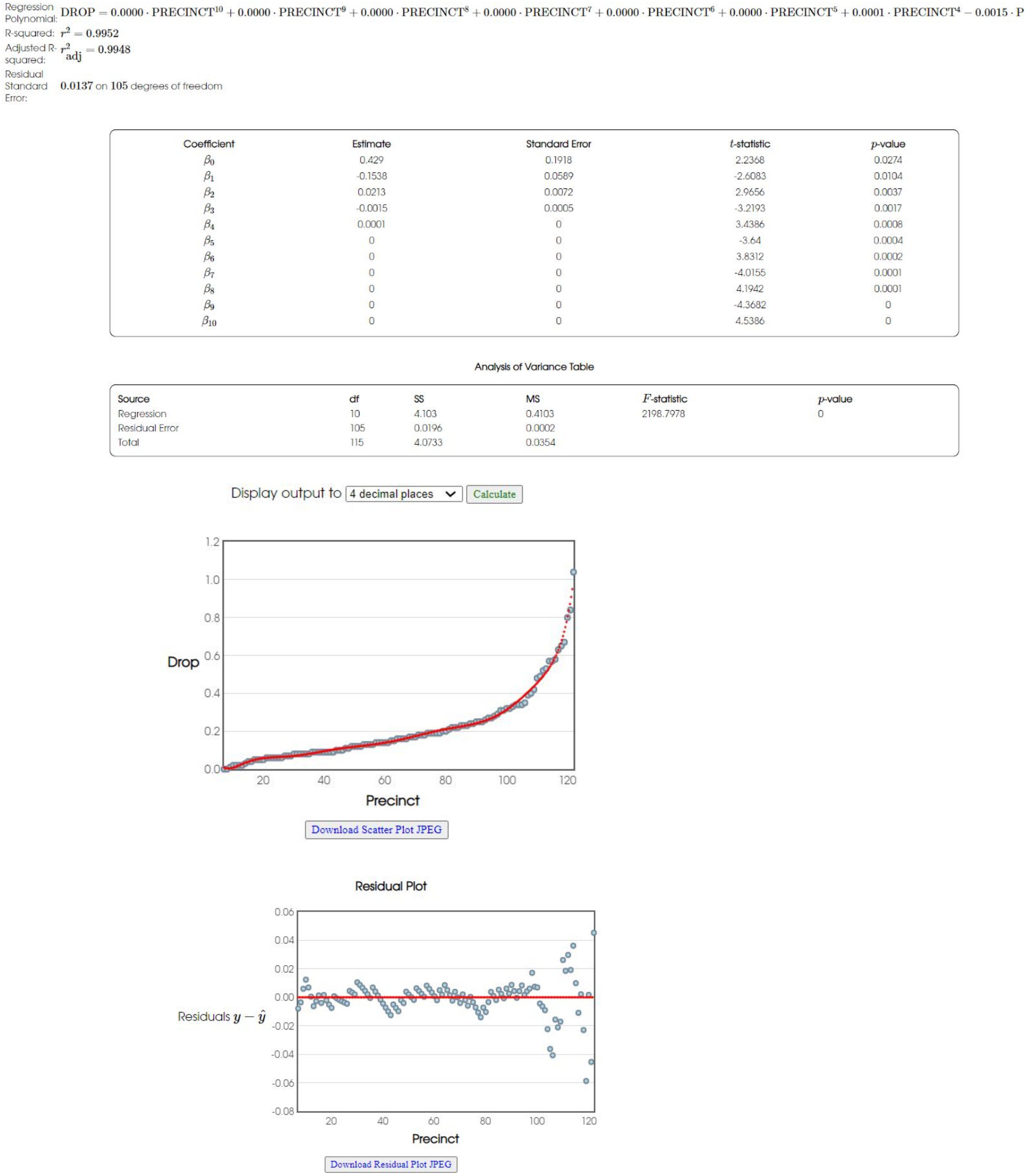
```
Macomb,  Actual Polynomial
Mode: normal x,y analysis
Polynomial degree 10, 329 x,y data pairs.
Correlation coefficient = 0.9988304639559799
Standard error = 0.11627129677001413

Output form: mathematical function:

f(x) =  3.0561895992198469e+000 * x^0
    +  4.1100698612116471e-001 * x^1
    + -1.1805899419019659e-002 * x^2
    +  2.5493886392480172e-004 * x^3
    + -3.6617291579924573e-006 * x^4
    +  3.4281574570430148e-008 * x^5
    + -2.0898417091744871e-010 * x^6
    +  8.2042534443506227e-013 * x^7
    + -1.9958170920886724e-015 * x^8
    +  2.7330308081619253e-018 * x^9
    + -1.6086412956942258e-021 * x^10
```

Macomb Hist, Bucket Size at 0.18 for best fit.



Detroit, AVCB Boards. Regression, 010Evo

After the sets $\mathbf{E}_{\mathbf{j}-\varepsilon}$ and $\mathbf{F}_{\mathbf{j}-\varepsilon}$ were determined for the AVCB BCounty, the result of the polynomial regression returned a 99.82% R^2 value for degree five (quintic regression) and a 99.52% R^2 for degree ten (Decic Regression). All successive degrees fail catastrophically, degree 11 also fails catastrophically. Trump suffers a net loss of 225 votes from this act; however, it is believed that this was used as the starting point, as in Ingham County, before further corrupting the vote tabulations via the Wheel Model, to which the remainder of this Analysis shall be dedicated.



Regression Polynomial: $\mathrm{DROP} = 0.0000 \cdot \mathrm{PRECINCT}^{10} + 0.0000 \cdot \mathrm{PRECINCT}^{9} + 0.0000 \cdot \mathrm{PRECINCT}^{8} + 0.0000 \cdot \mathrm{PRECINCT}^{7} + 0.0000 \cdot \mathrm{PRECINCT}^{6} + 0.0000 \cdot \mathrm{PRECINCT}^{5} + 0.0001 \cdot \mathrm{PRECINCT}^{4} - 0.0015 \cdot \mathrm{P}$

R-squared: $r^2 = 0.9952$

Adjusted R-squared: $r^2_{\mathrm{adj}} = 0.9948$

Residual Standard Error: 0.0137 on 105 degrees of freedom

Coefficient	Estimate	Standard Error	t-statistic	p-value
β_0	0.429	0.1918	2.2368	0.0274
β_1	-0.1538	0.0589	-2.6083	0.0104
β_2	0.0213	0.0072	2.9656	0.0037
β_3	-0.0015	0.0005	-3.2193	0.0017
β_4	0.0001	0	3.4386	0.0008
β_5	0	0	-3.64	0.0004
β_6	0	0	3.8312	0.0002
β_7	0	0	-4.0155	0.0001
β_8	0	0	4.1942	0.0001
β_9	0	0	-4.3682	0
β_{10}	0	0	4.5386	0

Analysis of Variance Table

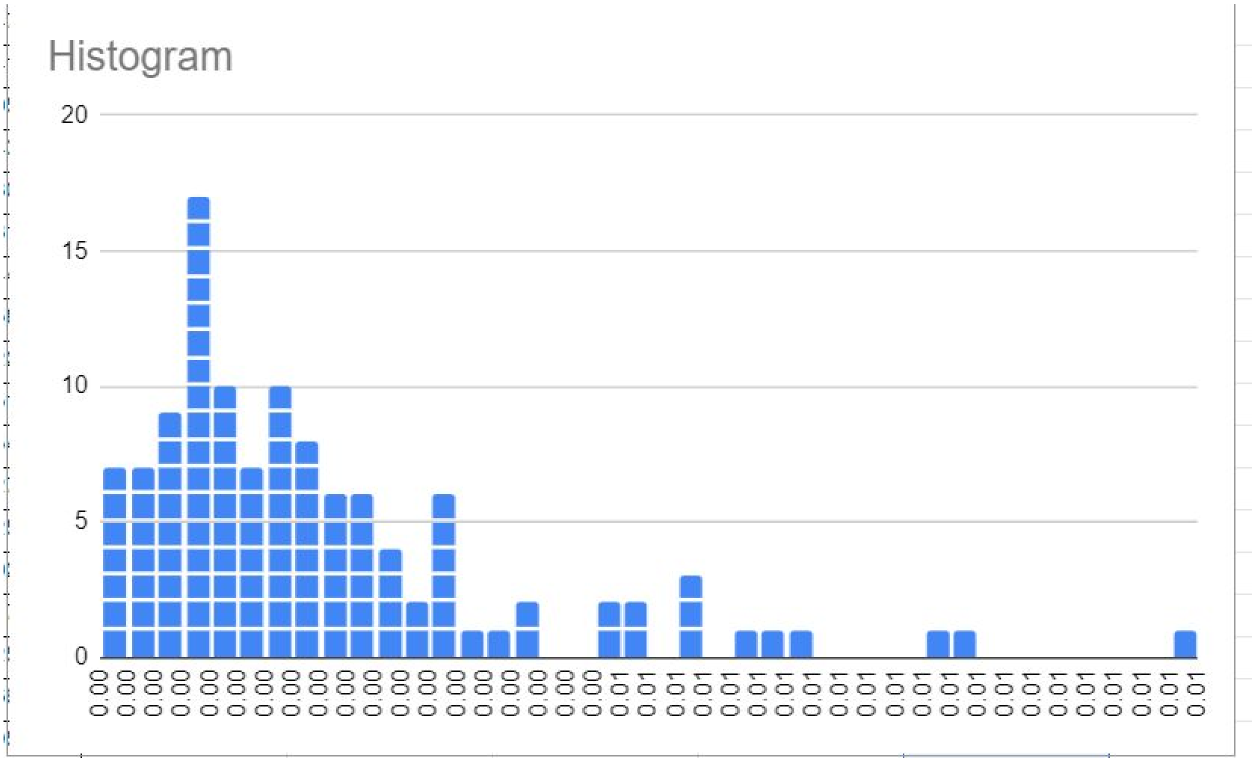
Source	df	SS	MS	F-statistic	p-value
Regression	10	4.103	0.4103	2198.7978	0
Residual Error	105	0.0196	0.0002		
Total	115	4.0733	0.0354		

Display output to 4 decimal places Calculate

AVCB Detroit, Actual Polynomial
Mode: normal x,y analysis
Polynomial degree 10, 116 x,y data pairs.
Correlation coefficient = 0.9952398459819962
Standard error = 0.013041544376612434

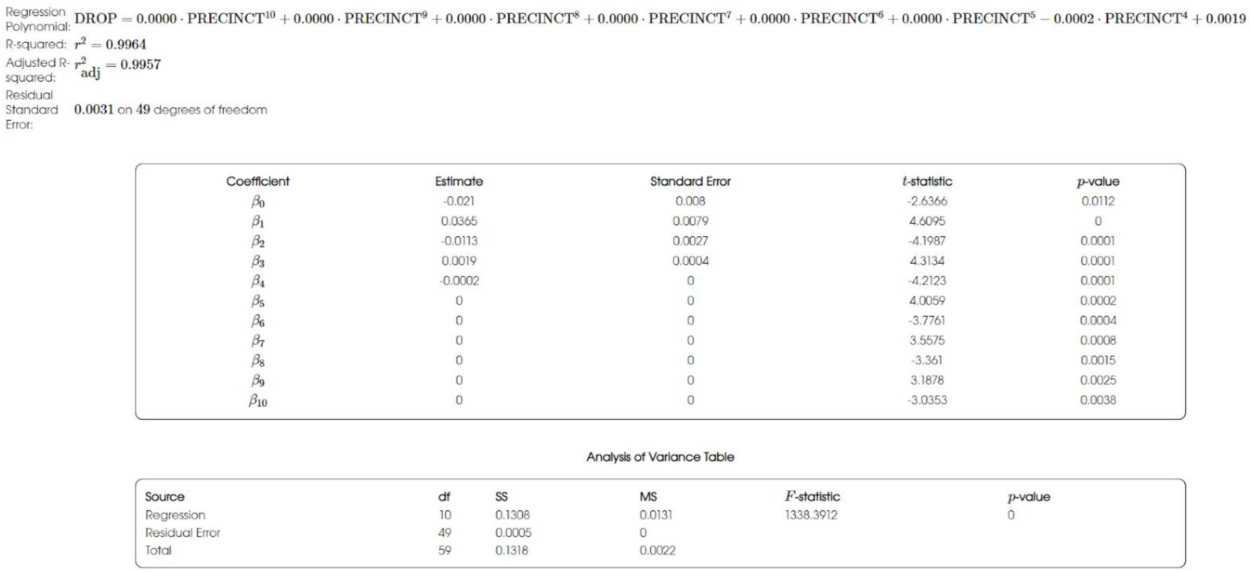
```
 4.2746943894595224e-001,
-1.5326156765998397e-001,
 2.1244043133314738e-002,
-1.4914149427036336e-003,
 6.1471000271997819e-005,
-1.5868305875228917e-006,
 2.6365144539362871e-008,
-2.8146871089117555e-010,
 1.8638283835865803e-012,
-6.9615227754315517e-015,
 1.1203565629165077e-017
```

Detroit Hist, Bucket Size 0.0002666 for best fit:



All Remaining Counties, 010Evo

We now take all remaining precincts from the remaining counties with recorded negative changes. The regression returned a UPDATE% R^2 value for degree five and a 99.64% R^2 for degree ten. All successive degrees fail catastrophically, including degree 11 . Trump then suffers a net loss of 34,518 votes from this act.



Regression Polynomial: $\mathrm{DROP} = 0.0000 \cdot \mathrm{PRECINCT}^{10} + 0.0000 \cdot \mathrm{PRECINCT}^{9} + 0.0000 \cdot \mathrm{PRECINCT}^{8} + 0.0000 \cdot \mathrm{PRECINCT}^{7} + 0.0000 \cdot \mathrm{PRECINCT}^{6} + 0.0000 \cdot \mathrm{PRECINCT}^{5} - 0.0002 \cdot \mathrm{PRECINCT}^{4} + 0.0019$

R-squared: $r^2 = 0.9964$

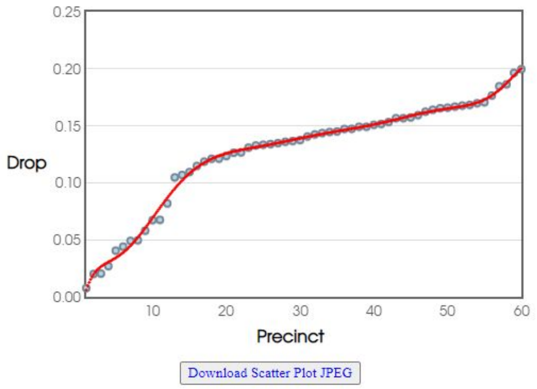
Adjusted R-squared: $r^2_{\mathrm{adj}} = 0.9957$

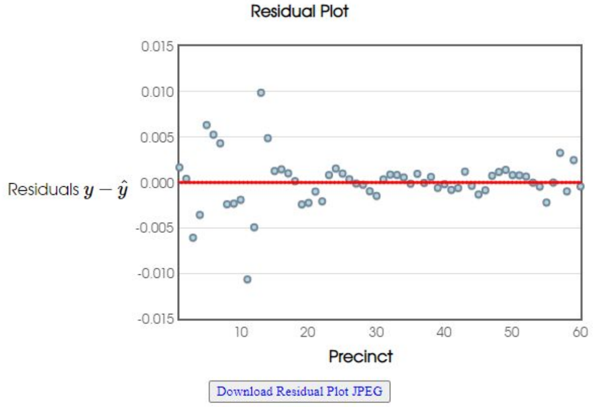
Residual Standard Error: **0.0031** on **49** degrees of freedom

Coefficient	Estimate	Standard Error	t-statistic	p-value
β_0	-0.021	0.008	-2.6366	0.0112
β_1	0.0365	0.0079	4.6095	0
β_2	-0.0113	0.0027	-4.1987	0.0001
β_3	0.0019	0.0004	4.3134	0.0001
β_4	-0.0002	0	-4.2123	0.0001
β_5	0	0	4.0059	0.0002
β_6	0	0	-3.7761	0.0004
β_7	0	0	3.5575	0.0008
β_8	0	0	-3.361	0.0015
β_9	0	0	3.1878	0.0025
β_{10}	0	0	-3.0353	0.0038

Analysis of Variance Table

Source	df	SS	MS	F-statistic	p-value
Regression	10	0.1308	0.0131	1338.3912	0
Residual Error	49	0.0005	0		
Total	59	0.1318	0.0022		

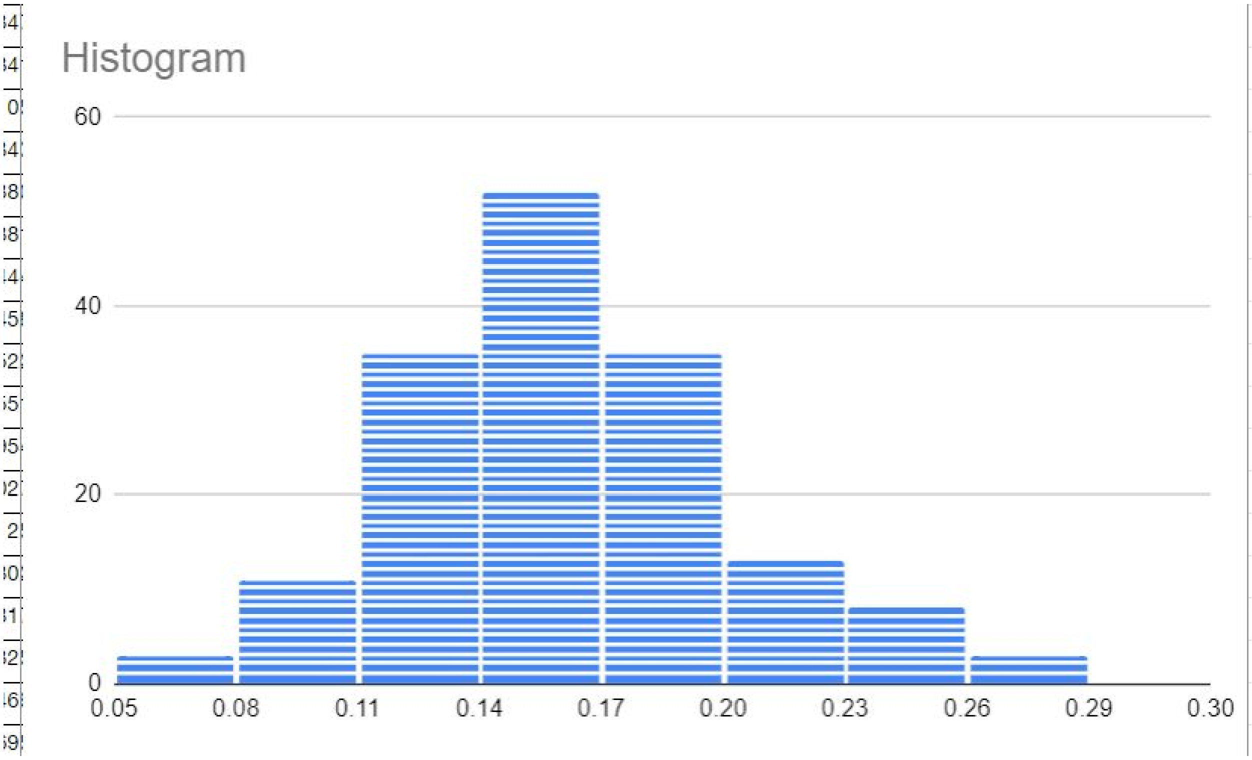




All Remaining Counties, Actual Polynomial
Mode: normal x,y analysis
Polynomial degree 10, 160 x,y data pairs.
Correlation coefficient = 0.9990098240610615
Standard error = 0.12800858300649948

```
-4.4756011437802890e+001,
 8.8320639686920011e+000,
-6.1323158446672588e-001,
 2.4132851892921900e-002,
-5.8896794872416902e-004,
 9.3322475711270948e-006,
-9.7618546395537602e-008,
 6.6849241163281526e-010,
-2.8797011919923757e-012,
 7.0720837718086629e-015,
-7.5431309525061190e-018
```

All Remaining Counties, Bucket Size 0.03% for best fit:



Conclusion of Regression Analysis, The Polynomic Sledgehammer

The total sum of votes stolen from President Donald J. Trump, by this algorithm, hitherto named the Polynomic Sledgehammer, is: 287,980 votes; bear in mind that Trump lost the state by 154,188 votes.

Let it be known, that natural data (in this manner) should be distributed *normally*, such that it has a strong linear regression, with a second derivative of zero (third model).

Let it be known, that purely randomly generated computer data (second model of random number generation between two bounds), is distributed *evenly,* such that it also has a strong linear regression, with a second derivative of zero.

Let it be known, that cumulative randomly generated computer data (first model) is not distributed *evenly,* and that it has strong exponential regression. As witnessed above, Detroit's regression of the maximum delta (greatest percentage drop in each precinct), perfectly fits the model of Cumulative Random Number Generation.

Let it be known, that no weak polynomial of degree five (the weak polynomial of degree *k*), nor its corresponding strong polynomial of degree ten (the strong polynomial of degree 2*k*), can capture the tail ends of natural delta regression (the polynomial always exits the bounds of the graph); furthermore, that the distribution of such deltas, should be natural, assuming that the distribution of ratios for the county were both natural (not fabricated!) on the two consecutive timestamps (for each precinct) where Trump was winning on the former the timestamp, and then was losing on the latter timestamp.

Let it be known that the residuals to such polynomials (for natural data) all pool below the polynomial regression line on the left side of the graph, and all pool above the polynomial regression line on the right side of the graph; furthermore, that although the residuals in the main band have a very shallow sinusoidal regression...all but visible to the human eye.

Yet the actual data, for each county, has degree five polynomials with only three changes in concavity (instead of its maximal limit of four), that perfectly capture the tail ends of the delta regressions; likewise, the degree ten polynomial also has three (at most four), changes in concavity (instead of its maximal limit of nine), and that the floating point of the stats.blue calculator was insufficient to carry the regression beyond degree 12 (and often in degree 11); yet, since the data was actually fabricated to a degree five polynomial (Quintic Regression), this calculator had no issues detecting the corresponding Decic Regression of the imposed Quintic Polynomial.

It took the full power of the arachnoid.com calculator to carry these regressions beyond degree ten, since the data was not fabricated to fit a degree six or higher polynomial, and thus had no easily determined fits for degrees at or above 12 (twice the degree of six).

We also observe that the histograms show that the maximal deltas for each precinct are not naturally distributed, but we can see that the algorithm attempted its best to fake such a distribution, and failed miserably in most cases.

Finally, we observe that the residuals in the main band have a very broad sinusoidal regression, clearly visible to the human eye, where the algorithm attempted to fake the residual distances from the polynomial; furthermore, that the residuals corresponding to the tail ends of these polynomials were not concentrated below the left side, nor concentrated above the right side, but rather the algorithm simply spray painted these residual distances with a random number generator.

Section II

First Stases and Drop Stases
The Polynomic Sledgehammer

Definition of the Active First Stasis and the Drop Stasis.

For each precinct, p_i, in **P,** let $\mathbf{C}_j$ be the subset of **P**, such that all for p_i in $\mathbf{C}_j$, p_i is from the same county.

For each precinct, $c_u \in \mathbf{C}_j$, let $\mathbf{H}_j$ be the set containing the first recorded percentage for each precinct (where the total number of votes is not equal to zero), let $|\mathbf{H}_j| = h$; $|\mathbf{C}_j| = h$.

Let $\mathbf{H}_{j,2}$ be the adjoinment of the arrays $\mathbf{H}_j$ and $\mathbf{C}_j$.

Now let $\mathbf{C'}_j \subset \mathbf{C}_j$, such that each $c'_u \in \mathbf{C'}_j$ has a future stasis (that the precinct reports at least one more time after its first report; we are culling inactive precincts from the set).

Let **F** be the subset of precincts in $\mathbf{C'}_j$ that have no negatives changes in Trump's percentage over the course of their history (we are culling precincts with no negative changes).

Now let $\mathbf{C''}_j = \mathbf{C'}_j$ **- F**.

For each precinct, $c''_u \in \mathbf{C''}_j$, let $\mathbf{H''}_j$ be the set containing the first recorded percentage for each precinct in $\mathbf{C''}_j$, let $|\mathbf{H''}_j| = h\text{-}m$; $|\mathbf{C''}_j| = h\text{-}m$.

Let $\mathbf{H''}_{j,2}$ be the adjoinment of the arrays $\mathbf{H''}_j$ and $\mathbf{C''}_j$. ***This is the Active First Stasis***.

Sort $\mathbf{H''}_{j,2}$ by $\mathbf{H''}_j$ from least to greatest.
Now assign each precinct in $\mathbf{C''}_j$ a local index from 1 to (h-m) **after** the sort, and let this array be $\mathbf{X''}_j$.
Now adjoin $\mathbf{X''}_j$ to the Array $\mathbf{H''}_{j,2}$, giving us $\mathbf{H''}_{j,3}$.

For each precinct, $c''_u \in \mathbf{C''}_j$, let $\mathbf{E''}_j$ be the set containing the largest negative drop in Trump's percentage for the entire history of that precinct; thus $|\mathbf{E''}_j| = h\text{-}m$.

Now for each precinct, $c''_u \in \mathbf{C''}_j$, let $\mathbf{G''}_j$ be the set of each recorded percentage on the timestamp that corresponds to the respective precinct, c''_u, in $\mathbf{E''}_j$; thus $|\mathbf{G''}_j| = h\text{-}m$.

Let $\mathbf{G''}_{j,2}$ be the adjoinment of the arrays $\mathbf{G''}_j$ and $\mathbf{C''}_j$. ***This is the Active Drop Stasis***.
Sort $\mathbf{G''}_{j,2}$ by $\mathbf{G''}_j$ from least to greatest.

Now assign each precinct in $\mathbf{C''}_{j}$ a local index from 1 to (h-m) after the sort, and let this array be $\mathbf{Y''}_{j}$.
Now adjoin $\mathbf{Y''}_{j}$ to the Array $\mathbf{G''}_{j,2}$, giving us $\mathbf{G''}_{j,3}$.

Soon we shall reset the values of $\mathbf{Y''}_{j}$ in accordance to the values $\mathbf{X''}_{j}$, in order to allow us to track the movement of each precinct before and after the application of the Polynomic Sledgehammer.

Now resort $\mathbf{H''}_{j,3}$ by $\mathbf{C''}_{j}$ (the precinct component).
Now resort $\mathbf{G''}_{j,3}$ by $\mathbf{C''}_{j}$ (the precinct component).

Now adjoin $\mathbf{H''}_{j,3}$ and $\mathbf{G''}_{j,3}$ into the new array $\mathbf{V''}_{j,6}$, maintaining the order of the columns.

Now resort $\mathbf{V''}_{j,6}$ by $\mathbf{X''}_{j}$, from least to greatest.

Now extract the $\mathbf{X''}_{j}$ and $\mathbf{Y''}_{j}$ columns of $\mathbf{V''}_{j,6}$.

Now plot the linear regression of each x in $\mathbf{V''}_{j,6}$ against each corresponding y in $\mathbf{V''}_{j,6}$.

Algorithm used to resort the order of the precincts when input into the Polynomial: (Old) The Golden Algorithm

It was noted that Oakland's linear regression (and residual plot) for the positions of the precincts when ordered from least to greatest by their ratio in their Active First Stasis (x-axis) against their new positions when ordered from least to greatest by their ratio in their Active Drop Stasis matched the following algorithm (for which we shall now apply as a test to all of the counties in the dataset):

Let $\boldsymbol{\tau}$ be equal to the number of precincts in $\mathbf{V''}_{\mathbf{j,6}}$, in Oakland $n = 380$; $380 = h\text{-}m$.

Let $z = \boldsymbol{\pi}/\boldsymbol{\tau}$.

Let $w = k{*}\sin(zx)^{\mathrm{m}}$, for Oakland, k was equal to 100, the greatest residual in the linear regression, excluding the very obvious outliers; m was equal to 1.

Now round w down to the nearest integer (floor).

Let r =NORMINV(RAND(),0,(0.34)*w); where zero in the mean, and 34% is one standard deviation above the mean, and w is the input. RAND allows us to choose random probability from a normal distribution with a mean of zero and a standard deviation of (0.34)w.

Excel NORMINV Function (excelfunctions.net)

The Excel NORMINV function calculates the inverse of the Cumulative Normal Distribution Function for a supplied value of x, and a supplied distribution mean & standard deviation.

Now round r down to the nearest integer (floor), and let c be the scaling factor for r, such that the residual distance from the linear regression in the form of $mx+b$ will be $r(c)$. For Oakland, $c = 5/3$.

Let $\mathbf{R''}_{\mathbf{j}}$ be the array for each residual generated for each index of $\mathbf{X''}_{\mathbf{j}}$,

In Oakland $\mathbf{m} = 0.9$ and $b = 19$.

When we add each $r \in \mathbf{R''}_{\mathbf{j}}$ to each y generated by the line $y = 0.9x + 19$ for each $x \in \mathbf{X''}_{\mathbf{j}}$, we resulted with the exact same distribution, regression and residual plot and correlation of the actual linear regression of each x in $\mathbf{V''}_{\mathbf{j,6}}$ against each corresponding y in $\mathbf{V''}_{\mathbf{j,6}}$ in Oakland.

To create a bijection between each $x \in \mathbf{X''}_\mathrm{j}$ and each $r \in \mathbf{R''}_\mathrm{j}$, the algorithm then truncates the indexes of $\mathbf{X''}_\mathrm{j}$ using a Fibonacci Sequence.

Let F(n) be the first Fibonacci Number less than one-half the number of precincts in $\mathbf{V''}_{\mathbf{u,6}}$; let F($n$-4) be the fourth Fibonacci Number that is less than one-half the number of precincts in $\mathbf{V''}_{\mathbf{u,6}}$, which is $\boldsymbol{\tau}/2$.

Quarantine the indices of $\mathbf{X''}_\mathrm{j}$ in the following order:

Let $\mathbf{X}_1 \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ 1 \le x_\mathrm{k} \le 1$

Let $\mathbf{X}_2 \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ 2 \le x_\mathrm{k} \le 2$

Let $\mathbf{X}_3 \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ 3 \le x_\mathrm{k} \le 4$

Let $\mathbf{X}_4 \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ 5 \le x_\mathrm{k} \le 7$

Let $\mathbf{X}_5 \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ 8 \le x_\mathrm{k} \le 12$

...

Let $\mathbf{X}_\mathbf{k} \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ \mathrm{F}(k) \le x_\mathrm{k} \le -1+\mathrm{F}(k+1)$

...

Let $\mathbf{X}_{\mathbf{n-4}} \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ \mathrm{F}(n\text{-}4) \le x_\mathrm{k} \le -1+\mathrm{F}(n\text{-}3)$

Then:

Let $\mathbf{X}_{\mathbf{n-4+2}} \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ (1+\boldsymbol{\tau})\text{-}1 \le x_\mathrm{k} \le (1+\boldsymbol{\tau})\text{-}1$

Let $\mathbf{X}_{\mathbf{n-4+3}} \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ (1+\boldsymbol{\tau})\text{-}2 \le x_\mathrm{k} \le (1+\boldsymbol{\tau})\text{-}2$

Let $\mathbf{X}_{\mathbf{n-4+4}} \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ (1+\boldsymbol{\tau})\text{-}3 \le x_\mathrm{k} \le (1+\boldsymbol{\tau})\text{-}4$

Let $\mathbf{X}_{\mathbf{n-4+5}} \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ (1+\boldsymbol{\tau})\text{-}5 \le x_\mathrm{k} \le (1+\boldsymbol{\tau})\text{-}7$

Let $\mathbf{X}_{\mathbf{n-4+6}} \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ (1+\boldsymbol{\tau})\text{-}8 \le x_\mathrm{k} \le (1+\boldsymbol{\tau})\text{-}12$

...

Let $\mathbf{X}_{\mathbf{n-4+c}} \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ (1+\boldsymbol{\tau})\text{-}\mathrm{F}(c\text{-}1) \le x_\mathrm{k} \le (1+\boldsymbol{\tau}) - (-1+\mathrm{F}(c\text{-}1+1))$

...

Let $\mathbf{X}_{\mathbf{2(n-4)+1}} \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ (1+\boldsymbol{\tau})\text{-}\mathrm{F}(n\text{-}4) \le x_\mathrm{k} \le (1+\boldsymbol{\tau})\text{-}(-1+\mathrm{F}(n\text{-}3))$

Finally let:

Let $\mathbf{X}_{\mathbf{n-4+1}} \subset \mathbf{X''}_\mathrm{j}$ $\forall x,\ \mathrm{F}(n\text{-}3) \le x_\mathrm{k} \le (\boldsymbol{\tau})\text{-}\mathrm{F}(n\text{-}3))$

Now sort each quarantined partition by $\mathbf{R''}_\mathrm{j}$ from least to greatest.
Now recombine the quarantined partitions to reform $\mathbf{V''}_{\mathbf{u,\,6}}$.

Now set each $y \in \mathbf{Y''}_\mathrm{j}$ equal to each $x \in \mathbf{X''}_\mathrm{j}$; **AFTER** this, resort only $\mathbf{X''}_\mathrm{j}$ from least to greatest. The map from $\mathbf{X''}_\mathrm{j}$ to $\mathbf{Y''}_\mathrm{j}$ is how the precincts in their starting order, by ratio, from least to greatest in respect to $\mathbf{X''}_\mathrm{j}$, are assigned their input value into the new polynomial. In words, this is the bijective map between the domains of both polynomials.

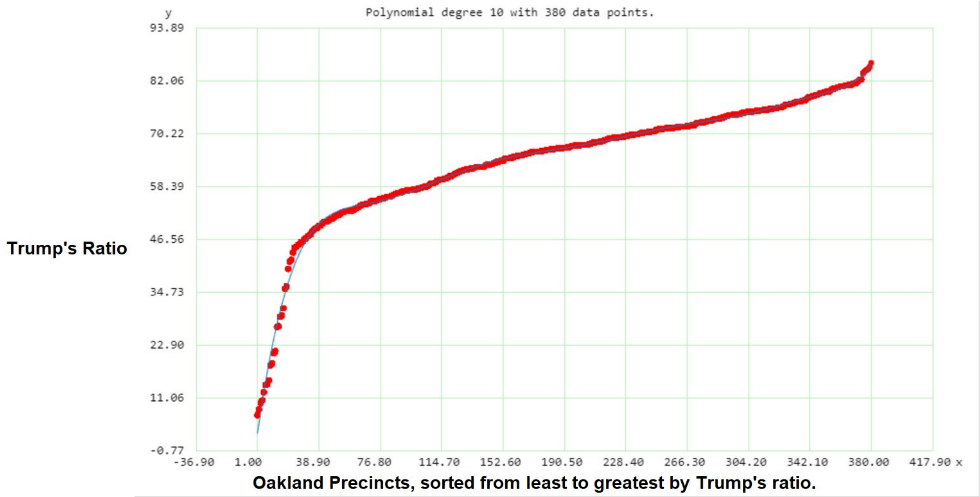
Now set each $g \in \mathbf{V''}_{\mathbf{j,\,6}}$ to $g_\mathrm{i} = (\mathrm{a}_{10})y^{10}+(\mathrm{a}_9)y^9+(\mathrm{a}_8)y^8+\ldots(\mathrm{a}_1)y+(\mathrm{a}_0)$
The values in $\mathbf{G''}_{\mathbf{j,3}}$ are Trump’s new percentages after the application of the decic polynomial.

First Active Stasis of Oakland, 011Evo

The First Stasis of Detroit. The regression returned a 99.61% R^2 for degree ten.

For each $h \in \mathbf{V''}_{\mathbf{j, 6}}$, set $h_i = (a_{10})x^{10}+(a_9)x^9+(a_8)x^8+...(a_1)x+(a_0)$

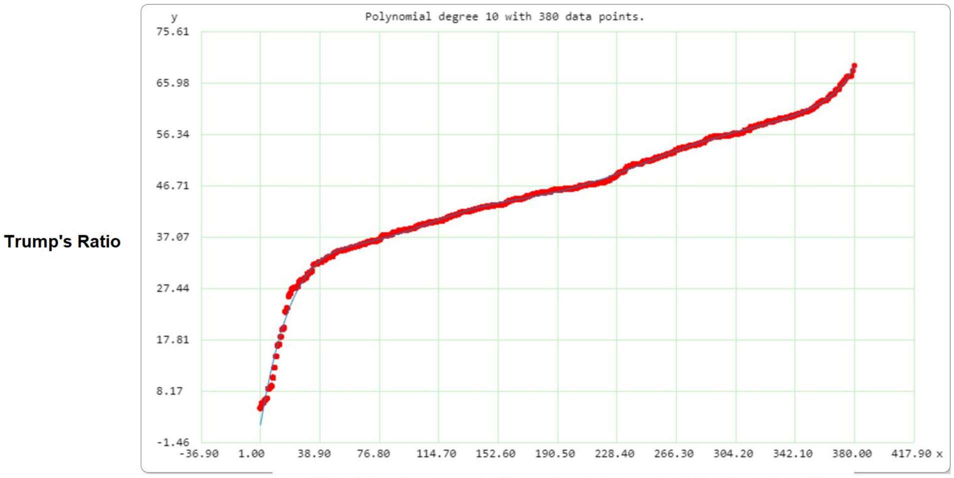
Mode: normal x,y analysis
Polynomial degree 10, 380 x,y data pairs.
Correlation coefficient = 0.9967999131719192
Standard error = 0.7917976045395744
a0=4.6600876365549948e-001,
a1=2.5995972693563800e+000,
a2= -4.2945965823361680e-002,
a3= 9.9474910443903540e-005,
a4= 5.8373747448365518e-006,
a5= -8.6027254591587400e-008,
a6= 5.8605675748832633e-010,
a7= -2.2808061248816555e-012,
a8= 5.2000442772899654e-015,
a9= -6.4777658181934627e-018,
a10=3.4108375424397E-021



The Active Drop Stasis of Oakland. The regression returned a 99.61% R^2 for degree ten.

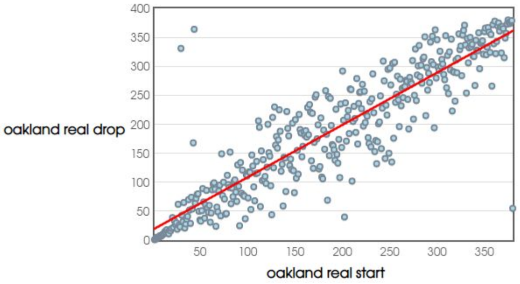
For each $g \in \mathbf{V''}_{\mathbf{j,\,6}}$ to $g_i = (a_{10})y^{10}+(a_9)y^9+(a_8)y^8+...(a_1)y+(a_0)$

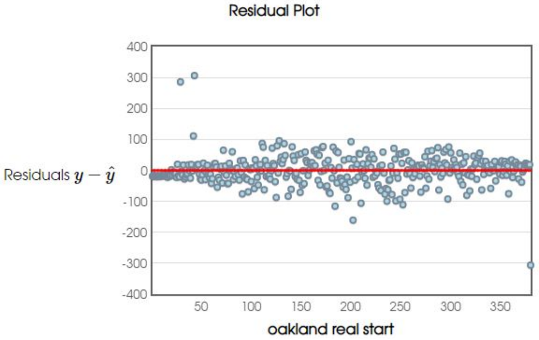
```
Mode: normal x,y analysis
Polynomial degree 10, 380 x,y data pairs.
Correlation coefficient = 0.9967999131719192
Standard error = 0.7917976045395744
a0=   3.2261387627234295e-002,
a1=    1.7796330339497057e+000,
a2=   -3.3799739084705890e-002,
a3=    2.3297815041056638e-004,
a4=    1.3006043144617245e-006,
a5=   -3.3973840419950054e-008,
a6=    2.5880133055409800e-010,
a7=   -1.0466190825701117e-012,
a8=    2.4140324718631769e-015,
a9=   -3.0044928531493336e-018,
a10=   1.57119193762813E-021
```

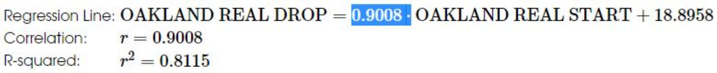


Oakland Precincts, sorted from least to greatest by Trump's ratio.

Actual Positions of the precincts in the first polynomial (x-axis) vs. the *Actual Positions* of the precincts in the second polynomial.

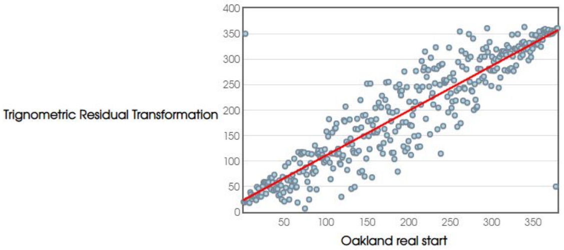


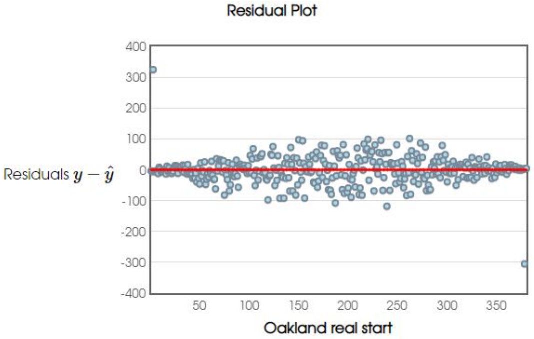


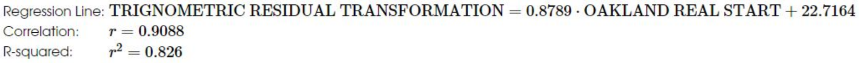


Regression Line: $\mathrm{OAKLAND\ REAL\ DROP} = 0.9008 \cdot \mathrm{OAKLAND\ REAL\ START} + 18.8958$
Correlation: $r = 0.9008$
R-squared: $r^2 = 0.8115$

Simulated Positions of the precincts in the first polynomial (x-axis) vs. the *Simulated Positions* of the precincts in the second polynomial using the algorithm on pages 42-43:

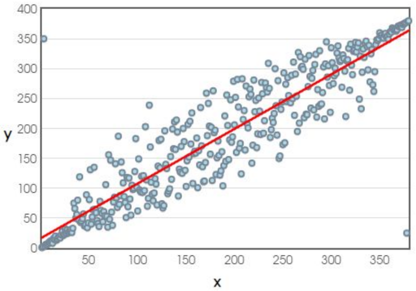


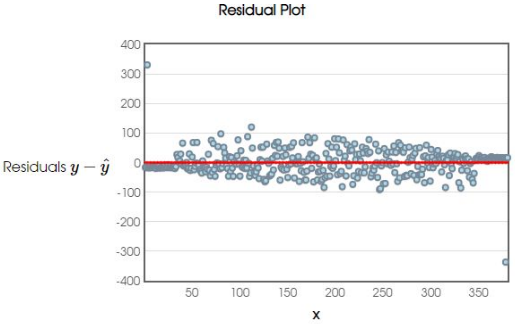


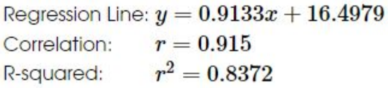


Regression Line: $\mathrm{TRIGNOMETRIC\ RESIDUAL\ TRANSFORMATION} = 0.8789 \cdot \mathrm{OAKLAND\ REAL\ START} + 22.7164$
Correlation: $r = 0.9088$
R-squared: $r^2 = 0.826$

Post Fibonacci sort:







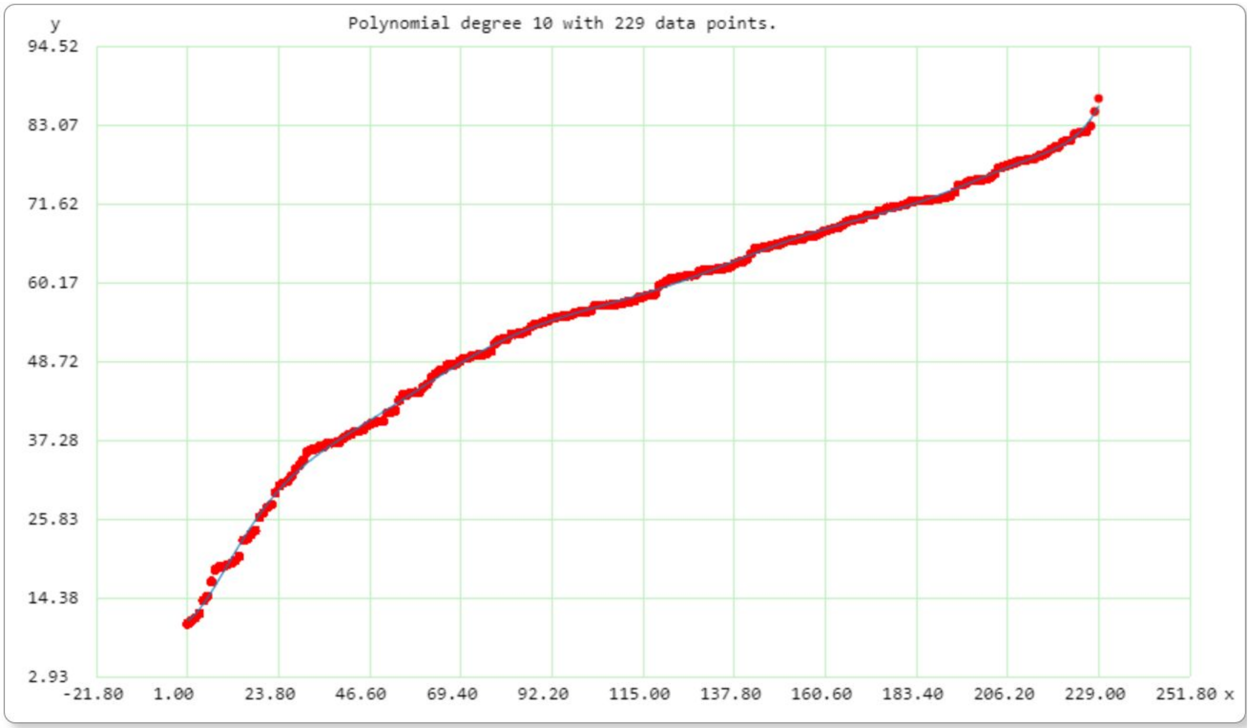
Regression Line: $y = 0.9133x + 16.4979$
Correlation: $r = 0.915$
R-squared: $r^2 = 0.8372$

First Active Stasis of Kent, 011KentActiveRegression

The Active First Stasis of Kent. The regression returned a 99.67% R^2 for degree ten.

For each $h \in \mathbf{V''}_{\mathbf{j,\,6}}$, set $h_{\mathrm{i}} = (\mathrm{a}_{10})x^{10}+(\mathrm{a}_9)x^9+(\mathrm{a}_8)x^8+...(\mathrm{a}_1)x+(\mathrm{a}_0)$

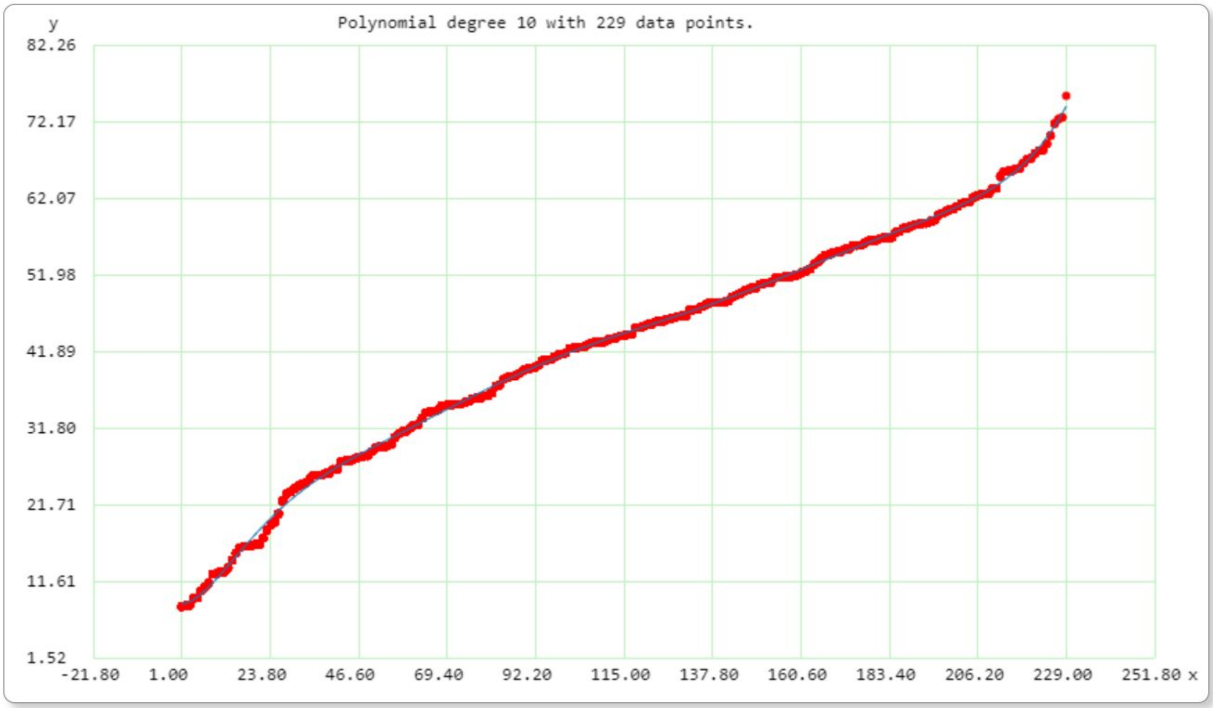
```
Mode: normal x,y analysis
Polynomial degree 10, 380 x,y data pairs.
Correlation coefficient = 0.9967999131719192
Standard error = 0.7917976045395744
a0=    1.0784805325640800e+001,
a1=    1.5803693211342074e-001,
a2=    1.0015142551323464e-001,
a3=   -5.5330306524332441e-003,
a4=    1.4670824968965408e-004,
a5=   -2.2473263824033006e-006,
a6=    2.1173091206258551e-008,
a7=   -1.2457519446206591e-010,
a8=    4.4624049302323303e-013,
a9=   -8.9050267524778307e-016,
a10=   7.59265667483203E-019
```



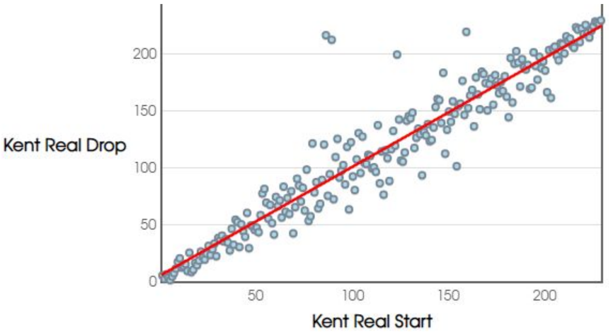
The Active Drop Stasis of Kent. The regression returned a 99.9% R^2 for degree ten.

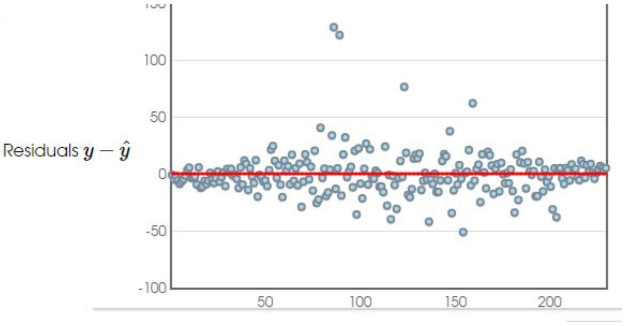
For each $g \in \mathbf{V''}_{\mathbf{j,6}}$ to $g_i = (a_{10})y^{10}+(a_9)y^9+(a_8)y^8+...(a_1)y+(a_0)$

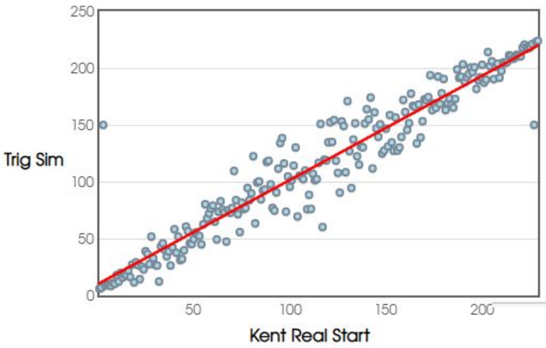
```
Mode: normal x,y analysis
Polynomial degree 10, 229 x,y data pairs.
Correlation coefficient = 0.9991243388585443
Standard error = 0.47202580025082186

a0=8.6190852171538204e+000,
a1=   -6.3758086465988878e-002,
a2=    6.0645145329730400e-002,
a3=   -2.6196689034361257e-003,
a4=    5.7217562722577090e-005,
a5=   -7.3810558122249967e-007,
a6=    5.9430098172852662e-009,
a7=   -3.0241727370664750e-011,
a8=    9.4750777408537677e-014,
a9=   -1.6735698703097777e-016,
a10=1.28011208155769E-019
```

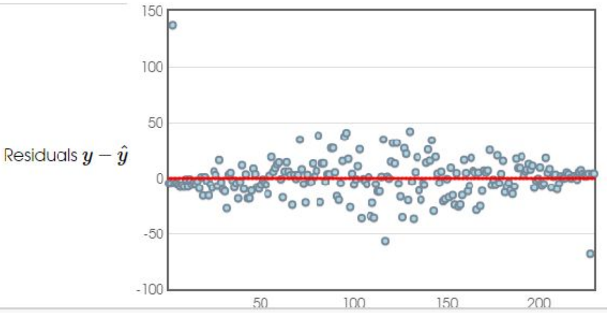


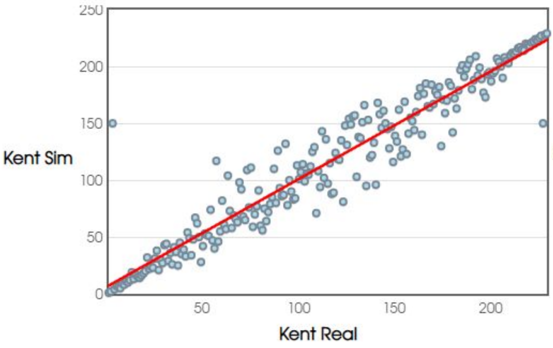
1st. Actual Positions of the precincts (Kent) in the first polynomial (x-axis) vs. the *Actual Positions* of the precincts (Kent) in the second polynomial. *2nd Simulated Positions* of the precincts in the first polynomial (x-axis) vs. the *Simulated Positions* of the precincts in the second polynomial using the algorithm on pages 42-43:
3rd Post *Fibonacci Sort*:

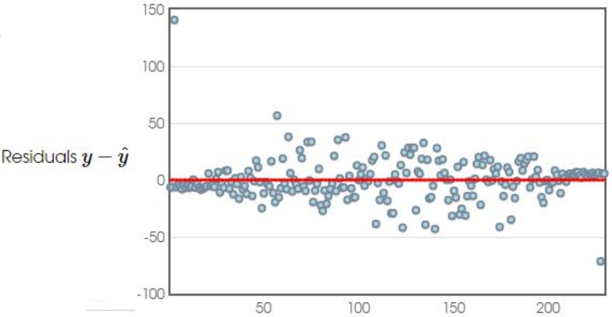












Regression Line: $\text{TRIG SIM} - 0.9171 \cdot \text{KENT REAL START} + 9.8104$
Correlation: $r - 0.9586$
R-squared: $r^2 = 0.919$

Regression Line: $\text{KENT REAL DROP} = 0.9561 \cdot \text{KENT REAL START} + 5.0498$
Correlation: $r \quad 0.9561$
R squared: $r^2 = 0.9141$

Residual Plot

Regression Line: $\text{KENT SIM} = 0.9478 \cdot \text{KENT REAL} + 6.3091$
Correlation: $r = 0.9586$
R-squared: $r^2 - 0.919$

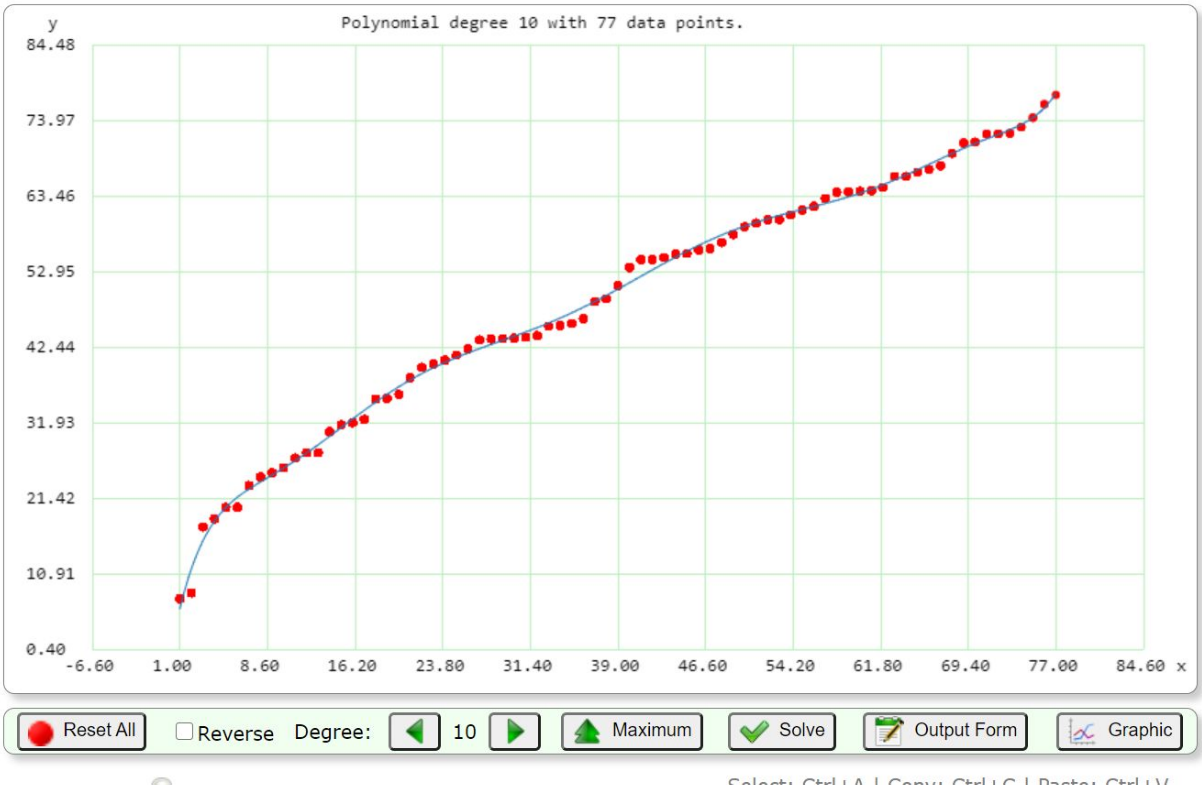
Residual Plot

First Active Stasis of Kalamazoo, 011KalamazooActiveRegression

The Active First Stasis of Kalamazoo. The regression returned a 99.74% R^2 for degree ten.

For each $h \in \mathbf{V''}_{j,6}$, set $h_i = (a_{10})x^{10}+(a_9)x^9+(a_8)x^8+...(a_1)x+(a_0)$

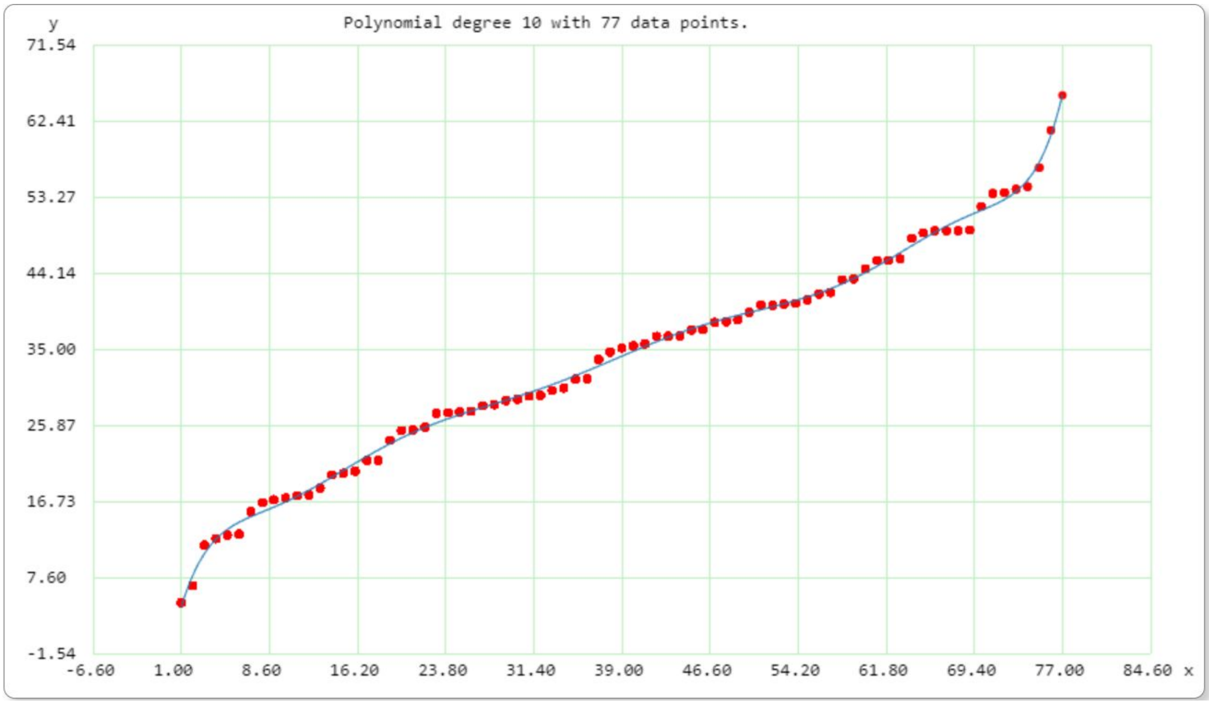
Mode: normal x,y analysis
Polynomial degree 10, 77 x,y data pairs.
Correlation coefficient = 0.9974223237114349
Standard error = 0.8938820149961619
a0=   -1.8609254231787888e+000,
a1=   9.2492372437821366e+000,
a2=   -1.5307377251025651e+000,
a3=   1.4479154812363490e-001,
a4=   -7.6482041799231810e-003,
a5=   2.3539986675720200e-004,
a6=   -4.2179965022167561e-006,
a7=   4.0625575254091940e-008,
a8=   -1.4151702202620011e-010,
a9=   -6.1309655359280639e-013,
a10=   4.69261257211385E-015



The Active Drop Stasis of Kalamazoo. The regression returned a 99.74% R^2 for degree ten.

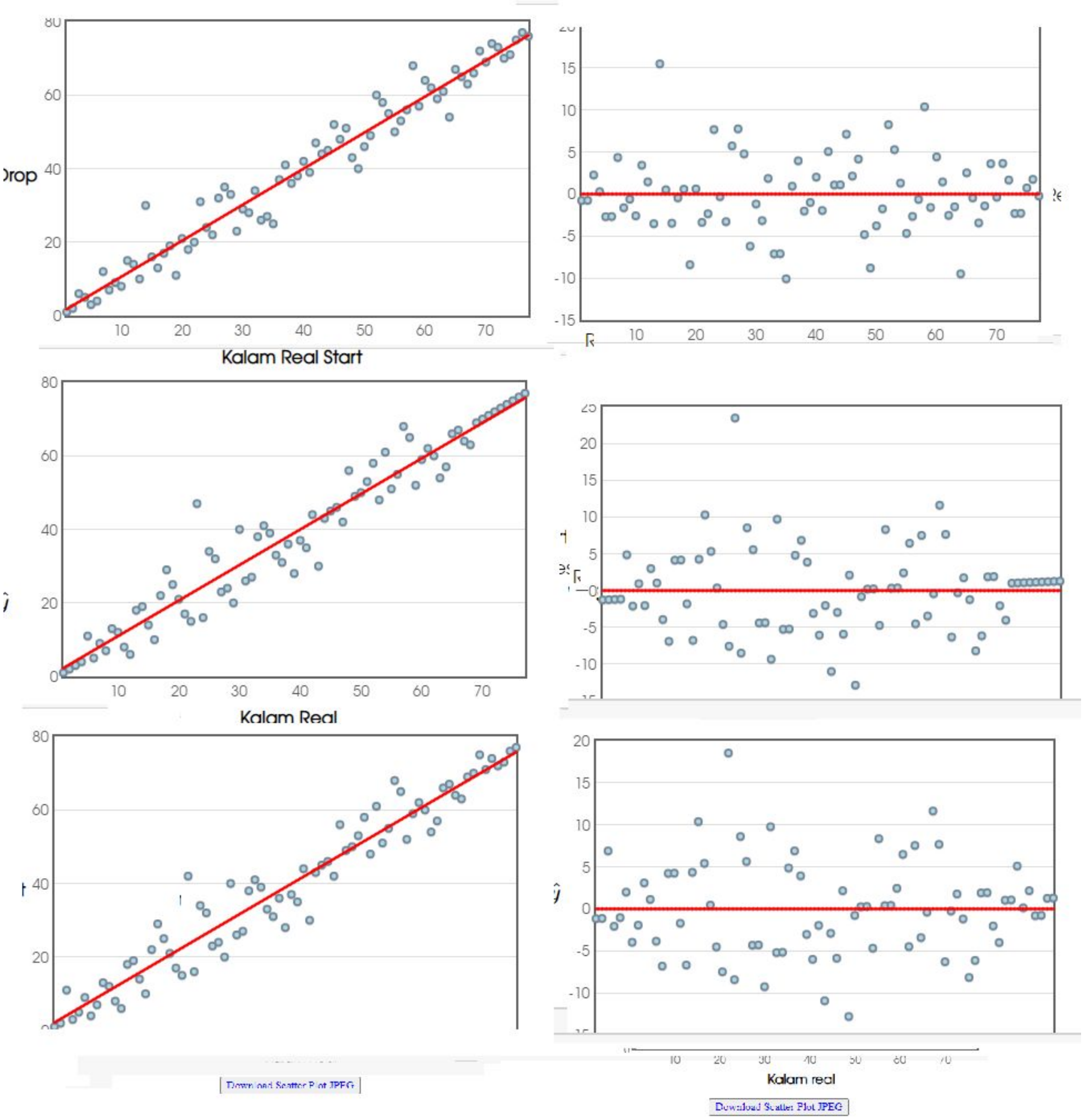
For each $g \in \mathbf{V''}_{\mathbf{j,\,6}}$ to $g_i = (a_{10})y^{10}+(a_9)y^9+(a_8)y^8+...(a_1)y+(a_0)$

```
Mode: normal x,y analysis
Polynomial degree 10, 77 x,y data pairs.
Correlation coefficient = 0.9974375299741658
Standard error = 0.686732574733712
a0=-1.6255617945686689e+000,
a1=    6.6778898327718945e+000,
a2=   -1.1857418811220237e+000,
a3=    1.1780914314600435e-001,
a4=   -6.5329937025539447e-003,
a5=    2.1244864145455940e-004,
a6=   -4.0707858126468865e-006,
a7=    4.3153790613211292e-008,
a8=   -1.9265448267903033e-010,
a9=   -3.1260432695261474e-013,
a10=4.24540021437619E-015
```



1st. Actual Positions of the precincts (Kalazamoo) in the first polynomial (x-axis) vs. the *Actual Positions* of the precincts (Kalamazoo) in the second polynomial. *2nd Simulated Positions* of the precincts in the first polynomial (x-axis) vs. the *Simulated Positions* of the precincts in the second polynomial using the algorithm on pages 42-43:

3rd Post *Fibonacci Sort*: k*sin(zx)$^{\text{m}}$, k = 10, m = 0.5



Regression Line: TRIG 0.9665 · KALAM REAL + 1.3083
Correlation: $r = 0.9665$
R-squared: $r^2 = 0.934$

Regression Line: FIB SORT = 0.9677 · KALAM REAL + 1.1962
Correlation: r 0.9684
R-squared: r^2 – 0.9377

Residual Plot

ssion Line: KALAM REAL DROP = 0.9797 · KALAM REAL START + 0.7915
lation: r 0.9797
ared: $r^2 = 0.9598$

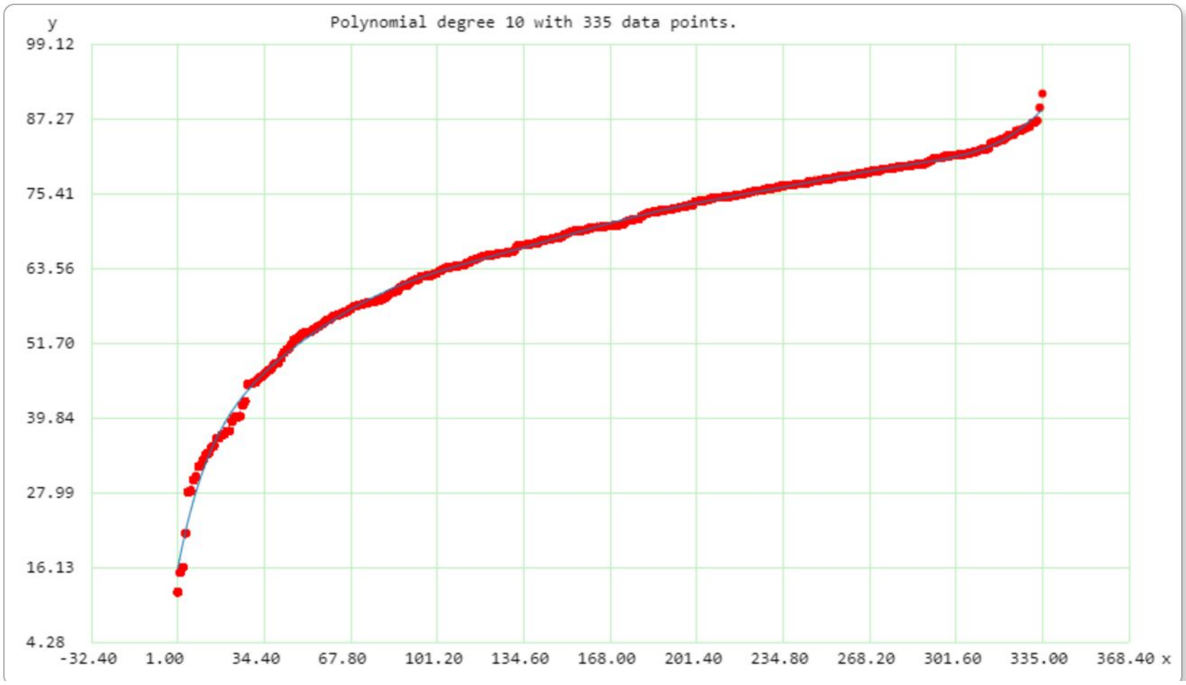
Residual Plot

First Active Stasis of Macomb, 011MacombActiveRegression

The Active First Stasis of Macomb. The regression returned a 99.74% R^2 for degree ten.

For each $h \in \mathbf{V''}_{\mathbf{j,\,6}}$, set $h_i = (a_{10})x^{10}+(a_9)x^9+(a_8)x^8+...(a_1)x+(a_0)$

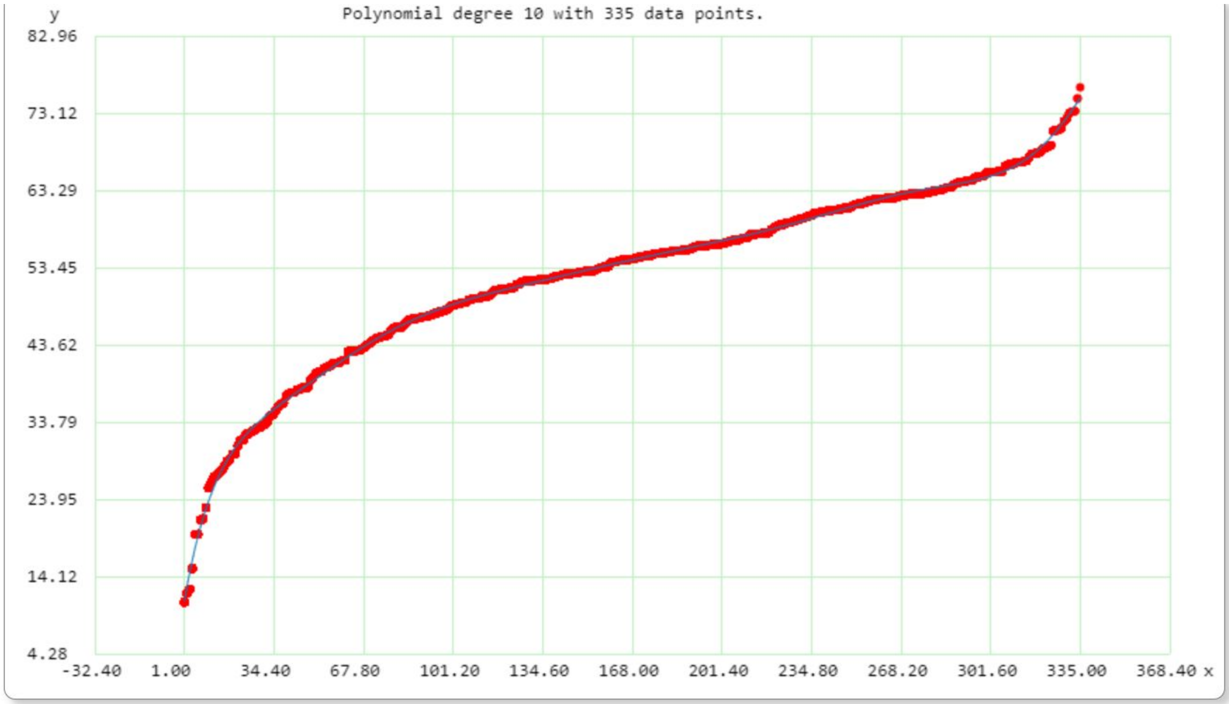
```
Mode: normal x,y analysis
Polynomial degree 10, 335 x,y data pairs.
Correlation coefficient = 0.997424710320435
Standard error = 0.727595820514032
a0= 1.3715156499883552e+001,
a1=    2.2970092385307841e+000,
a2=   -7.5870970197635529e-002,
a3=    1.6817341379880280e-003,
a4=   -2.3342451380583678e-005,
a5=    2.0582229017607551e-007,
a6=   -1.1686053383044540e-009,
a7=    4.2517975697960989e-012,
a8=   -9.5672444606133768e-015,
a9=    1.2111500825805020e-017,
a10=-6.58995988859231E-021
```



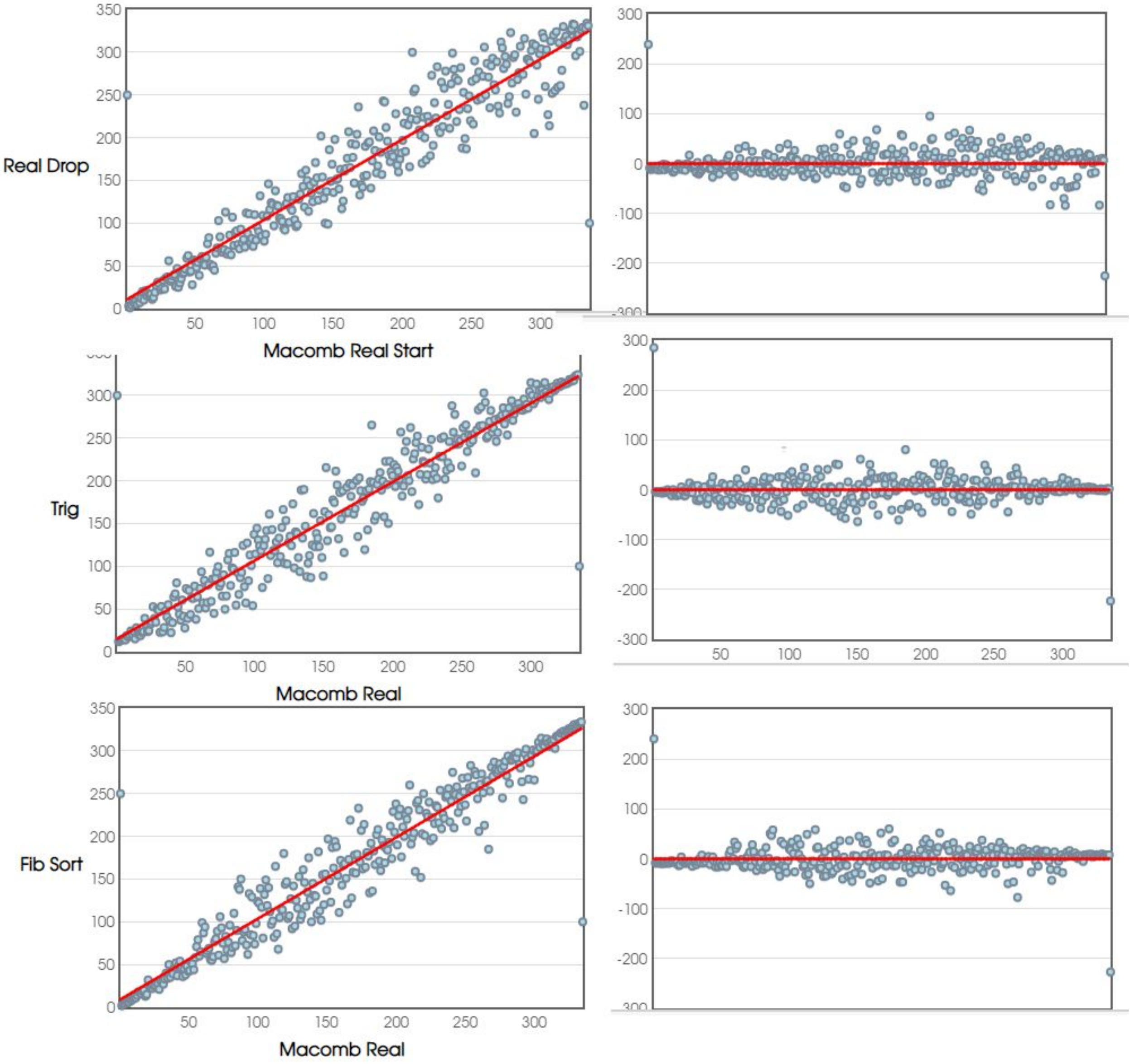
The Active Drop Stasis of Macomb. The regression returned a 99.89% R^2 for degree ten.

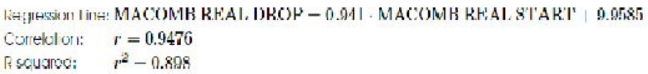
For each $g \in \mathbf{V''}_{\mathbf{j,\,6}}$ to $g_i = (a_{10})y^{10}+(a_9)y^9+(a_8)y^8+...(a_1)y+(a_0)$

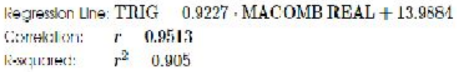
```
Mode: normal x,y analysis
Polynomial degree 10, 335 x,y data pairs.
Correlation coefficient = 0.9989822395116155
Standard error = 0.38543405595588864
a0=8.9591249205265093e+000,
a1=    2.1209816484032444e+000,
a2=   -8.1890164957138301e-002,
a3=    1.9598785247115069e-003,
a4=   -2.8199494750483026e-005,
a5=    2.5426126731439718e-007,
a6=   -1.4730909288534316e-009,
a7=    5.4779800920310911e-012,
a8=   -1.2626067172106675e-014,
a9=    1.6399533244866629e-017,
a10=-9.16401783226612E-021
```

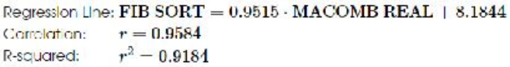


1st. Actual Positions of the precincts (Macomb) in the first polynomial (x-axis) vs. the *Actual Positions* of the precincts (Macomb) in the second polynomial. *2nd Simulated Positions* of the precincts in the first polynomial (x-axis) vs. the *Simulated Positions* of the precincts in the second polynomial using the algorithm on pages 42-43: *3rd* Post *Fibonacci Sort*: For Macomb, c = 5/3; Fibonacci quarantine stopped at 34 to 54, 54= (55-1).




Regression Line: MACOMB REAL DROP = 0.941 · MACOMB REAL START + 9.0585
Correlation: $r = 0.9476$
R-squared: $r^2 = 0.898$


Regression Line: TRIG = 0.9227 · MACOMB REAL + 13.9884
Correlation: $r = 0.9513$
R-squared: $r^2 = 0.905$


Regression Line: FIB SORT = 0.9515 · MACOMB REAL + 8.1844
Correlation: $r = 0.9584$
R-squared: $r^2 = 0.9184$

Section III

Linear Regression by Precinct, for Each County, of Trump's Net Loss Against His Starting Ratio.

Confirming Doctor Shiva

Repeat of the Definitions for Section III:

For each precinct, p_i, in $\mathbf{P}$, let $\mathbf{C}_j$ be the subset of $\mathbf{P}$, such that all for p_i in $\mathbf{C}_j$, p_i is from the same county.

For each precinct, $c_k \in \mathbf{C}_j$, let $\mathbf{H}_j$ be the set containing the first recorded percentage for each precinct (where the total number of votes is not equal to zero), let $|\mathbf{H}_j| = h$; $|\mathbf{C}_j| = h$.

Let $\mathbf{H}_{u,2}$ be the adjoinment of the arrays $\mathbf{H}_j$ and $\mathbf{C}_j$.

Now let $\mathbf{C}'_j \subset \mathbf{C}_j$, such that each $c'_k \in \mathbf{C}'_j$ has a future stasis (that the precinct reports at least one more time after its first report; we are culling inactive precincts from the set).

Let $\mathbf{F}$ be the subset of precincts in $\mathbf{C}'_j$ that have no negatives changes in Trump's percentage over the course of their history (we are culling precincts with no negative changes).

Now let $\mathbf{C}''_j = \mathbf{C}'_j - \mathbf{F}$.

For each precinct, $c''_u \in \mathbf{C}''_j$, let $\mathbf{H}''_j$ be the set containing the first recorded percentage for each precinct in $\mathbf{C}''_j$, let $|\mathbf{H}''_j| = h\text{-}m$; $|\mathbf{C}''_j| = h\text{-}m$.

Let $\mathbf{H}''_{j,2}$ be the adjoinment of the arrays $\mathbf{H}''_j$ and $\mathbf{C}''_j$. ***This is the Active First Stasis***.

Sort $\mathbf{H}''_{j,2}$ by $\mathbf{H}''_j$ from least to greatest.
Now assign each precinct in $\mathbf{C}''_j$ a local index from 1 to ($h\text{-}m$) after the sort, and let this array be $\mathbf{X}''_j$.
Now adjoin $\mathbf{X}''_j$ to the Array $\mathbf{H}''_{j,2}$, giving us $\mathbf{H}''_{j,3}$.

For each precinct, $c''_u \in \mathbf{C}''_j$, let $\mathbf{E}''_j$ be the set containing the largest negative drop in Trump's percentage for the entire history of that precinct; thus $|\mathbf{E}''_j| = h\text{-}m$.

Now for each precinct, $c''_u \in \mathbf{C}''_j$, let $\mathbf{G}''_j$ be the set of each recorded percentage on the timestamp that corresponds to the respective precinct, c''_u, in $\mathbf{E}''_j$; thus $|\mathbf{G}''_j| = h\text{-}m$.

Let $\mathbf{G}''_{j,2}$ be the adjoinment of the arrays $\mathbf{G}''_j$ and $\mathbf{C}''_j$. ***This is the Active Drop Stasis***.
Sort $\mathbf{G}''_{j,2}$ by $\mathbf{G}''_j$ from least to greatest.
Now assign each precinct in $\mathbf{C}''_j$ a local index from 1 to ($h\text{-}m$) after the sort, and let this array be $\mathbf{Y}''_j$.
Now adjoin $\mathbf{Y}''_j$ to the Array $\mathbf{G}''_{j,2}$, giving us $\mathbf{G}''_{j,3}$.

This allows us to track the movement of each precinct before and after the application of the Sledgehammer.

Now resort $\mathbf{H}''_{j,3}$ by $\mathbf{C}''_j$ (the precinct component).
Now resort $\mathbf{G}''_{j,3}$ by $\mathbf{C}''_j$ (the precinct component).

Now adjoin $\mathbf{H}''_{j,3}$ and $\mathbf{G}''_{u,3}$ into the new array $\mathbf{V}''_{j,6}$, maintaining the order of the columns.

Now resort $\mathbf{V}''_{j,6}$ by $\mathbf{X}''_j$, from least to greatest.

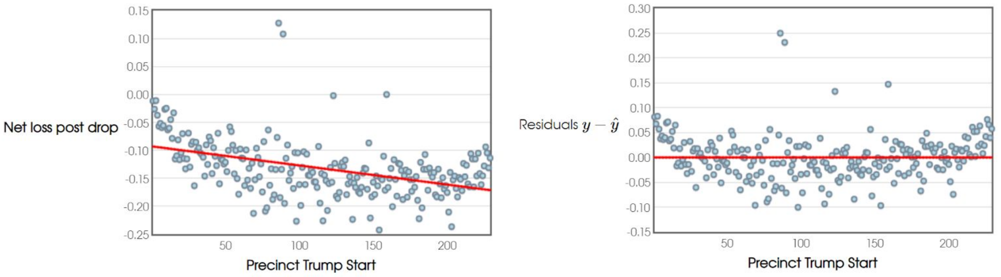
Now extract the $\mathbf{X}''_j$ and $\mathbf{Y}''_j$ columns of $\mathbf{V}''_{j,6}$.

Now plot the linear regression of each x in $\mathbf{V}''_{j,6}$ against each corresponding y in $\mathbf{V}''_{j,6}$.

Difference Between the Active First and Drop Stases of Kent, 011KentCountyActiveRegression

Start with the array $\mathbf{V''}_{j,6}$.

Resort this Array by Column $\mathbf{C''}_j$, and now plot ($\mathbf{G''}_j$- $\mathbf{H''}_j$) against $\mathbf{X''}_j$.



Regression Line: $\text{NET LOSS POST DROP} = -0.0003 \cdot \text{PRECINCT TRUMP START} - 0.0928$

Correlation: $r = -0.4491$

R-squared: $r^2 = 0.2017$

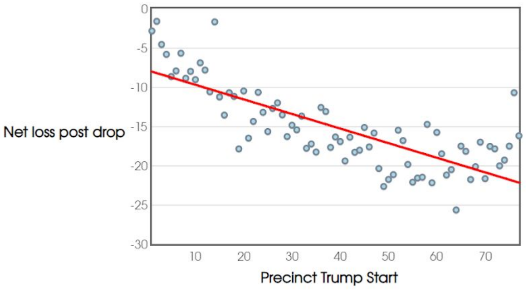
Note that Precincts that were performing better for Trump overall suffered a greater loss in their percentage for Trump after the application of the second polynomial. This confirm’s Dr Shiva’s claim that in excess of 44,000 votes were stolen from Trump via this algorithm in Kent County.

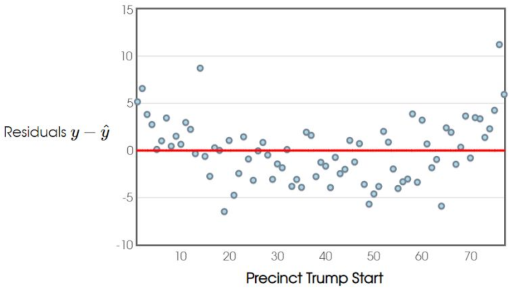
To calculate the net loss, multiply ($\boldsymbol{g''}_u$- $\boldsymbol{h''}_u$) by Trump’s Total vote in each precinct, $\boldsymbol{x''}_u$, after the application of the *Polynomic Sledgehammer*.

Difference Between the Active First and Drop Stases of Kalamazoo, 011KalamazooCountyActiveRegression

Start with the array $\mathbf{V''}_{j,6}$.

Resort this Array by Column $\mathbf{C''}_j$, and now plot ($\mathbf{G''}_j$- $\mathbf{H''}_j$) against $\mathbf{X''}_j$.





Regression Line: $\text{NET LOSS POST DROP} = -0.1861 \cdot \text{PRECINCT TRUMP START} - 7.843$
Correlation: $r = -0.782$
R-squared: $r^2 = 0.6115$

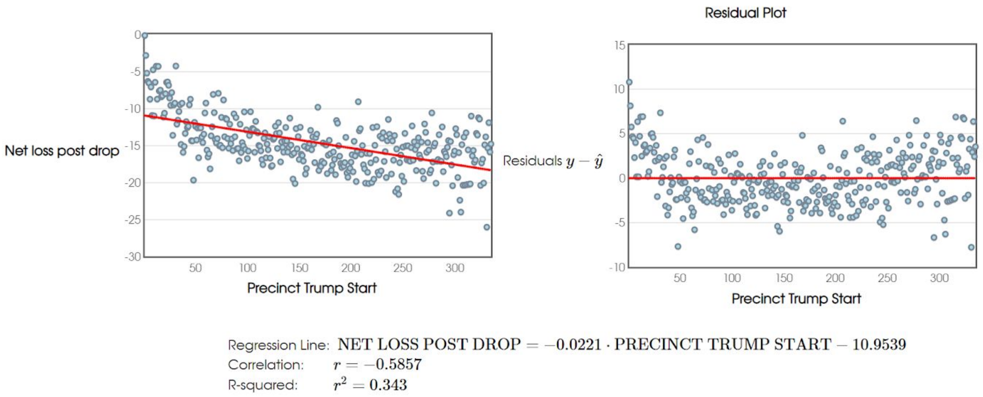
Note that Precincts that were performing better for Trump overall suffered a greater loss in their percentage for Trump after the application of the second polynomial. This results in a net loss of 16,307 votes for Trump.

To calculate the net loss, multiply ($\boldsymbol{g''}_u$- $\boldsymbol{h''}_u$) by Trump's Total vote in each precinct, $\boldsymbol{x''}_u$, after the application of the *Polynomic Sledgehammer*.

Difference Between the Active First and Drop Stases of Macomb, 011MacombCountyActiveRegression

Start with the array $\mathbf{V''}_{\mathbf{j,6}}$.

Resort this Array by Column $\mathbf{C''}_{\mathbf{j}}$, and now plot ($\mathbf{G''}_{\mathbf{j}}$- $\mathbf{H''}_{\mathbf{j}}$) against $\mathbf{X''}_{\mathbf{j}}$.



Regression Line: NET LOSS POST DROP $= -0.0221 \cdot$ PRECINCT TRUMP START $- 10.9539$
Correlation: $r = -0.5857$
R-squared: $r^2 = 0.343$

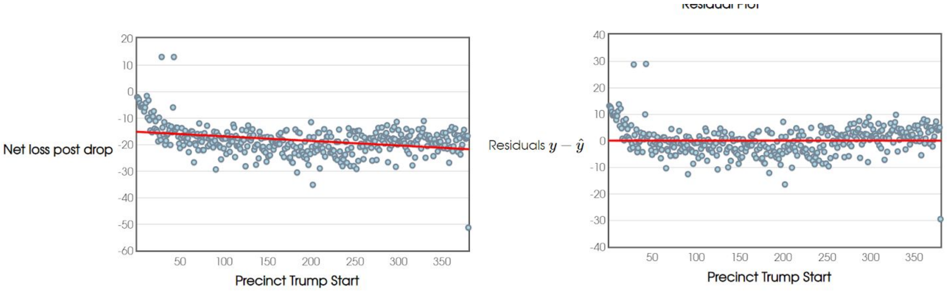
Note that Precincts that were performing better for Trump overall suffered a greater loss in their percentage for Trump after the application of the second polynomial. This results in a net loss of 72,608 votes for Trump.

To calculate the net loss, multiply ($\boldsymbol{g''}_{\mathbf{u}}$- $\boldsymbol{h''}_{\mathbf{u}}$) by Trump's Total vote in each precinct, $\boldsymbol{x''}_{\mathbf{u}}$, after the application of the *Polynomic Sledgehammer*.

Difference Between the Active First and Drop Stases of Oakland, 011OaklandCountyActiveRegression

Start with the array **$V''_{j,6}$**.

Resort this Array by Column **C''_j**, and now plot (**G''_j- H''_j**) against **X''_j**.

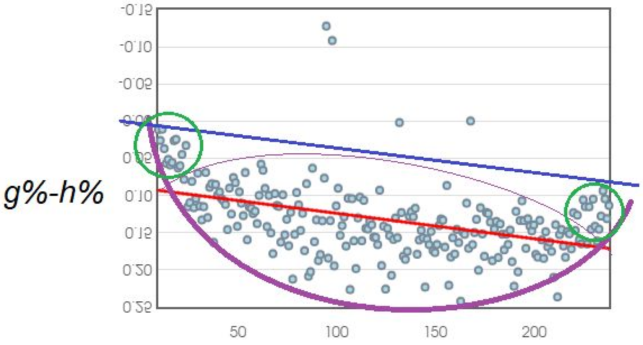


Regression Line: NET LOSS POST DROP $= -0.0173 \cdot$ PRECINCT TRUMP START $- 15.2357$
Correlation: $r = -0.3357$
R-squared: $r^2 = 0.1127$

Note that Precincts that were performing better for Trump overall suffered a greater loss in their percentage for Trump after the application of the second polynomial. This results in a net loss of 112,472 votes for Trump.

To calculate the net loss, multiply (***g''_u*- *h''_u***) by Trump's Total vote in each precinct, ***x''_u***, after the application of the *Polynomic Sledgehammer*.

Take note that in each residual plot of the above four counties, that a distinct and warped crescent moon shaped line can be drawn under the residuals, this is the effect of the polynomial itself; take note that more shallow crescent can be drawn above the residuals, this is the effect of the trigonometric bounding. The blue line is the imprint of the rotation of the polynomial.



This confirms the analysis that was done by Dr. Shiva and his claim that precincts that were performing better for Trump overall suffered a greater loss in their percentage for Trump after the application of some malicious algorithm.

Section IV

Rotation and Translation of a Polynomial Impersonating A Standard Binomial Distribution

The Wheel of Michigan

Confirming Doctor Shiva

Repeat of the Definitions for Section IV:

For each precinct, p_i, in $\mathbf{P}$, let $\mathbf{C}_j$ be the subset of $\mathbf{P}$, such that for all p_i in $\mathbf{C}_j$, p_i is from the same county.

For each precinct, $c_k \in \mathbf{C}_j$, let $\mathbf{H}_j$ be the set containing the first recorded percentage for each precinct (where the total number of votes is not equal to zero), let $|\mathbf{H}_j| = h$; $|\mathbf{C}_j| = h$.

Let $\mathbf{H}_{u,2}$ be the adjoinment of the arrays $\mathbf{H}_j$ and $\mathbf{C}_j$.

Now let $\mathbf{C}'_j \subset \mathbf{C}_j$, such that each $c'_k \in \mathbf{C}'_j$ has a future stasis (that the precinct reports at least one more time after its first report; we are culling inactive precincts from the set).

Let $\mathbf{F}$ be the subset of precincts in $\mathbf{C}'_j$ that have no negatives changes in Trump's percentage over the course of their history (we are culling precincts with no negative changes).

Now let $\mathbf{C}''_j = \mathbf{C}'_j - \mathbf{F}$.

For each precinct, $c''_u \in \mathbf{C}''_j$, let $\mathbf{H}''_j$ be the set containing the first recorded percentage for each precinct in $\mathbf{C}''_j$, let $|\mathbf{H}''_j| = h\text{-}m$; $|\mathbf{C}''_j| = h\text{-}m$.

Let $\mathbf{H}''_{j,2}$ be the adjoinment of the arrays $\mathbf{H}''_j$ and $\mathbf{C}''_j$. ***This is the Active First Stasis***.

Sort $\mathbf{H}''_{j,2}$ by $\mathbf{H}''_j$ from least to greatest.
Now assign each precinct in $\mathbf{C}''_j$ a local index from 1 to (h-m) after the sort, and let this array be $\mathbf{X}''_j$.
Now adjoin $\mathbf{X}''_j$ to the Array $\mathbf{H}''_{j,2}$, giving us $\mathbf{H}''_{j,3}$.

For each precinct, $c''_u \in \mathbf{C}''_j$, let $\mathbf{E}''_j$ be the set containing the largest negative drop in Trump's percentage for the entire history of that precinct; thus $|\mathbf{E}''_j| = h\text{-}m$.

Now for each precinct, $c''_u \in \mathbf{C}''_j$, let $\mathbf{G}''_j$ be the set of each recorded percentage on the timestamp that corresponds to the respective precinct, c''_u, in $\mathbf{E}''_j$; thus $|\mathbf{G}''_j| = h\text{-}m$.

Let $\mathbf{G}''_{j,2}$ be the adjoinment of the arrays $\mathbf{G}''_j$ and $\mathbf{C}''_j$. ***This is the Active Drop Stasis***.
Sort $\mathbf{G}''_{j,2}$ by $\mathbf{G}''_j$ from least to greatest.
Now assign each precinct in $\mathbf{C}''_j$ a local index from 1 to (h-m) after the sort, and let this array be $\mathbf{Y}''_j$.
Now adjoin $\mathbf{Y}''_j$ to the Array $\mathbf{G}''_{j,2}$, giving us $\mathbf{G}''_{j,3}$.

This allows us to track the movement of each precinct before and after the application of the Sledgehammer.

Now resort $\mathbf{H}''_{j,3}$ by $\mathbf{C}''_j$ (the precinct component).
Now resort $\mathbf{G}''_{j,3}$ by $\mathbf{C}''_j$ (the precinct component).

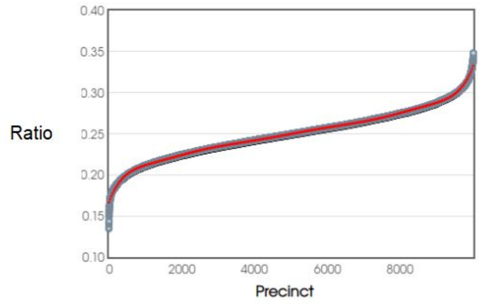
Now adjoin $\mathbf{H}''_{j,3}$ and $\mathbf{G}''_{u,3}$ into the new array $\mathbf{V}''_{j,6}$, maintaining the order of the columns.

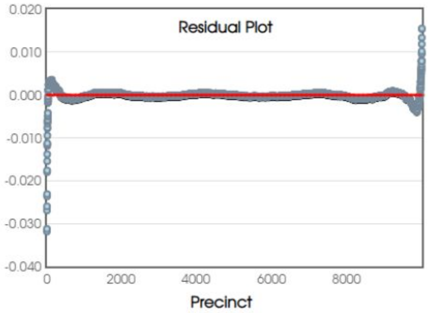
Now resort $\mathbf{V}''_{j,6}$ by $\mathbf{X}''_j$, from least to greatest.

Now extract the $\mathbf{X}''_j$ and $\mathbf{Y}''_j$ columns of $\mathbf{V}''_{j,6}$.

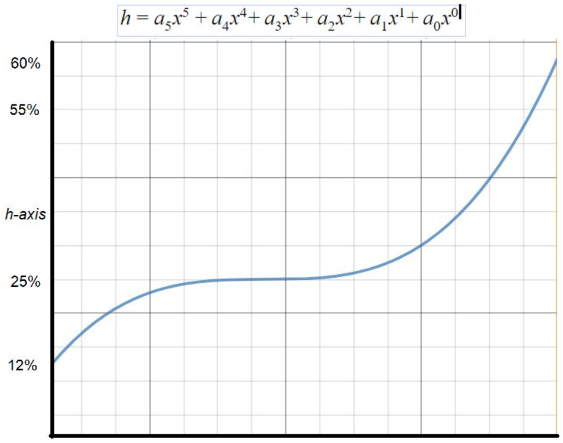
Now plot the linear regression of each x in $\mathbf{V}''_{j,6}$ against each corresponding y in $\mathbf{V}''_{j,6}$.

Quintic Approximation of the Standard Binomial Distribution of Ratios.



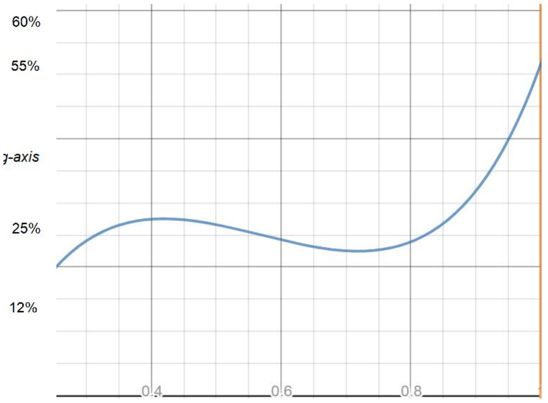


Polynomial Fabrication of a Standard Binomial Distribution



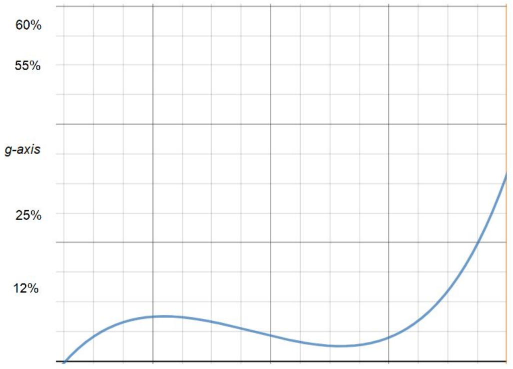
x represents the precincts in their order from least to greatest by ratio

Rotation of the Fabricated Polynomial

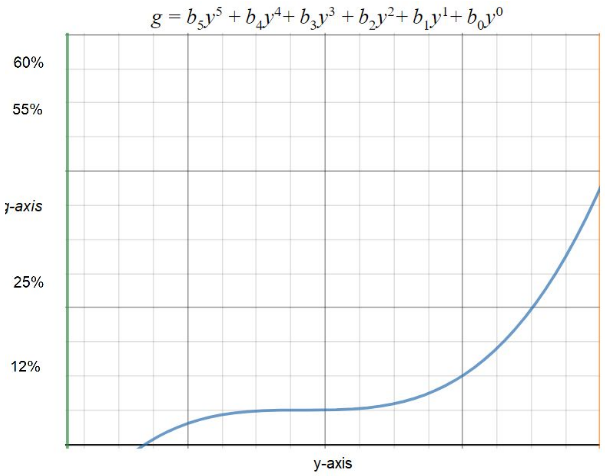


*x-axis preserves the original order of the precincts when ordered from least to greatest by their ratios in **H**. As we can see* the ordering of the precincts will change when we order them from least to greatest in respect to **G**. This is why there was such dramatic movement (reordering) of the precincts on the second application of the Polynomic Sledgehammer

Negative Translation of the Rotated Fabrication of a Standard Binomial Distribution; Precincts then reordered by Ratio (y-axis) https://www.desmos.com/calculator/uxkbbqatmt Use this link to transform, *t* in radians rotates the polynomial, *k* is the mean, the two parameters for *x=c* and *x=d* (the green and orange lines) are the relative bounds within the unit square.



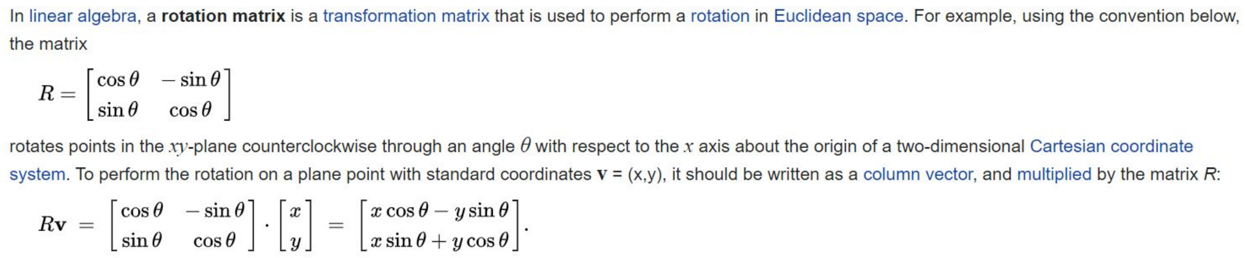
We now subtract a flat percetange from all precincts, this simply translates the entire polynomial downwards in respect to g-axis. The percentages seen after this translation are the actual percentages in **G**. The original ordering of the precincts remains.



This is the *y*-axis, the order of the precincts when ordered from least to greatest by their percentage in **G**. As you can see, this hides the rotation of the original polynomial, and how this unholy algorihtm conceals this heinous transgression. However, this algorithm goes further to conceal itself, as you will now see.

How a polynomial is rotated Using a Standard Rotation Matrix

We will use a Standard Rotation Matrix to Rotate a Polynomial:
Rotation matrix - Wikipedia , quoted from Wikipedia:



In linear algebra, a **rotation matrix** is a transformation matrix that is used to perform a rotation in Euclidean space. For example, using the convention below, the matrix

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

rotates points in the xy-plane counterclockwise through an angle θ with respect to the x axis about the origin of a two-dimensional Cartesian coordinate system. To perform the rotation on a plane point with standard coordinates **v** = (x,y), it should be written as a column vector, and multiplied by the matrix *R*:

$$R\mathbf{v} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix}.$$

We shall start with a quintic polynomial with three roots at zero, and two roots that are conjugates, as the algorithm itself does:

$$y = -v\Big(c\big((ux+10)(ux-10)\big)\Big)\Big((ux+r)^3\Big)$$

The variables *u* and *v* are used to scale the polynomial in a manner that isolates and flattens the first two concavities on either side of the origin. This allows for practical use of the Quintic. The scaling is set to compress the useful part of the polynomial (the two consecutive concavities on either side of the origin) inside of the unit circle.

Once the isolated portion of the Quintic is inside the unit circle, it allows for two things:
1: All evaluations of the polynomial are now bounded between -1 and +1, thus the absolute value of all of these evaluations are equivalent to percentages.
2: The polynomial can now be rotated inside of the unit circle with a simple rotation matrix.

Simple scaling can compress this to -50% and +50%, allowing the diabolical engineers in command of this algorithm to maintain the balance at 0%, where 0% is the relative mean at which the polynomial itself is balanced. The variable *k* in the polynomial in the next page is what is used to vertically shift the mean to desired average **after** rotation; the variable *z* is used to shift the polynomial horizontally **prior** to rotation.

The engineers can also alter the shape of the polynomial inside of this wheel as desired, use the variables *m* in this link to change the original values of the conjugate roots for the starting polynomial. The engineers will adjust the *m* variables before performing a rotation so that all of the roots of the polynomial are real (not complex) prior to the rotation. After the polynomial shape is achieved, they will simply translate the polynomial so that the change in concavity is recentered at the origin.

https://www.desmos.com/calculator/pzftlukvim

Once the polynomial is shaped, the rotation is now applied (see next page):

Using the transformation matrix, our polynomial is in the form $y = f(x)$. Thus, to rotate this polynomial we must alter the form of the *x* and *y* inputs as follows, where *t* is the angle of rotation in radians (the variable *t* in the link):

$$x \rightarrow xcos(t) - f(x)sin(t)$$
$$y_1 = f(x) \rightarrow xcos(t) + f(x)sin(t)$$

The rotated polynomial now appears as follows, $y_2 = g(x)$, the transformation is over 7 feet long, so it had to be truncated vertically (the variable *s* is a horizontal inverse scaling factor to further engineer the polynomial):

$$y_2 = k + ((-v(c((u((sx-z)\cos(t) - (-v(c((u(sx-z)+m_1)(u(sx-z)+m_2)))$$ *continue product next line*

$$((u(sx-z)+r)^3))\sin(t))+m_1)(u((sx-z)\cos(t) - (-v(c((u(sx-z)+m_1)(u(sx-z)+m_2)))$$ *continue product next line*

$$((u(sx-z)+r)^3))\sin(t))+m_2)))((u((sx-z)\cos(t) - (-v(c((u(sx-z)+m_1)(u(sx-z)+m_2)))$$ *continue product next line*

$$((u(sx-z)+r)^3))\sin(t))+r)^3))+(((sx-z)\cos(t) - (-v(c((u(sx-z)+m_1)(u(sx-z)+m_2)))$$ *continue product next line*

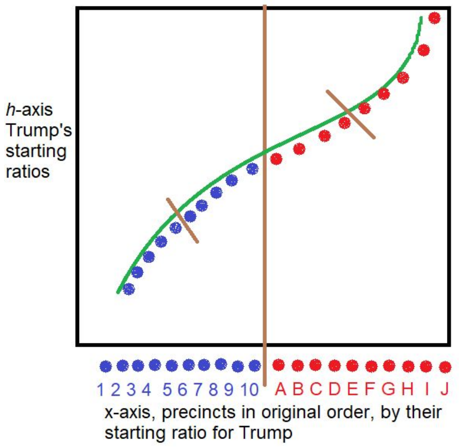
$$((u(sx-z)+r)^3))\sin(t))\sin(t)))$$ *Next line start with addition*

$$+(-v(c((u((sx-z)\cos(t) - (-v(c((u(sx-z)+m_1)(u(sx-z)+m_2)))((u(sx-z)+r)^3))\sin(t))+m_1)$$ *continue product next line*

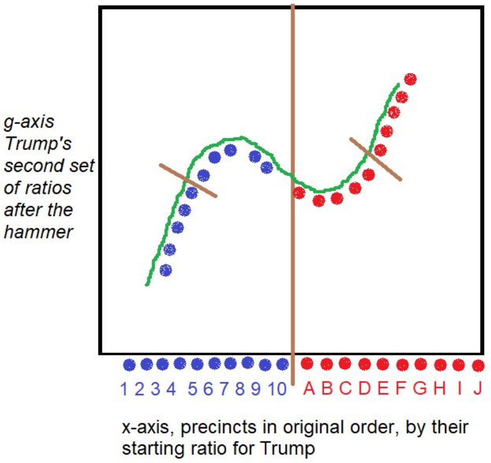
$$(u((sx-z)\cos(t) - (-v(c((u(sx-z)+m_1)(u(sx-z)+m_2)))((u(sx-z)+r)^3))\sin(t))+m_2)))$$ *continue product next line*

$$((u((sx-z)\cos(t) - (-v(c((u(sx-z)+m_1)(u(sx-z)+m_2)))((u(sx-z)+r)^3))\sin(t))+r)^3))(\cos(t)))$$

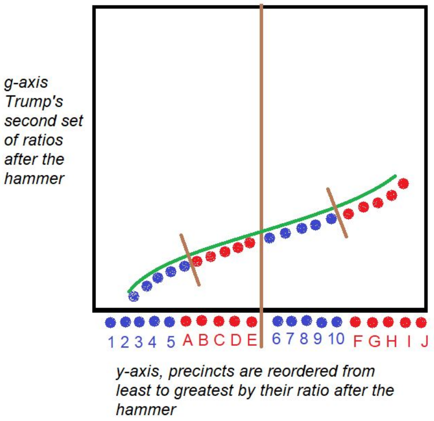
Here we see the original percentages on the first timestamp. The x-axis is the original ordering of the precincts, from least to greatest by their percentages.


h-axis
Trump's
starting
ratios
1 2 3 4 5 6 7 8 9 10 A B C D E F G H I J
x-axis, precincts in original order, by their starting ratio for Trump

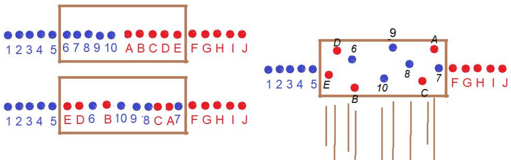
Below we see them rotated, the original ordering of the precincts remains intact in this picture.


g-axis
Trump's
second set
of ratios
after the
hammer
1 2 3 4 5 6 7 8 9 10 A B C D E F G H I J
x-axis, precincts in original order, by their starting ratio for Trump

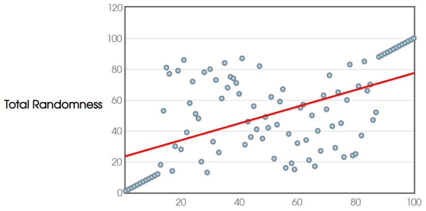
Here we see the new percentages on the second timestamp. The y-axis is the new ordering of the precincts, from least to greatest by their new percentages.


g-axis
Trump's
second set
of ratios
after the
hammer
1 2 3 4 5 A B C D E 6 7 8 9 10 F G H I J
y-axis, precincts are reordered from
least to greatest by their ratio after the
hammer

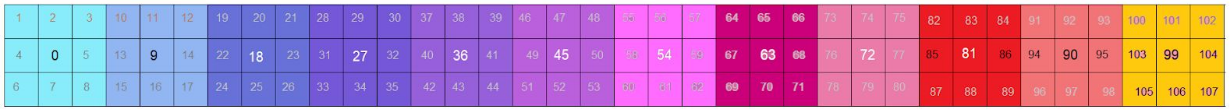
However, if they allowed half of the precincts (in red) to remain on the other side of the average line (brown), it would be too Suspicious, even an amateur would eventually detect this, so instead they five of the blue precincts and five of the red precincts nearest the average line and place them on a pool table, like billiard Balls, and collide them and use their new relative vertical on the pool table as their position in the new polynomial.


1 2 3 4 5 6 7 8 9 10 A B C D E F G H I J
1 2 3 4 5 E D 6 B 10 9 8 C A 7 F G H I J
1 2 3 4 5
D
6
9
A
8
7
E
10
B
C
F G H I J

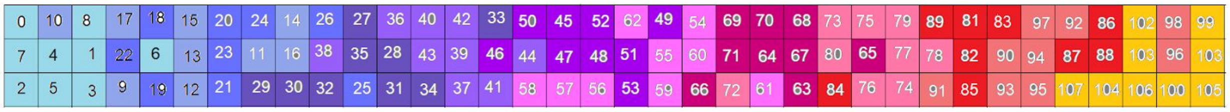
Thus, when the original domain **X** (for the Range **H**) is mapped to the new domain **Y** (for the Range **G**), they then randomize the precincts within the main band of the distribution (the region between the peak of both concavities impersonating a standard distribution); however, as you will see on the next page, the engineers running this algorithm aren't stupid enough to use a square or rectangular pool table to simulate their collisions of the billiard balls, because such a dramatic reordering of the main band would ultimately have been detectable within a fews of analysis. This would be the obvious footprint of random repositioning, while ignoring the tail ends.



This image below represents the natural change in the order precincts between consecutive timestamps.

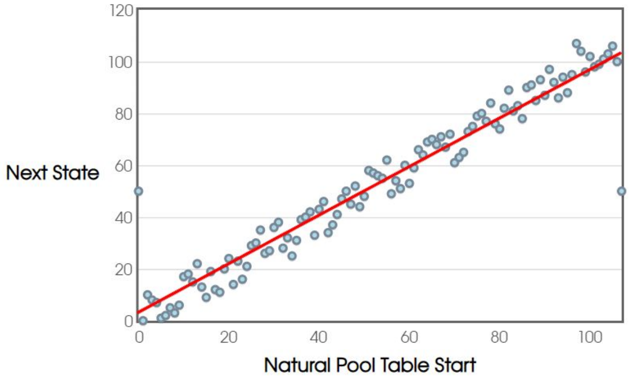


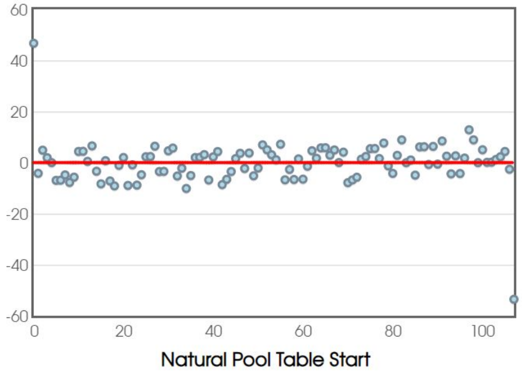
1	2	3	10	11	12	19	20	21	28	29	30	37	38	39	46	47	48	55	56	57	64	65	66	73	74	75	82	83	84	91	92	93	100	101	102
4	0	5	13	9	14	22	18	23	31	27	32	40	36	41	49	45	50	58	54	59	67	63	68	76	72	77	85	81	86	94	90	95	103	99	104
6	7	8	15	16	17	24	25	26	33	34	35	42	43	44	51	52	53	60	61	62	69	70	71	78	79	80	87	88	89	96	97	98	105	106	107



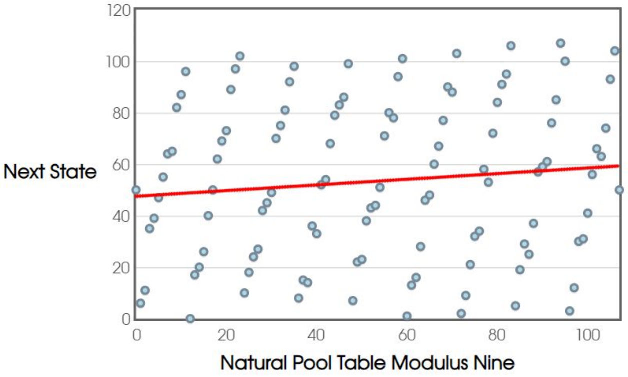
0	10	8	17	18	15	20	24	14	26	27	36	40	42	33	50	45	52	62	49	54	69	70	68	73	75	79	89	81	83	97	92	86	102	98	99
7	4	1	22	6	13	23	11	16	38	35	28	43	39	46	44	47	48	51	55	60	71	64	67	80	65	77	78	82	90	94	87	88	103	96	103
2	5	3	9	19	12	21	29	30	32	25	31	34	37	41	58	57	56	53	59	66	72	61	63	84	76	74	91	85	93	95	107	104	106	100	105

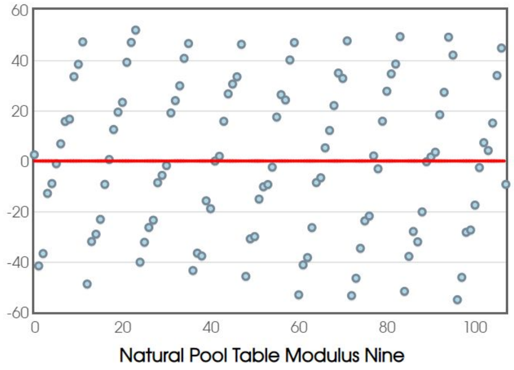
Here is the scatter plot for a such change:



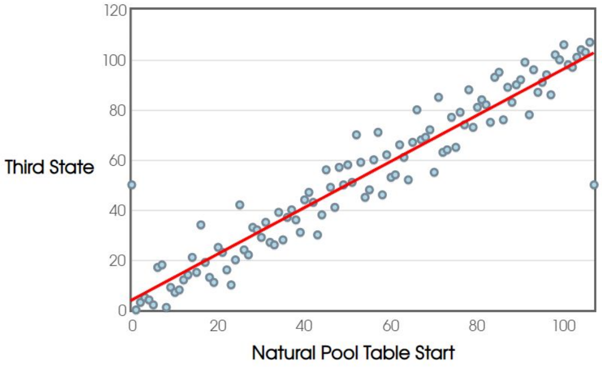


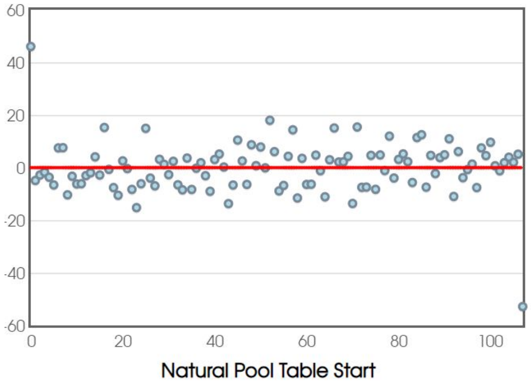
If we examine each billiard by its respective residue modulus nine, we can get an even clearer picture:



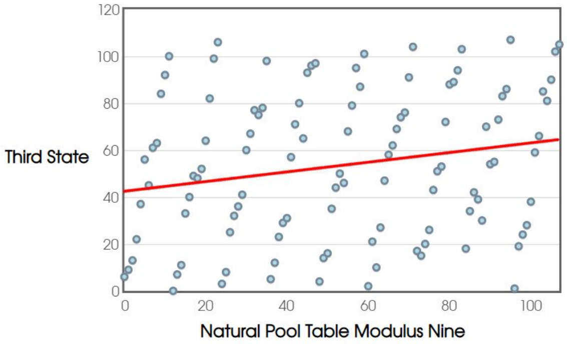


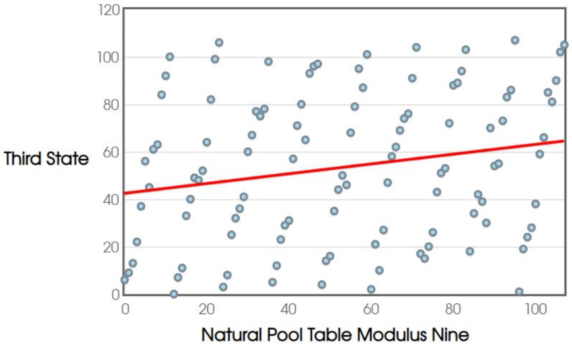
If we change collide these balls (precincts) again to third time stamp the scatter plot would appear as follows:





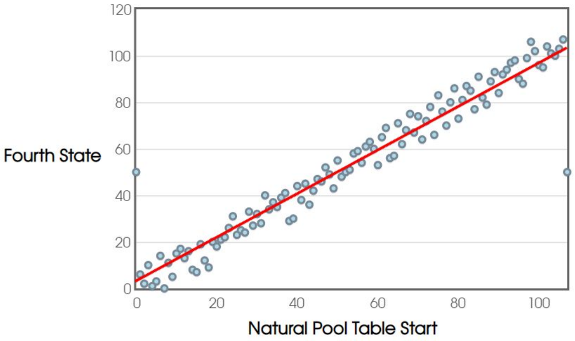
Modulus Nine.

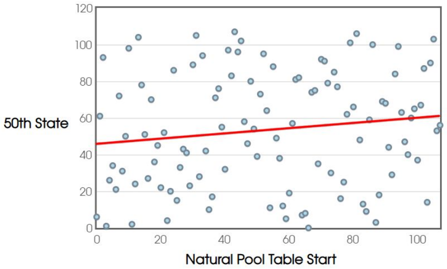




The Fourth State, notice how stable the state remains compared to the original.

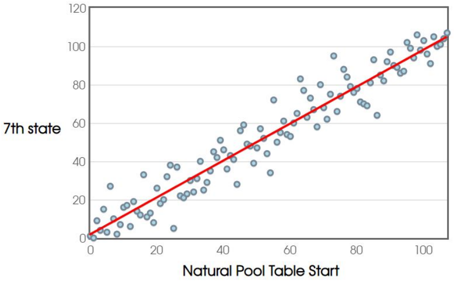
The 47th State (after 47 extra trials, typo in axis label)

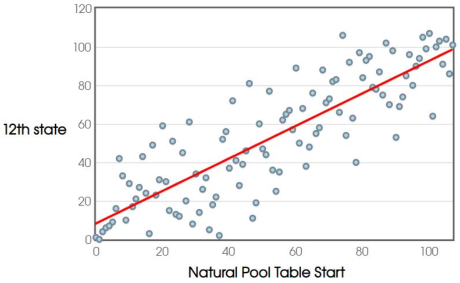




The 7th State: 12th State:

The scatter is now an ellipse, with constant positive curvature; however, in Michigan, their scatter plots take on a form with six changes in the polarity of the curvature...an act that defies the Laws of Nature.



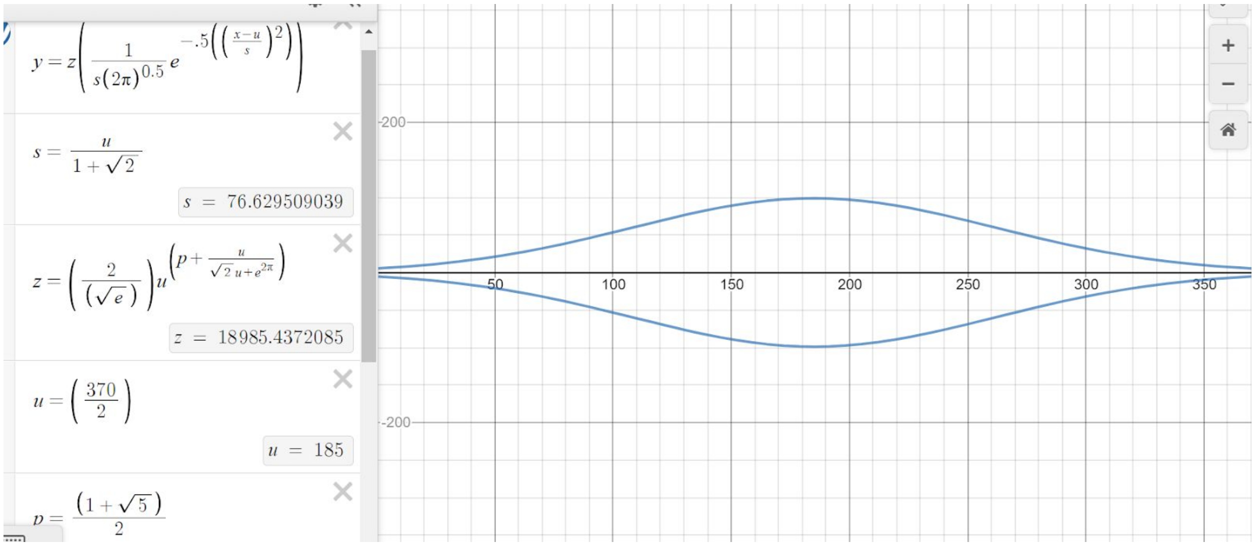


Thus, the engineers are left with a conundrum, if they use pure random number generation, the footprint will immediately assume the 47th State, and be immediately detectable; if they only go through several state changes, an insufficient amount of precincts will swap sides, and if they do too many trials, the pattern will become elliptical and immediately become detectable again (note the pattern in Kalamazoo is elliptical!)

Thus they used a clever mechanism, they altered the boundaries of the pool table itself to fit the following shape (in Macomb $k = 40$ for the below equation, $k = 0$ for Oakland and Kent):

$$y = z\left(\left(\frac{1}{\sigma\sqrt{2\pi}}\right)e^{\left(\left(\left(\frac{x-\mu-k}{\sigma}\right)\right)\right)}\right);\ n = number\ of\ precincts,\ \mu = \frac{n}{2};$$

$$\sigma = \frac{\mu}{1+\sqrt{2}};\ \ = \frac{1+\sqrt{5}}{2};\ z = \left(\frac{2}{\sqrt{e}}\right)\mu^{\left(+\frac{\mu}{\sqrt{2}\mu+e^{2\pi}}\right)}$$



$y = z\left(\frac{1}{s(2\pi)^{0.5}}e^{-.5\left(\left(\frac{x-u}{s}\right)^2\right)}\right)$

$s = \frac{u}{1+\sqrt{2}}$

$s = 76.629509039$

$z = \left(\frac{2}{(\sqrt{e})}\right)u^{\left(p+\frac{u}{\sqrt{2}u+e^{2\pi}}\right)}$

$z = 18985.4372085$

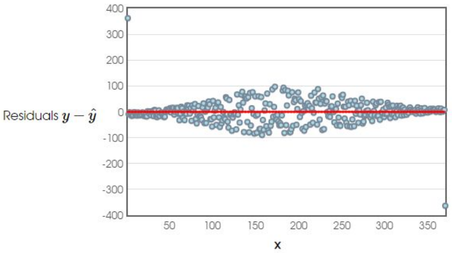
$u = \left(\frac{370}{2}\right)$

$u = 185$

$p = \frac{(1+\sqrt{5})}{2}$

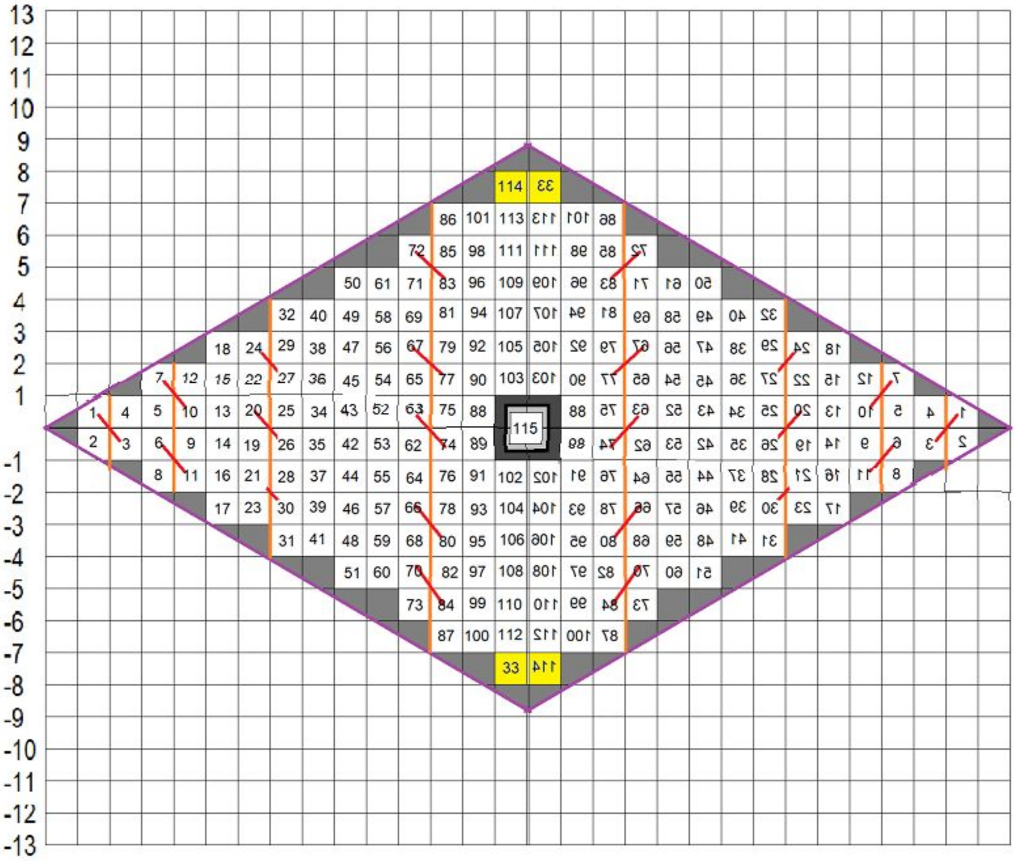
Oakland: https://www.desmos.com/calculator/msicvpvk1u
Kent: https://www.desmos.com/calculator/ivcgmofadc
Macomb: https://www.desmos.com/calculator/lx8jyiq0x6
Kalamazoo: https://www.desmos.com/calculator/qioovq9vk3

They start with all of the billiard balls superimposed at the origin of the table, in the same manner one starts a game of pool with all of the balls packed together. The locations of where the balls stop function as the residual that is added to the line of the linear regression itself. Thus, each residual corresponds to the value added to each actual precinct number along the x-axis, this is the *ideal* transformation from $\mathbf{X''}_j$ to $\mathbf{Y''}_j$.



However, in order to prevent filling out the entire *ideal* shape of the desired residual plot, which would immediately expose itself, while also ensuring a stable bijection between the starting and final positions of precincts, (**X″**, to **Y″**$_j$), they used the algorithm on pages 42-43; below is a drawing of the concept behind that algorithm.

The orange lines represent the Fibonacci gates. These gates can be lifted individually for two partitions (not necessarily adjacent, like in Macomb) to allow for greater movement. The red lines are "Right of Passage" gates that force a handful of precincts to pass through the Fibonacci Gates, in order to obfuscate the hard boundaries of the gates themselves; however, the imprints of these gates are still seen in the actual data, despite this operation.



How the Golden Ratio Manifests in Nature (treehugger.com) .

The golden ratio is quickly converged upon by the quotient of consecutive Fibonacci Numbers. *This choice was also made to give them some sort of plausible deniability and argue that the algorithmic scatter plot is a natural event, especially since they added a binomial weight to the choice of their random number in the algorithm that they used on pages 42-43. The golden ratio also appears in the diagonal sums of binomial coefficients of consecutive degrees.*

Take note that the golden ratio appears, most unnaturally, in the exponent of μ, for the scaling factor of *z*, in the idealized curve.

Section V

Dissecting the Golden Algorithm

Confirming Doctor Shiva

The Algorithm itself uses a total of 362 Billiard Balls, half of which is 181, which is a Centered Square Number: [Centered square number - Wikipedia](), $181 = 9^2 + 10^2$. We will see the number 181 shortly in the derivation of the Golden Algorithm.

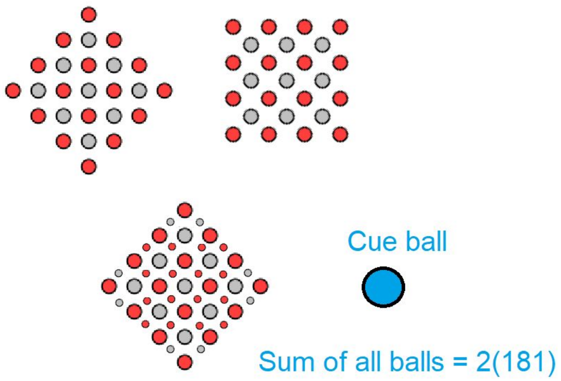
Using a smaller Centered Square Number of 25, which is $3^2 + 4^2$; two sets of 25 balls can be imposed upon each other by reducing the Volume and increasing the density of one set of balls, so that they have the same mass as the first set of balls.

ρ **(rho) = density,** ***m*** **= mass,** ***V*** **= volume**
Volume of a sphere = $\frac{4}{3}\pi r^3$
First set of balls: $\rho_1 = m\ V_1$ **;** $m = \frac{\rho_1}{V_1}$ **.**
First set of balls: $\rho_2 = m\ V_2$ **;** $m = \frac{\rho_2}{V_2}$ **.**

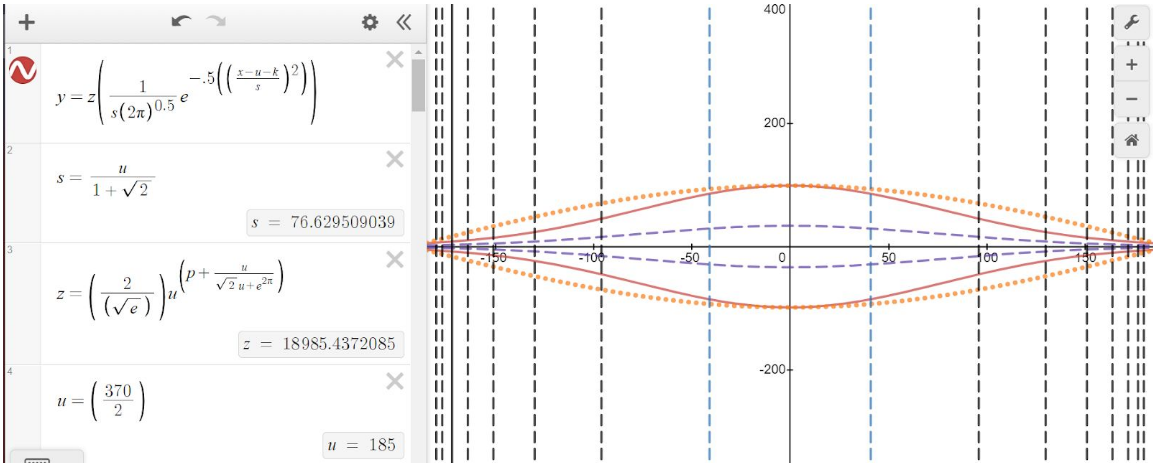
Thus, to retain the same mass for both sets of balls, one must maintain and inversely proportional relationship of the density and volume of either.

The change in the radius for the smaller set of balls is equal to $\sqrt[3]{\frac{\rho_1}{\rho_2}}$. This transformation of the radius for the smaller set of balls, allows two equal sets of balls sharing the same centered square number to be imposed upon each within the same square brace (whereas, an actual game of pool uses a regular three sided polygon as its brace, but the simulation used by the engineers uses a regular four sided polygon, aka a square). Thus, all of the balls start in the center of the table.

When these sets are imposed upon each other, a single ball from the smaller set cannot be included within the brace, this ball functions as the Cue Ball. The Cue Ball is given incredible momentum at the start of the simulation.



We now examine the exact algorithm that was used in Oakland, and perfectly matches the other counties in this study, including Kalamazoo! This is the link to the graph of the Golden Algorithm, below is the image: https://www.desmos.com/calculator/nesjtgvrqd



1 $y=z\left(\frac{1}{s(2\pi)^{0.5}}e^{-.5\left(\left(\frac{x-u-k}{s}\right)^{2}\right)}\right)$

2 $s=\frac{u}{1+\sqrt{2}}$

$s\ =\ 76.629509039$

3 $z=\left(\frac{2}{(\sqrt{e})}\right)u^{\left(p+\frac{u}{\sqrt{2}u+e^{2\pi}}\right)}$

$z\ =\ 18985.4372085$

4 $u=\left(\frac{370}{2}\right)$

$u\ =\ 185$

The engineers start with the red bell curve lines centered at the origin, by subtracting a horizontal translation variable, *k*, from the argument of function, *x*. In this graph, $k = \ -\mu$; however, in Macomb, where a bulge is seen on the right side of the scatter plot, *k* was increased, shifting the entire graph to the right, an act which caused greater mobility of the Pro-Trump precincts when reassigned to the new polynomial.

They then circumscribed two cosine waves about the red bell curves with the following formula:

$$y_1 = \ +\tfrac{z}{w}cos\left(\left(\left(\frac{\sqrt{\pi^{\pi}}}{4\mu}\right)(x-\mu-k)\right)\right);\ w=\left(\tfrac{\mu}{181}\right)e^{(2\ ^{2})};\ z=\left(\tfrac{2}{\sqrt{e}}\right)\mu^{\left(+\ \frac{\mu}{\sqrt{2}\mu+e^{2\pi}}\right)}$$

$$y_2 = \ -\tfrac{z}{w}cos\left(\left(\left(\frac{\sqrt{\pi^{\pi}}}{4\mu}\right)(x-\mu-k)\right)\right);\ w=\left(\tfrac{\mu}{181}\right)e^{(2\ ^{2})};\ z=\left(\tfrac{2}{\sqrt{e}}\right)\mu^{\left(+\ \frac{\mu}{\sqrt{2}\mu+e^{2\pi}}\right)}$$

Where, $\ =\frac{1+\sqrt{5}}{2};\ \mu\ =\frac{n}{2};$ *e is the natural logarithm base, n is the number of precincts seized.*

The equations of the bell curves themselves:

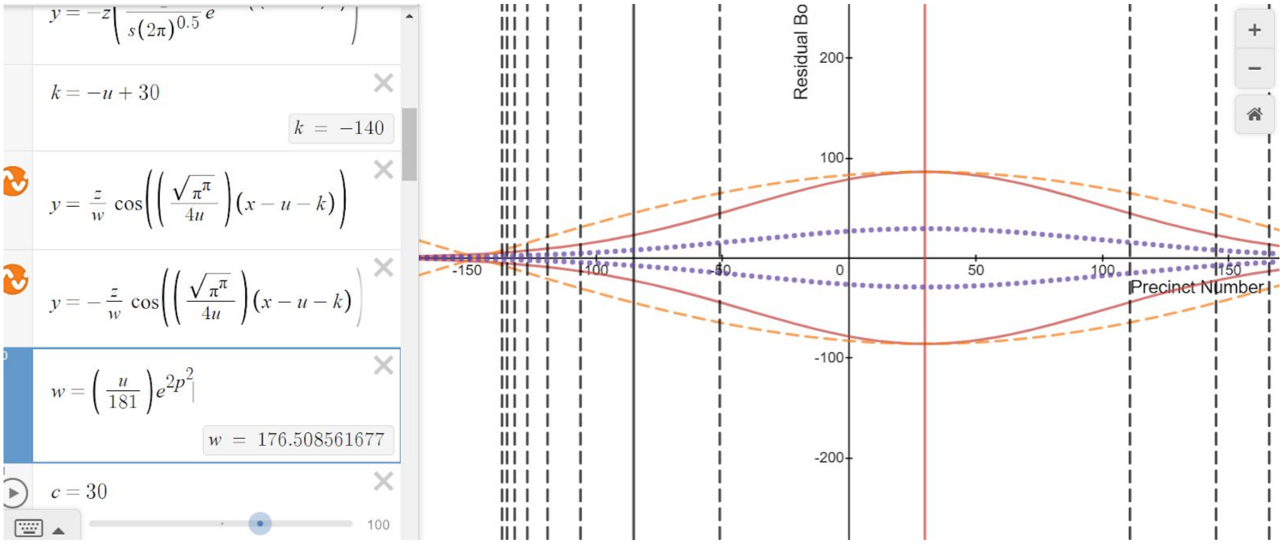
$$y_3 = \ +z\left(\left(\tfrac{1}{\sigma\sqrt{2\pi}}\right)\left(e^{-.5\left(\left(\frac{x-\mu-k}{\sigma}\right)^{2}\right)}\right)\right);\ \mu=\tfrac{n}{2};\ \sigma=\tfrac{\mu}{1+\sqrt{2}};\ z=\left(\tfrac{2}{\sqrt{e}}\right)\mu^{\left(+\ \frac{\mu}{\sqrt{2}\mu+e^{2\pi}}\right)}$$

$$y_4 = \ -z\left(\left(\tfrac{1}{\sigma\sqrt{2\pi}}\right)\left(e^{-.5\left(\left(\frac{x-\mu-k}{\sigma}\right)^{2}\right)}\right)\right);\ \mu=\tfrac{n}{2};\ \sigma=\tfrac{\mu}{1+\sqrt{2}};\ z=\left(\tfrac{2}{\sqrt{e}}\right)\mu^{\left(+\ \frac{\mu}{\sqrt{2}\mu+e^{2\pi}}\right)}$$

The equations of the vertical Fibonacci Gates:

Array $\mathbf{X_{m,2}} = (x_{v,1} = (f_v-\mu)+c;\ x_{v,2} =\ \ (-f_v+\mu)+c;\ \ \forall v,\ 1\le v\le m$, where f_m is the greatest Fibonacci Number less than μ. The variable c allows for dynamic relocation of the gates via a horizontal translation of the entire array, the gates can be individually opened and closed at will to allow for greater movement between two truncated partitions. Those gates are the dotted blacks lines the graph above.

Macomb: https://www.desmos.com/calculator/cbca9tq2um



$y=-z\left(\frac{}{s(2\pi)^{0.5}}e\right)$

$k=-u+30$

$k = -140$

$y=\frac{z}{w}\cos\left(\left(\frac{\sqrt{\pi^{\pi}}}{4u}\right)(x-u-k)\right)$

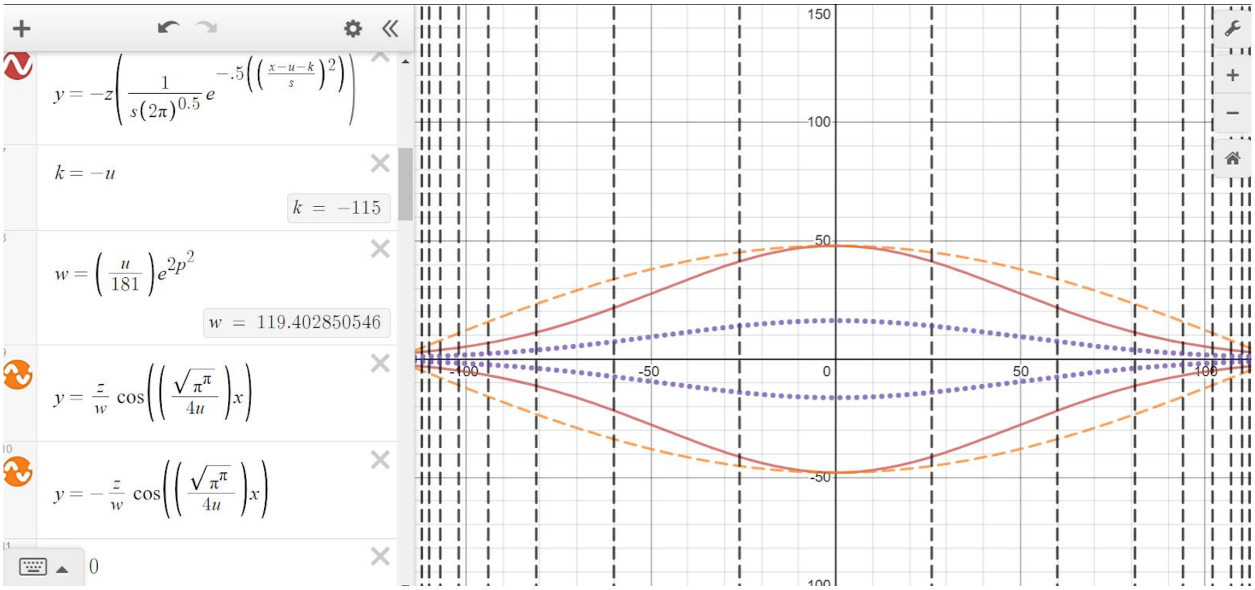
$y=-\frac{z}{w}\cos\left(\left(\frac{\sqrt{\pi^{\pi}}}{4u}\right)(x-u-k)\right)$

$w=\left(\frac{u}{181}\right)e^{2p^{2}}$

$w = 176.508561677$

$c=30$

Kent: https://www.desmos.com/calculator/468qrj3hug



$y=-z\left(\frac{1}{s(2\pi)^{0.5}}e^{-.5\left(\left(\frac{x-u-k}{s}\right)^{2}\right)}\right)$

$k=-u$

$k = -115$

$w=\left(\frac{u}{181}\right)e^{2p^{2}}$

$w = 119.402850546$

$y=\frac{z}{w}\cos\left(\left(\frac{\sqrt{\pi^{\pi}}}{4u}\right)x\right)$

$y=-\frac{z}{w}\cos\left(\left(\frac{\sqrt{\pi^{\pi}}}{4u}\right)x\right)$

0

Let us observe some of the remarkable features of the equations for the bell curve:

$$y_3 = \ +z\left(\left(\frac{1}{\sigma\sqrt{2\pi}}\right)\left(e^{-.5\left(\left(\frac{x-\mu-k}{\sigma}\right)^2\right)}\right)\right);\ \mu = \frac{n}{2};\ \sigma = \frac{\mu}{1+\sqrt{2}};\ z = \left(\frac{2}{\sqrt{e}}\right)\mu^{\left(\ + \frac{\mu}{\sqrt{2}\mu+e^{2\pi}}\right)}$$

We see that the scale of the curve, given by the variable z , has an arbitrary scaling factor itself, that is equal to $\frac{2}{\sqrt{e}}$, there is simply no natural explanation for such a constant, this was manual.

Next we observe the dynamic factor of z is exponential. The base of the exponent is the mean itself, μ , and exponent is the sum of the golden ratio,   , and the quotient of the mean, μ , divided by the square root of two, times the mean, added to $e^{2\pi}$. The presence of $e^{2\pi}$ is alarming, since this is Euler's Formula, immediately implying that a two-dimensional simulation was used using complex numbers to represent the vectors: Euler's formula - Wikipedia

The presence of Euler's Formula then shines further light on the square of two being multiplied against the mean. Here the square of root two is acting as a magnitude of the vectors in the simulation; yet, no sense can be made of the addition of the golden ratio to this exponent, other than it was included arbitrarily.

Now let us examine the standard deviation, $\sigma = \frac{\mu}{1+\sqrt{2}}$. Setting the standard deviation to such an arbitrary quotient of the mean divided by one plus the square root two again signifies direct human intervention in the composition of this algorithm.

Now we examine the circumscribed cosine waves:

$$y_1 = \ + \frac{z}{w}cos\left(\left(\frac{\sqrt{\pi^\pi}}{4\mu}\right)(x-\mu-k)\right);\ w = \left(\frac{\mu}{181}\right)e^{(2^{\ 2}\ )};\ z = \left(\frac{2}{\sqrt{e}}\right)\mu^{\left(\ + \frac{\mu}{\sqrt{2}\mu+e^{2\pi}}\right)}$$

The scaling coefficient of $\frac{\sqrt{\pi^\pi}}{4\mu}$ against the period (argument) of the cosine wave again has no natural explanation. Such an arbitrary formula can only be of direct human intervention in the composition of this algorithm.

Now we examine the scaling divisor of w: Here we see the number 181 directly inserted into the formula, the Centered Square Number of the size of the two sets of Billiard Balls, placed into a square brace at the center of the pool table. Thus, had there been exactly 362 precincts in the County, the quotient of the mean, divided by 181, would have been exactly 1.

Finally we examine the exponential factor that is multiplied against $\frac{\mu}{181}$:

The base of this exponent is e, and the exponent itself is twice the value of the golden ratio squared; this immediately implies that the engineers of this algorithm operated in the fractional base of phi, a base that is commonly used in science: https://en.wikipedia.org/wiki/Golden_ratio_base .

We take note of the Fibonacci Gates, and recall that the quotient of consecutive Fibonacci numbers rapidly converges on the Golden Ratio itself; thus, the engineer (or engineers) had a deep obsession with the golden ratio, and, very much how a serial killer collects trophies and leaves signatures, so did the engineer of this algorithm, by leaving a steady trail of *golden* breadcrumbs.

Now that we are fully armed with all of the information from this investigation, we can now rewrite the Golden Algorithm on pages 42-43, in its full glory, centered at the origin of the Cartesian Plane, in manner that applies uniformly to all counties...without the strange arbitrary scaling values we initially used that were specific to each county.

TRUE Algorithm used to resort the order of the precincts when input into the Polynomial:
The Golden Algorithm

It was noted that Oakland's linear regression (and residual plot) for the positions of the precincts when ordered from least to greatest by their ratio in their Active First Stasis (x-axis) against their new positions when ordered from least to greatest by their ratio in their Active Drop Stasis matched the following algorithm (for which we shall now apply as a test to all of the counties in the dataset):

Let $\boldsymbol{\tau}$ be equal to the number of precincts in $\mathbf{V''}_{\mathbf{j,6}}$, in Oakland $\boldsymbol{\tau} = 380$; $380 = h\text{-}m$.

Let $\boldsymbol{\mu} = \boldsymbol{\tau}/2$.
Let $\sigma = \frac{\mu}{1+\sqrt{2}}$

Let $= \frac{1+\sqrt{5}}{2}$

Let $z = \left(\frac{2}{\sqrt{e}}\right)\mu^{\left(+ \frac{\mu}{\sqrt{2}\mu + e^{2\pi}}\right)}$

Let $w = \left(\frac{\mu}{181}\right)e^{(2^{\ 2})}$

Let k be an arbitrary constant to enact a horizontal translation.

Let c be an arbitrary constant to enact a horizontal translation.

Let $k = \ -\mu + c$

Let $y_1 = \ + \frac{z}{w} cos\left(\left(\frac{\sqrt{\pi^\pi}}{4\mu}\right)(x - \mu - k)\right)$

Let the linear array $\mathbf{F_m}$ be the set of the first m consecutive Fibonacci Numbers, where f_m is the greatest Fibonacci Number less than μ.

Let array $\mathbf{Gate_{m,2}} = (g_{v,1} = (f_v - \mu) + c\ ; \quad g_{v,2} = \ (-f_v + \mu) + c\ ; \ \forall v,\ 1 \le v \le m$.

For each precinct , $\forall\ c_u\ \in \mathbf{C''}_{\mathbf{j}}$, let the linear array $\mathbf{R''}_{\mathbf{j}}$ be the corresponding Residual Array. For each residual , $\forall\ r_u\ \in \mathbf{R''}_{\mathbf{j}}$, set $r_u = \Psi_u$, where $\Psi_u = NORMINV[RAND(), 0,\ (0.341)y_1]$, where the respective value of $\mathbf{X''}_{\mathbf{j}}$ (to the precinct c_u) is injected into the argument of the curve y_1 .

Excel NORMINV Function (excelfunctions.net)
The Excel NORMINV function calculates the inverse of the Cumulative Normal Distribution Function for a supplied value of x, and a supplied distribution mean & standard deviation.

Now, $\forall\ r_u\ \in \mathbf{R''}_{\mathbf{j}}$, round r_u down to the nearest integer (floor), such that r_u is the residual distance from the arbitrary chosen linear regression in the form of $y_5 = \lambda x + b$ will be for the County.

In Oakland, $\lambda = 0.9$ and $b = 19$, were the arbitrary values chosen.

Before we add these residuals to the linear regression, we must first translate all of the residuals horizontally (centered at the origin of the graph) by the mean itself, μ. This is accomplished by simply setting $c = \mu$.

Now we add each $r_u \in \mathbf{R''}_\mathrm{j}$ to each y generated by the line $y_5 = \lambda x + b$, for each $x_u \in \mathbf{X''}_\mathrm{j}$.

To create a bijection between each $x_u \in \mathbf{X''}_\mathrm{j}$ and each $r_u \in \mathbf{R''}_\mathrm{j}$, the algorithm then truncates the indexes of $\mathbf{X''}_\mathrm{j}$ using the values of $\mathbf{Gate}_{\mathbf{m,2}}$.

Reposted:
Let array $\mathbf{Gate}_{\mathbf{m,2}} = (g_{v,1} = (f_v - \mu) + c\ ; \quad g_{v,2} = \ (-f_v + \mu) + c\ ;\ \ \forall v,\ 1 \le v \le m$.

Quarantine the indices of $\mathbf{X''}_\mathrm{j}$ in the following order:

Let $\mathbf{X}_{\mathbf{1,1}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{1,1} \le x_\mathrm{k} \le -1 + g_{2,1}$

Let $\mathbf{X}_{\mathbf{2\,,1}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{2,1} \le x_\mathrm{k} \le -1 + g_{3,1}$

Let $\mathbf{X}_{\mathbf{3,1}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{3,1} \le x_\mathrm{k} \le -1 + g_{4,1}$

Let $\mathbf{X}_{\mathbf{4,1}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{4,1} \le x_\mathrm{k} \le -1 + g_{5,1}$

Let $\mathbf{X}_{\mathbf{5,1}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{5,1} \le x_\mathrm{k} \le -1 + g_{6,1}$

...

Let $\mathbf{X}_{\mathbf{v,1}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{v,1} \le x_\mathrm{k} \le -1 + g_{v+1,1}$

...

Let $\mathbf{X}_{\mathbf{m\text{-}1,1}} \subset \mathbf{X''}_\mathrm{j}\ \forall x,\ g_{m-1,1} \le x_\mathrm{k} \le -1 + g_{m,1}$

Then:

Let $\mathbf{X}_{\mathbf{1,2}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{1,2} \ge x_\mathrm{k} \ge +1 + g_{2,2}$

Let $\mathbf{X}_{\mathbf{2\,,2}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{2,2} \ge x_\mathrm{k} \ge +1 + g_{3,2}$

Let $\mathbf{X}_{\mathbf{3,2}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{3,2} \ge x_\mathrm{k} \ge +1 + g_{4,2}$

Let $\mathbf{X}_{\mathbf{4,2}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{4,2} \ge x_\mathrm{k} \ge +1 + g_{5,2}$

Let $\mathbf{X}_{\mathbf{5,2}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{5,2} \ge x_\mathrm{k} \ge +1 + g_{6,2}$

...

Let $\mathbf{X}_{\mathbf{v,2}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{v,2} \ge x_\mathrm{k} \ge +1 + g_{v+1,2}$

...

Let $\mathbf{X}_{\mathbf{m\text{-}1,2}} \subset \mathbf{X''}_\mathrm{j}\ \forall x,\ g_{m-1,2} \ge x_\mathrm{k} \ge +1 + g_{m,2}$

Finally let:

Let $\mathbf{X}_{\mathbf{2m\text{-}1}} \subset \mathbf{X''}_\mathrm{j}\ \ \forall x,\ g_{m,1} \le x_\mathrm{k} \le g_{m,2}$

Now for each $x_u \in \mathbf{X}_{v,\, i}$, generate a local ξ in accordance to the arbitrary linear regression of the county in the form of $\xi = \lambda x + b$. Place each value of ξ_u into the new array $\mathbf{Z}_{v,\, i}$.

Now add the respective residual, r_u , to each respective element of $\mathbf{Z}_{v,\, i}$, ξ_u .

Now for each quarinted partition, $\mathbf{X}_{v,\, i}$, resort the partition from least to greatest by $\mathbf{Z}_{v,\, i}$. This will upset the order of $\mathbf{X''}_{j}$, as intended by the algorithm.

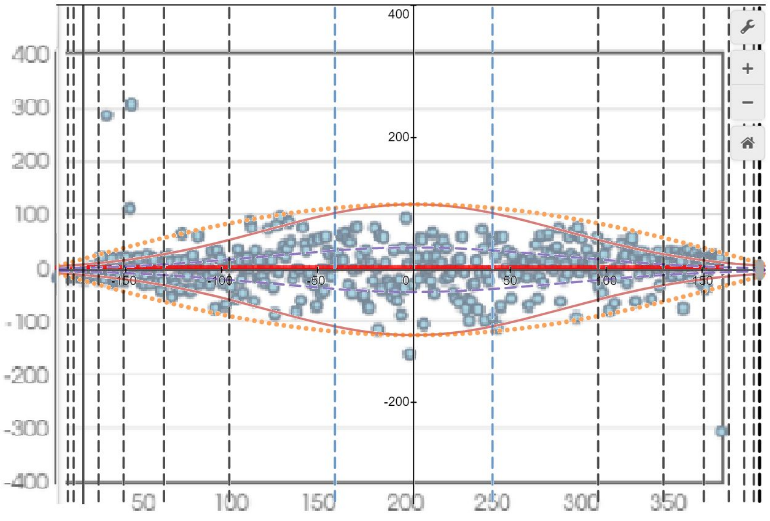
Now recombine all of the quarinted partitions back into the array $\mathbf{V''}_{j,\, 6}$.

Now set each $y \in \mathbf{Y''}_{j}$ equal to each $x \in \mathbf{X''}_{j}$; **AFTER** this, resort only $\mathbf{X''}_{j}$ from least to greatest. The map from $\mathbf{X''}_{j}$ to $\mathbf{Y''}_{j}$ is how the precincts in their starting order, by ratio, from least to greatest in respect to $\mathbf{X''}_{j}$, are assigned their input value into the new polynomial. In words, this is the bijective map between the domains of both polynomials.

At this point you can implement *Right of Passage Gates* as seen on page 69 if you wish (the red diagonal tick marks to exchange precincts on either side of a Fibonacci gate; such *Right of Passage Gates* can also be used to create a handful of outliers by exchanging two precincts from two non-adjacent quarantine zones).

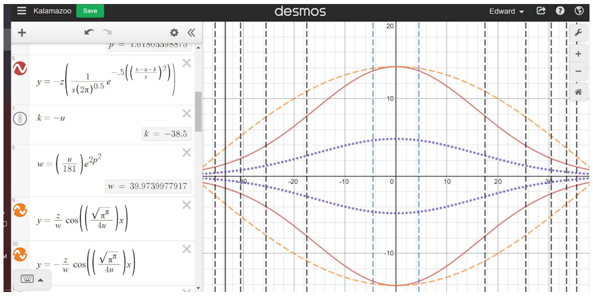
Now set each $g \in \mathbf{V''}_{j,\, 6}$ to $g_i = (a_{10})y^{10}+(a_9)y^9+(a_8)y^8+...(a_1)y+(a_0)$
The values in $\mathbf{G''}_{j,3}$ are Trump's new percentages after the application of the decic (quintic) polynomial.

Oakland's actual residual plot of its precinct movement between polynomials imposed upon the first graph:



:

Kalamazoo: https://www.desmos.com/calculator/twdltuyf66



$p = 1.61803398875$

6 $y = -z\left(\frac{1}{s(2\pi)^{0.5}}e^{-.5\left(\left(\frac{x-u-k}{s}\right)^2\right)}\right)$

7 $k = -u$

$k = -38.5$

8 $w = \left(\frac{u}{181}\right)e^{2p^2}$

$w = 39.9739977917$

9 $y = \frac{z}{w}\cos\left(\left(\frac{\sqrt{\pi^\pi}}{4u}\right)x\right)$

10 $y = -\frac{z}{w}\cos\left(\left(\frac{\sqrt{\pi^\pi}}{4u}\right)x\right)$

Section VI

Spreadsheets and Data Used

Original Data: bigdaddyMI.csv; mi_detroit_timeseries.csv
Fathersheet: 004bigdaddyMIComplete.ods; log of evolutions BigDaddyFathersheetReadme.odt
Mothersheet: 003EvoMothersheetCOMPLETE.ods ; log of evolutions MothersheetLog.odt
Logicboard: LogicBoardTemplate.ods; log of the Logicboard evolutions LogicBoard Evolution Log.odt

Data Spreadsheets from the LogicBoard Template:
010EvoQuinticRegression.ods
011EvoQuinticRegression.ods
011KalamazooActiveRegression.ods
011KentActiveRegression.ods
011MacombActiveRegression.ods

RandomNumberRegressionsExperiment.ods
VotingSimulationLargestDiff(delta).ods
OaklandPercentageDropsQuinticRegressions.ods
NaturalTemplate.ods
Revised Pool Table Experiment.ods (all 47 trials)
Odin's Table.ods (complex numbers could not be calculated in Librebase to finish this experiment).