Statistics and Color: How Color Schemes can be Statistically Configured Based Off Images

Sam C. Lee

Abstract: Color schemes can be the defining factor when it comes to the success of web design and other graphical displays. Color schemes are chosen mainly based on impression and color theory. Certain colors accent and complement each other, while other colors clash together. This study applies statistics to yield working color schemes. With any image, statistics can be used to approximate the main color values of the respective picture. With these statistical generalizations, certain colors can be artificially selected to yield congruent color schemes. The statistical process of yielding these color schemes can be compiled into a portable API: web designers can accessibly use this algorithm when using any image to find a favorable color scheme. The theory of this statistical algorithm relies primarily on two concepts: colors that are similar work well together, but at the same time, however, colors need to be distinguishable from the other colors in the set. This algorithm partitions a given image into cells which are then aggregated into one color. The compiled color of each cell is then compiled into a comprehendible programmatic dictionary. The algorithm randomly selects a color in the compiled dictionary which is then analyzed to see if that specific color would work in the color scheme. Random selection is implemented to achieve color diversity with each generation. This study found that not every image can be generalized in the same way; thus, the algorithm requires certain optimized controls to tailor to the respective image.

Introduction: Little research has been conducted with respect to statistical interpretations of color theory. While extensive research has been done with regard to the implications of color theory, the topic has never been explored thoroughly with statistical applications. Other applications that perform similar analyses, such as Canva and Palette Generator do not offer flexibility with their generations, and moreover, provide no documentation of their algorithm. This study addresses how colors can be extracted from images and can be compiled into a synergetic color set. Moreover, this study optimizes the control variables that are adjusted per each image given. This study explores how good color schemes can be first and foremost representative of the respective image, and secondly, how these palettes can be approximated with statistics.

Methodology: In order for any kind of data analysis to take place, the data has to be normalized. In the context of this study, the data is simply the image presented. The data is normalized through the following steps:

1) All images are simplified into a 480-by-480-pixel square image through scaling, resulting in 230,400 individual colored pixels.
2) The resulting square image is then divided into square cells based on a controlled cell size that will be optimized later in the study.

3) Each square cell is then compiled into an aggregate color. Essentially, the average color of all the pixels in the cell is compiled and then returned. Every colored pixel can be represented through values of red, green, and blue: each color is given a value 0 through 255, where the combinations of these three colors can yield any color visible to the eye. This is represented in the format *RGB([red], [green], [blue])*.

   The average color is found in this way, where *n* represents the total number of pixels in a cell, and *i* represents each pixel:

   a) $r = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(red_i)^2}$

   b) $g = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(green_i)^2}$

   c) $b = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(blue_i)^2}$

   d) returned: RGB(r, g, b)

4) Each color placement value, ranging from 0 to 255, is compiled in the format *RGB(r,g,b)*. This color is then displayed on a grid for visual purposes.

   Example (cell size = 48x48; $\frac{480 \cdot 480}{48 \cdot 48} = 100$, and thus a 10x10 grid):

5) Once the grid cells have been defined, each compiled color value of each individual cell is then congregated into a programmatic JavaScript dictionary. Each color is translated into a *score* value in order to normalize the colors into a universal comparative statistic. The *score* represents how much pigment an *RGB*-value has.

   Scores are calculated in the following way for each cell: $\sqrt{red^2 + green^2 + blue^2}$.

   Because each individual color value in the *RGB* format ranges from 0 to 255, a *score* can range from 0 to $255\sqrt{3}$:

   a) *minimum* : $\sqrt{0^2 + 0^2 + 0^2} = 0$

   b) *maximum* : $\sqrt{255^2 + 255^2 + 255^2} = 255\sqrt{3}$

   A *score* of 0 would represent absolute black, while a *score* of $255\sqrt{3}$ would represent absolute white.

Once the data is sorted in this matter, the color scheme selection can begin. Because this method is flexible, a variety of color schemes can be generated. For the purposes of this study, a dark-themed color scheme and a light-themed color scheme are generated. Each color scheme is generated through these steps:

1) An *anchor* is chosen by selecting a color at the *n*th percentile of all the scores in the dictionary. The percentile *n* is dependent on what kind of color scheme being generated. For the dark and light color schemes, $n = 15$ and $n = 85$, respectively, for the entirety of the study. These *n*-values can be adjusted to accommodate different color schemes. For an all-around color scheme, perhaps an *n* of 50 would be appropriate. Randomness is implemented here to accomplish color diversity. The randomness is incorporated programmatically such that each *anchor* percentile generated typically deviates 11.5 from the mean of *n* ($\sigma_n = 11.5$).

   Programmatically, the *anchor* for each color scheme is obtained from the *scores* dictionary (represented in this code snippet with the *sorted* keyword):

   a)
```
var darkAnchor =
sorted[Math.max(Math.round((0.15+((0.5-Math.random())/2.5))*sorte
d.length), 0)];
```

   b)
```
var lightAnchor =
sorted[Math.min(Math.round((0.85+((0.5-Math.random())/2.5))*sorte
d.length),sorted.length-1)];
```

   The theory behind this technique is that colors that are similar go well together. An *anchor* is defined such that all colors selected in the future for the respective color scheme can be compared to this *anchor* score.

2) Once the *anchors* have been defined, the principle of random selection comes in to play.

Each color scheme generated can contain up to five different colors. The number of colors generated depends on how colorfully diverse each respective image is. A new color is added to a color scheme through the following method:

a) A color is randomly selected from the grid that contains all cells and their respective *scores*.
b) The *score* of the selected color is then compared to the respective *anchor* value depending on what color scheme is currently being generated. If the selected *score* is within two standard deviations from the *anchor* score, the color proceeds to the next step. The standard deviation is defined as the typical distance a randomly selected *score* is from the mean *score*. If the prospective color fails this test, the program chooses another random color from the grid.
c) If the *score* is within two standard deviations from the *anchor*, the *score* is then checked for diversity. In order to validate color diversity, colors should not be too similar to each other. This is done by comparing the selected color *score* to each color *score* already in the respective color scheme. For each generation, an optimized bin size is generated. This bin size determines the minimum distance a selected *score* can be away from all other color *scores*. The bin size is the partition in which the data is grouped, equivalent to a bin size in a histogram. The bin size is optimized such that the selected colors will have appropriate comparative measurements with respect to the image as a whole. The process of bin size optimization is discussed later in this study. If the distance the selected color *score* is greater than a bin size away from every other color *score* already in the color scheme, the *score* passes the test, and the corresponding color is appended.
d) Steps (a) through (c) are repeated for each color scheme with the only difference being the *anchor score* reference.

The process of generating a color scheme can be optimized through the modification of specific variables. The two variables that this study focuses on is bin size optimization and cell size optimization.

Bin Size Optimization: The implementation and calculation of the ideal bin size ultimately used in this study come from research conducted by Hideaki Shimazaki and Shigeru Shinomoto (Shimazaki and Shinomoto, Neural Comput 19 1503-1527, 2007). According to this research, a bin size that minimizes the Mean Integrated Squared Error (MISE) is the one that is the most optimal. This is done with the following process:

1) The dataset is divided into N bins of width Δ.
2) The number of events k in each *ith* bin can be represented by k_i; the mean and variance of the number of events in each bin can be calculated as followed:

a) $\mu_k = \frac{1}{N}\sum_{i=1}^{N} k_i$

b) $\sigma_k{}^2 = \frac{1}{N}\sum_{i=1}^{N}(k_i - \mu_k)^2$

3) The mean and variance of the number of events in each bin are then compiled into the cost function $C(\Delta) = \frac{2\mu_k - \sigma_k{}^2}{\Delta^2}$.
4) The bin size Δ that minimizes $C(\Delta)$ is the one that is optimized.

In the context of this study, this process of finding the optimal bin size is determined by taking the dictionary of *scores* as the dataset. Foreach image uploaded, the process described above is repeated hundreds of times to minimize $C(\Delta)$. Programmatically, this is done by changing the bin size Δ by a consistent interval and repeating the optimization process. For the purposes of this study, the step interval is 0.01 in order to optimize computation time. The statistical algorithm begins at a small bin size and increments up by this step interval, logging the cost of each bin size in the process. The algorithm then defines the bin size used throughout the rest of the program by finding the smallest cost in the list of the ones recorded.

In order to determine what makes a good color scheme, there had to be a definition of what "good" actually meant. For the purposes of this study, it was determined that a diverse color scheme is a good one. By measuring the standard deviations of the *scores* of each color scheme generated, the *benefit* of each cell size used can be used. Because a lower cell size leads to more partitions, the computational time increases with respect to the number of squares or partitions used. As a result, the *benefit* of each cell size tested can be determined by the diversity—or the deviation—of the color schemes with respect to computational time required, which can be represented by $\frac{(s_L+s_D)^2}{t}$, where s_L is the sample standard deviation of the *scores* of the light color scheme, s_D is the sample standard deviation of the *scores* of the dark color scheme, and t is the computation time (milliseconds). The sum of the sample standard deviations is squared in order to allocate more weight to that part of the statistic. These statistics are generated programmatically after each generation and are recorded.

Results:

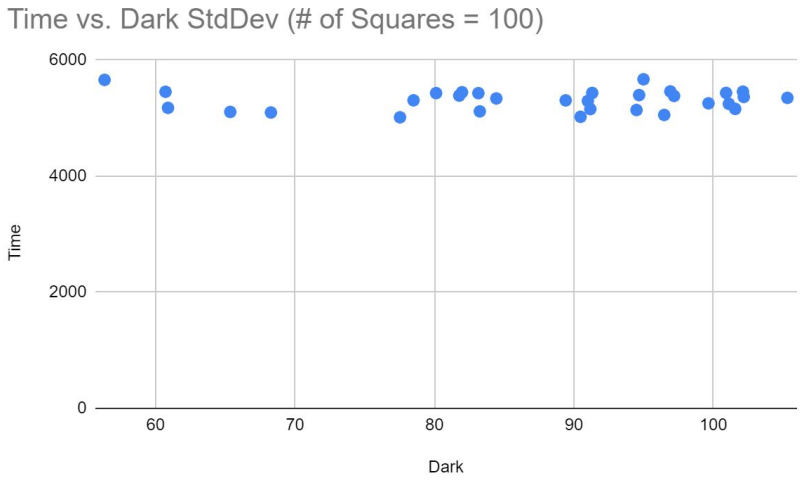
For every trial, statistics were returned for s_L, s_D, and t. These statistics are used to calculate the *benefit* $\frac{(s_L+s_D)^2}{t}$ of the dependent variable, the cell size, or the number of cells that partition the respective image. This study used the average s_L, s_D, and t of all trials to calculate the final *benefit* as a single reported statistic as opposed to taking the average *benefit* of all trials. This was done because the computational time *t* was found to be independent of the other statistics.

This study analyzed four images, measuring the *benefit* with respect to the cell sizes: 120, 96, 80, 60, 48, 40, 32, and 24, translating into 4x4, 5x5, 6x6, 8x8, 10x10, 12x12, 15x15, and 20x20 grids. For each cell size tested, $n = 30$ trials were used in order to satisfy the Central Limit Theorem.

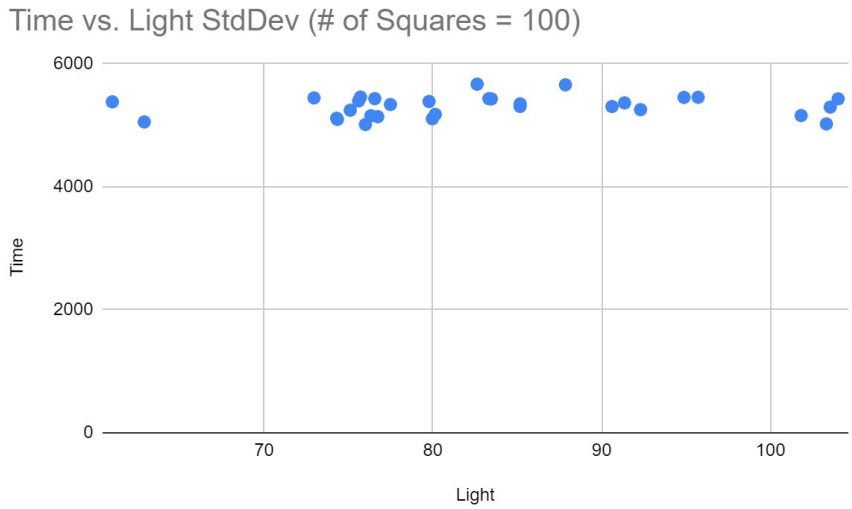
Image #1: Great Wall of China

Shown below is the relationship between the $(s_L + s_D)$ and t statistics for controlled cell size (48; 10x10 grid) and $n = 30$ trials.

Correlation between the sample standard deviation of the *scores* of the dark color scheme and the resulting computational time: 0.0152124019

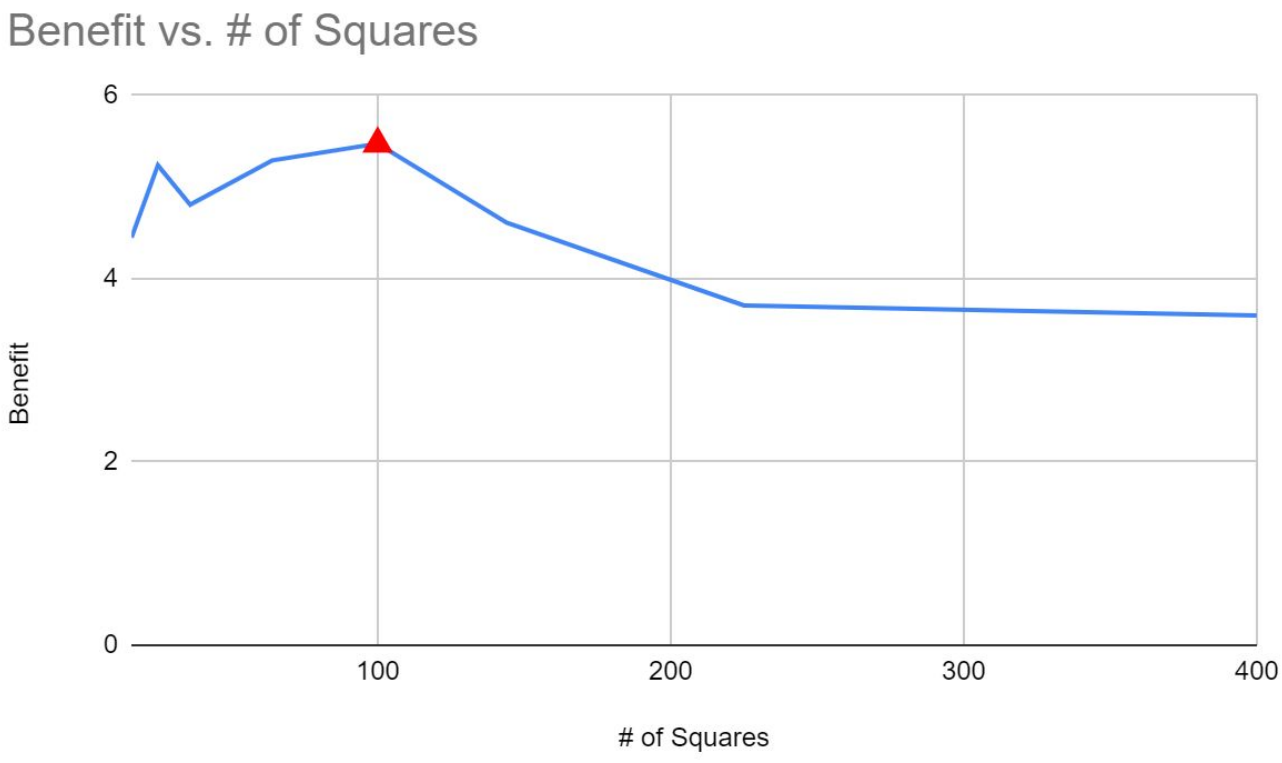


Correlation between the sample standard deviation of the *scores* of the light color scheme and the resulting computational time: 0.126711319



Because the correlation is weak, there is evidence to suggest that computational time t and $(s_L + s_D)$ are independent. Therefore, the *benefit* that will be analyzed as the dependent variable will be $\frac{(\bar{x}_{s_l}+\bar{x}_{s_D})^2}{\bar{x}_t}$.

Using this statistic as the dependent variable, this image as a control, and using the cell size or number of squares as the independent variable, the cell size can be optimized with n = 30 trials.



The red triangle indicates the cell size that optimizes the *benefit*, for this image, that cell size is 48, or a 10x10 grid, where the # of partitions/squares = 100.

Using this optimized cell size and optimized bin size, an ideal dark-themed and an ideal light-themed color scheme can be generated from this image:

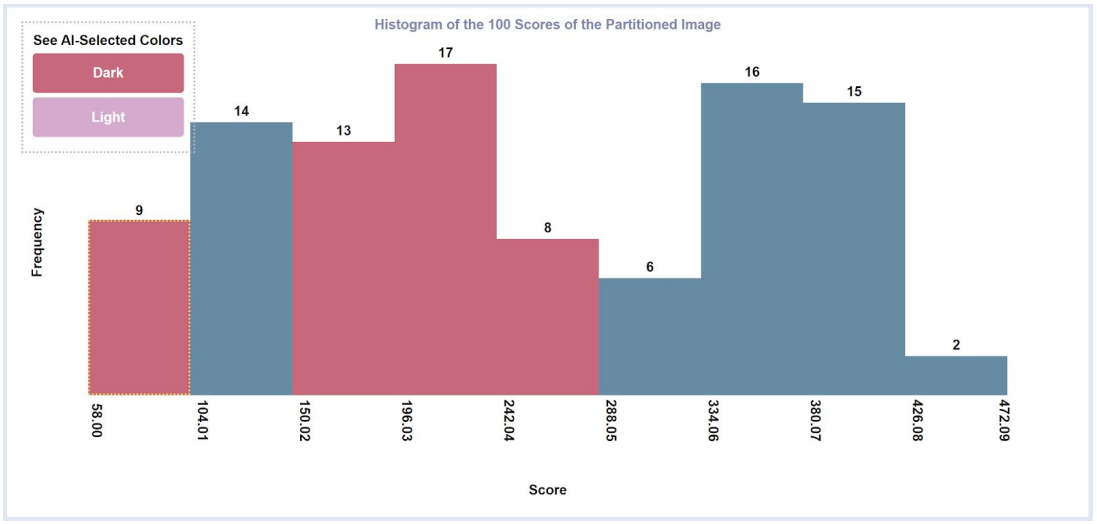
Dark-themed Color Scheme:

#363D18

#6A7142

#848D68

#8AACA8



Light-themed Color Scheme:

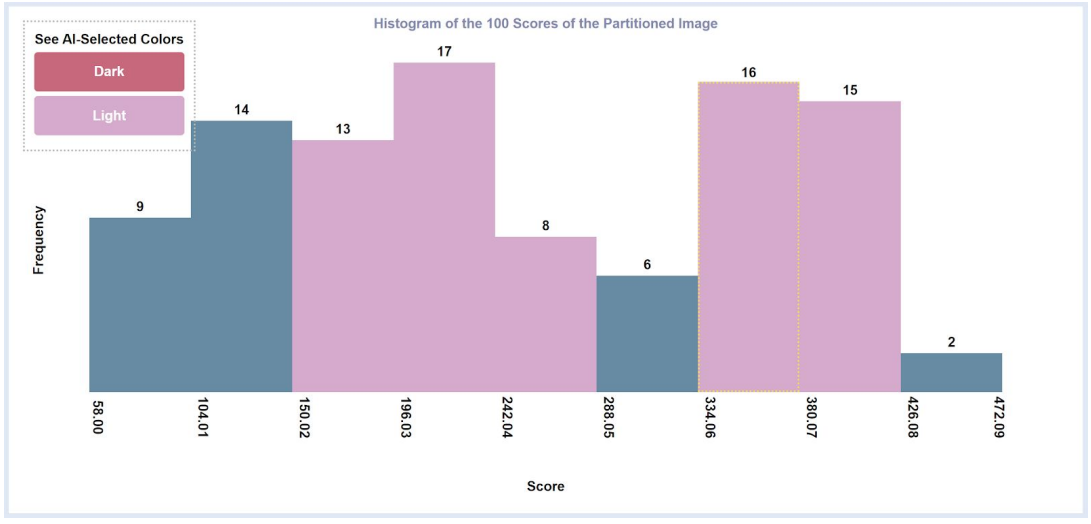
#75683E

#73927B

#93AFAF

#B4CAE0

#EFF4FA

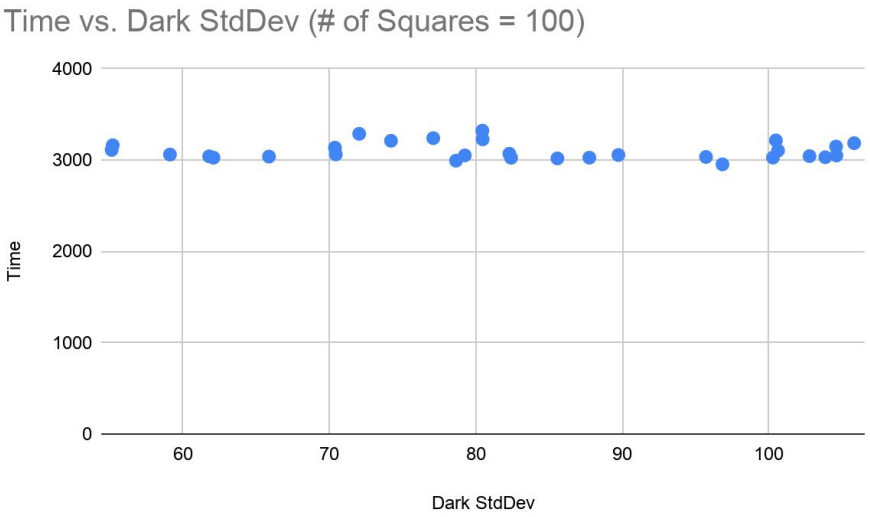


The above color schemes are summarized in the respective histograms. The histograms chart all the cellularized *scores* from the partitioned image with the optimized bin size. The highlighted bins represent where each color in the presented color scheme is drawn from; the dotted gold outline on the particular bin represents the bin from which the anchor *score* was chosen.

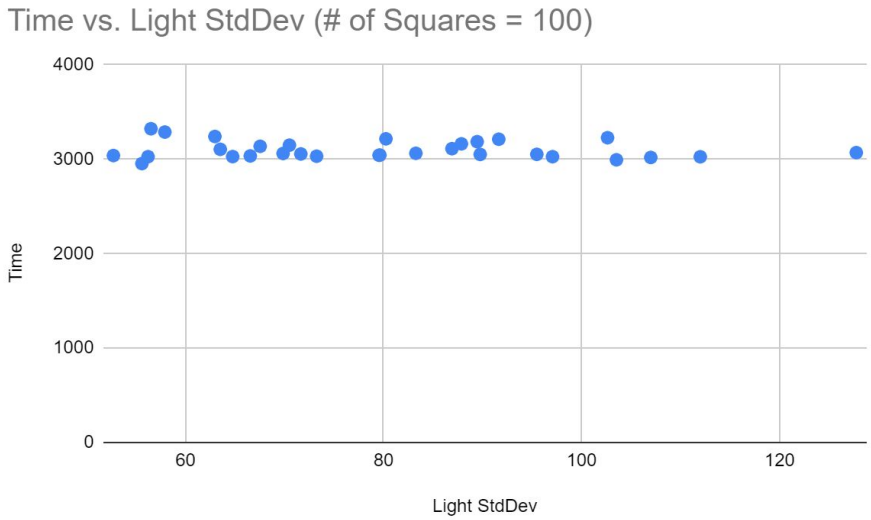
Image #2: Tulane University

Shown below is the relationship between the $(s_L + s_D)$ and t statistics for controlled cell size (48; 10x10 grid) and $n = 30$ trials.

Correlation between the sample standard deviation of the *scores* of the dark color scheme and the resulting computational time: -0.1104518983



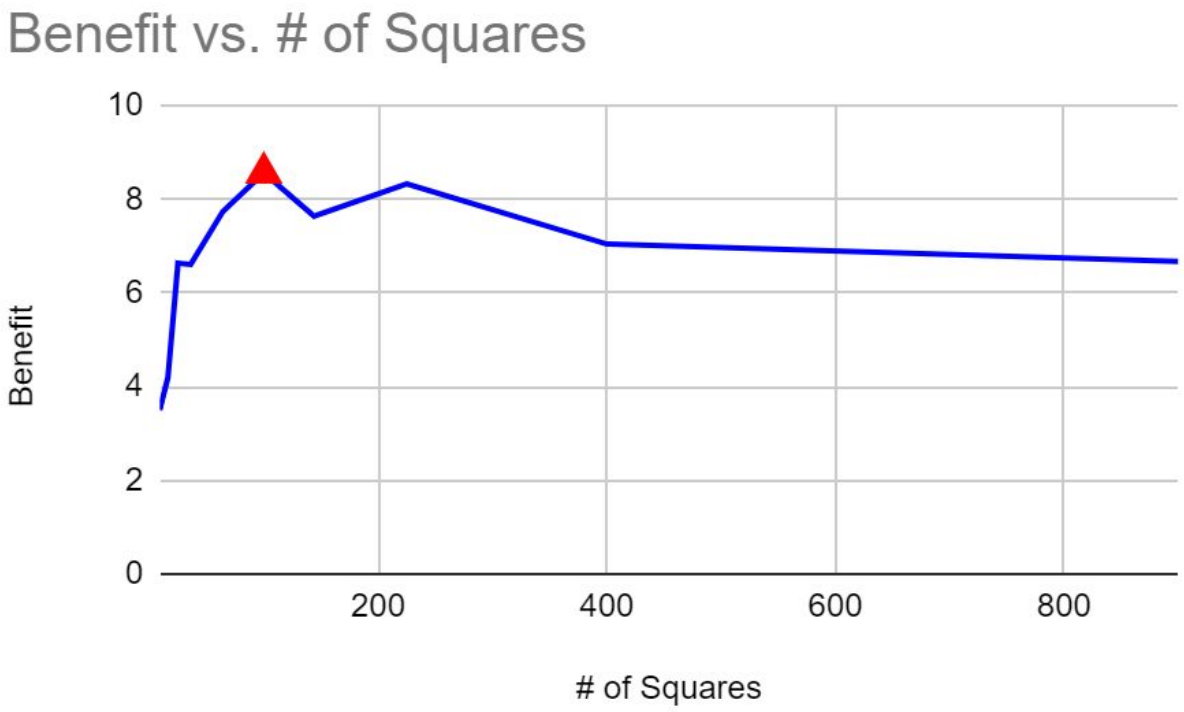
Correlation between the sample standard deviation of the *scores* of the light color scheme and the resulting computational time: -0.1564986414



Because the correlation is weak, there is evidence to suggest that computational time t and $(s_L + s_D)$ are independent. Therefore, the *benefit* that will be analyzed as the dependent variable will be $\frac{(\bar{x}_{s_l}+\bar{x}_{s_D})^2}{\bar{x}_t}$.

Using this statistic as the dependent variable, this image as a control, and using the cell size as

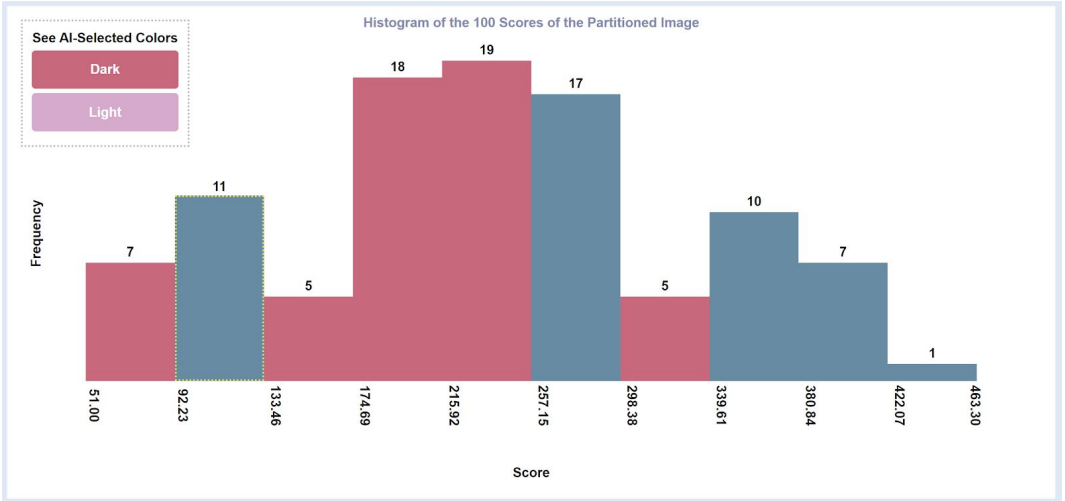
the independent variable, the cell size can be optimized with $n = 30$ trials.



The red triangle indicates the cell size that optimizes the *benefit*, for this image, that cell size is 48, or a 10x10 grid, where the # of partitions/squares = 100.

Using this optimized cell size and optimized bin size, an ideal dark-themed and an ideal light-themed color scheme can be generated from this image:

Dark-themed Color Scheme:



Light-themed Color Scheme:

#496162

#6B757F

#969B9E

#C3B5A9

#E2E9EB

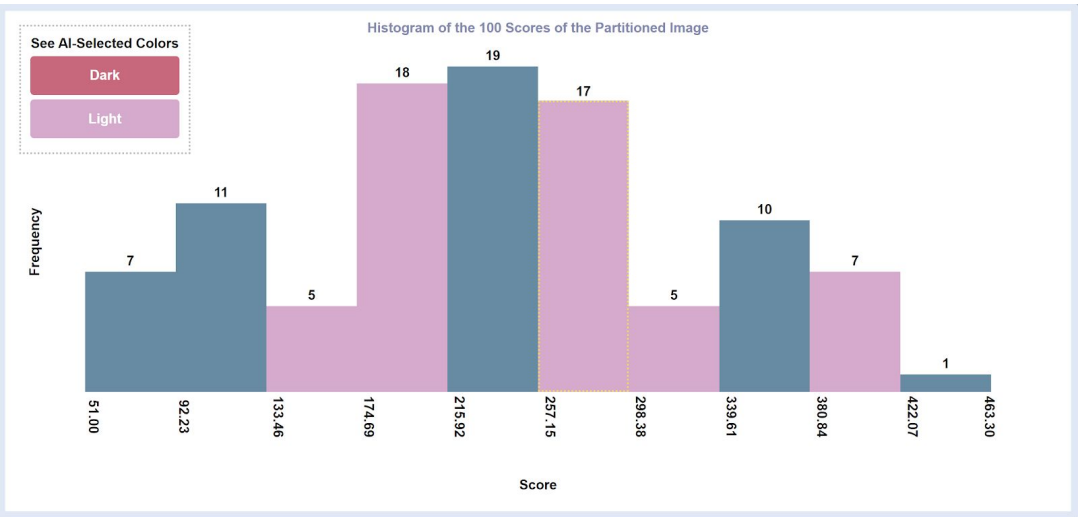
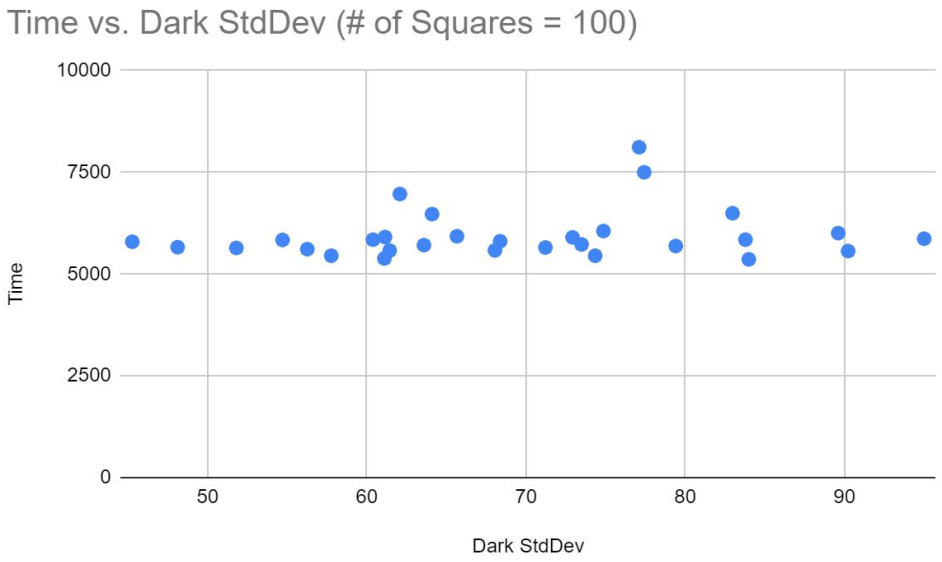


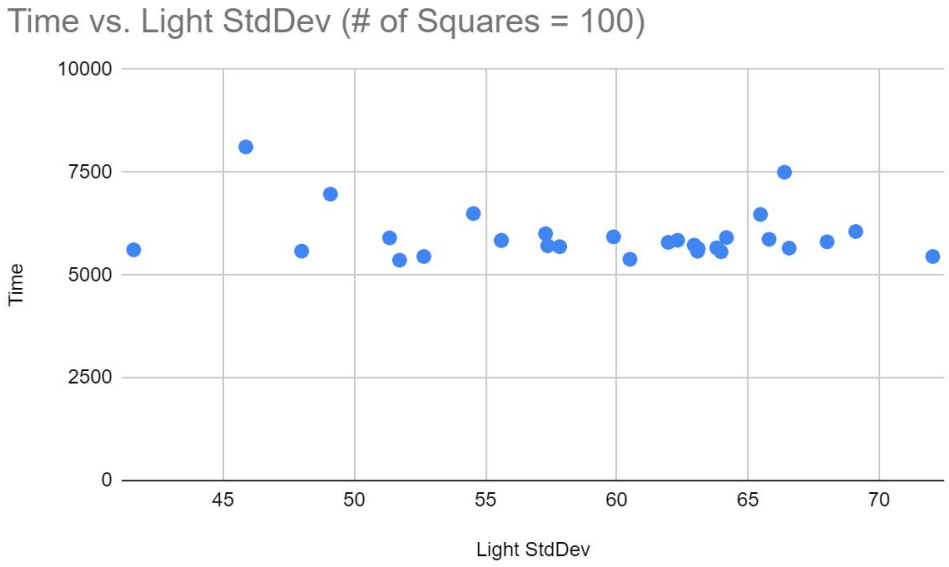
Image #3: Polk City Park

Shown below is the relationship between the $(s_L + s_D)$ and t statistics for controlled cell size (48; 10x10 grid) and $n = 30$ trials.

Correlation between the sample standard deviation of the *scores* of the dark color scheme and the resulting computational time: 0.1739673911

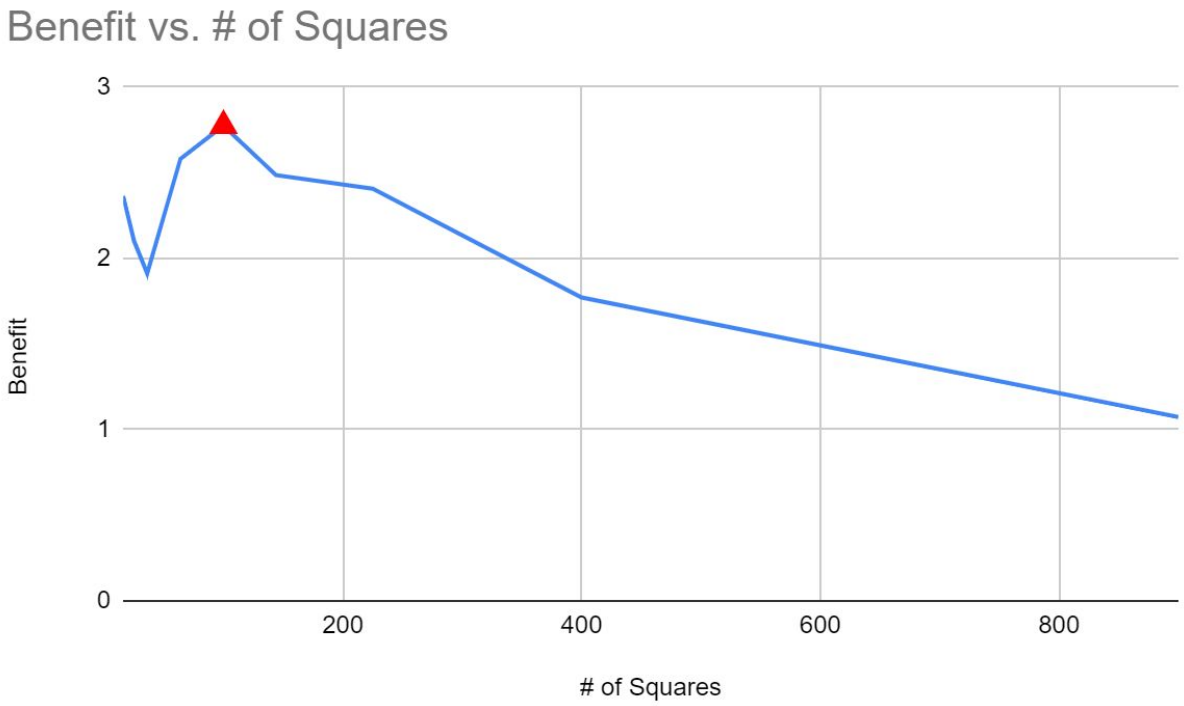


Correlation between the sample standard deviation of the *scores* of the light color scheme and the resulting computational time: -0.1920036585



Because the correlation is weak, there is evidence to suggest that computational time t and $(s_L + s_D)$ are independent. Therefore, the *benefit* that will be analyzed as the dependent variable will be $\frac{(\bar{x}_{s_l}+\bar{x}_{s_D})^2}{\bar{x}_t}$.

Using this statistic as the dependent variable, this image as a control, and using the cell size as the independent variable, the cell size can be optimized with n = 30 trials.

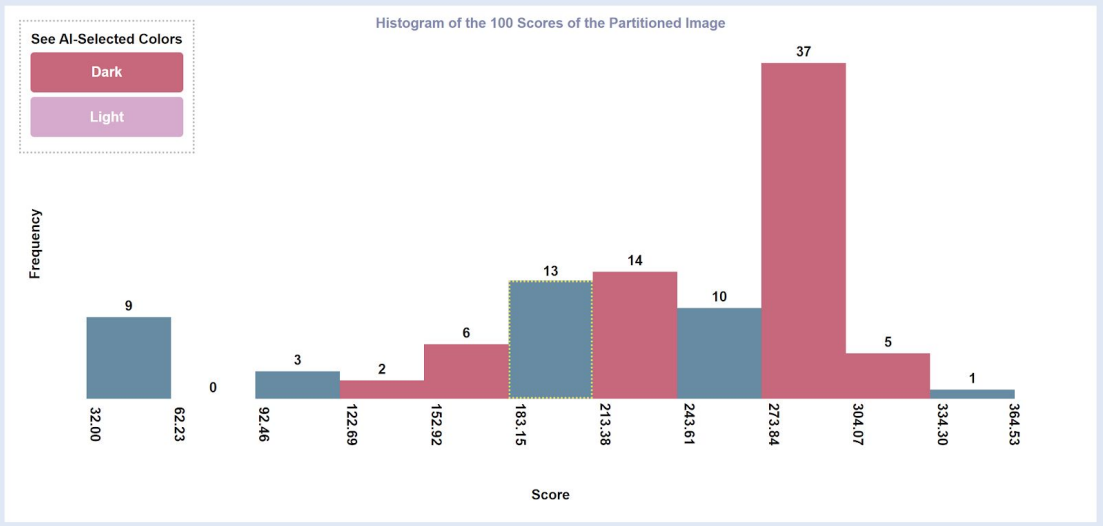


The red triangle indicates the cell size that optimizes the *benefit*, for this image, that cell size is 48, or a 10x10 grid, where the # of partitions/squares = 100.

Using this optimized cell size and optimized bin size, an ideal dark-themed and an ideal light-themed color scheme can be generated from this image:

Dark-themed Color Scheme:

#5E5131
#6F605F
#A97747
#739CC5
#CBBFB0



Light-themed Color Scheme:

#5B6B8E
#A07586
#66AAD2
#C8BBAF

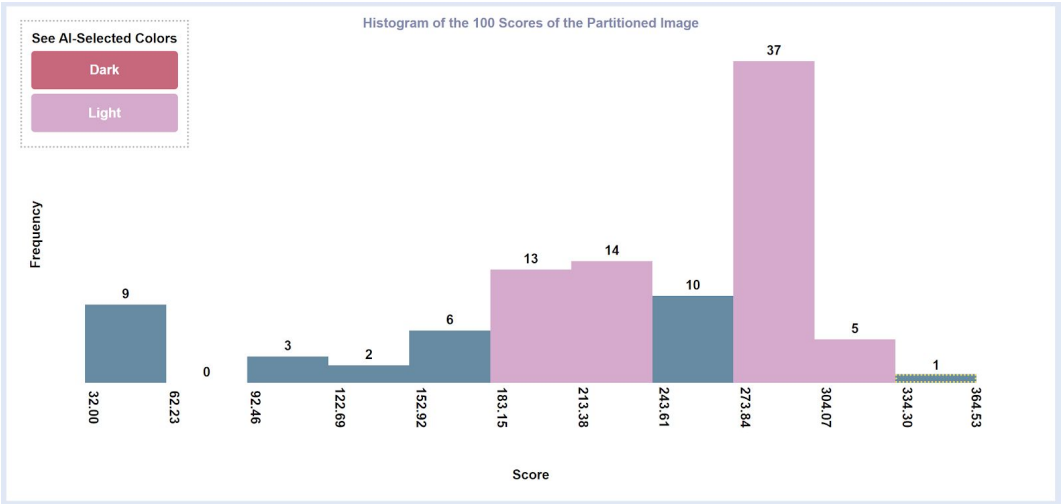
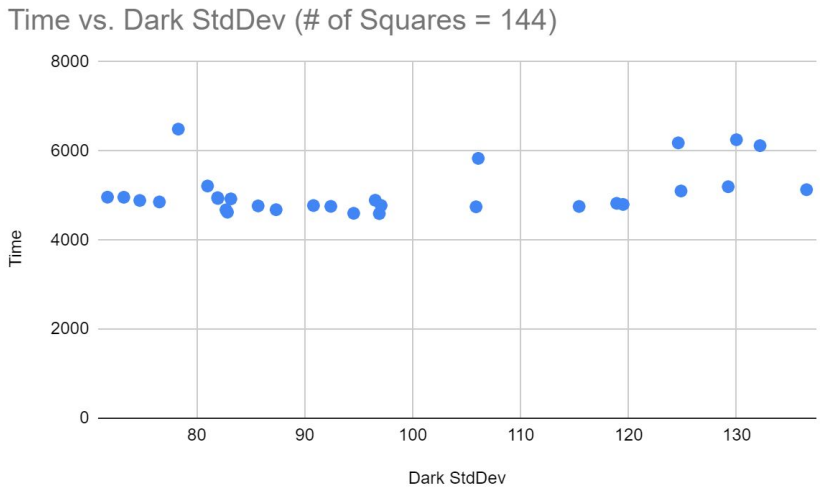


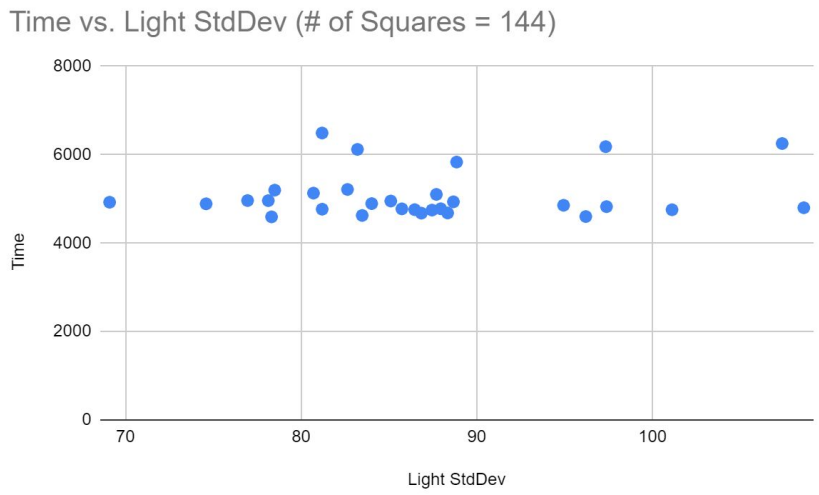
Image #4: Des Moines

Shown below is the relationship between the $(s_L + s_D)$ and t statistics for controlled cell size (40; 12x12 grid) and n = 30 trials.

Correlation between the sample standard deviation of the *scores* of the dark color scheme and the resulting computational time: 0.3666766673

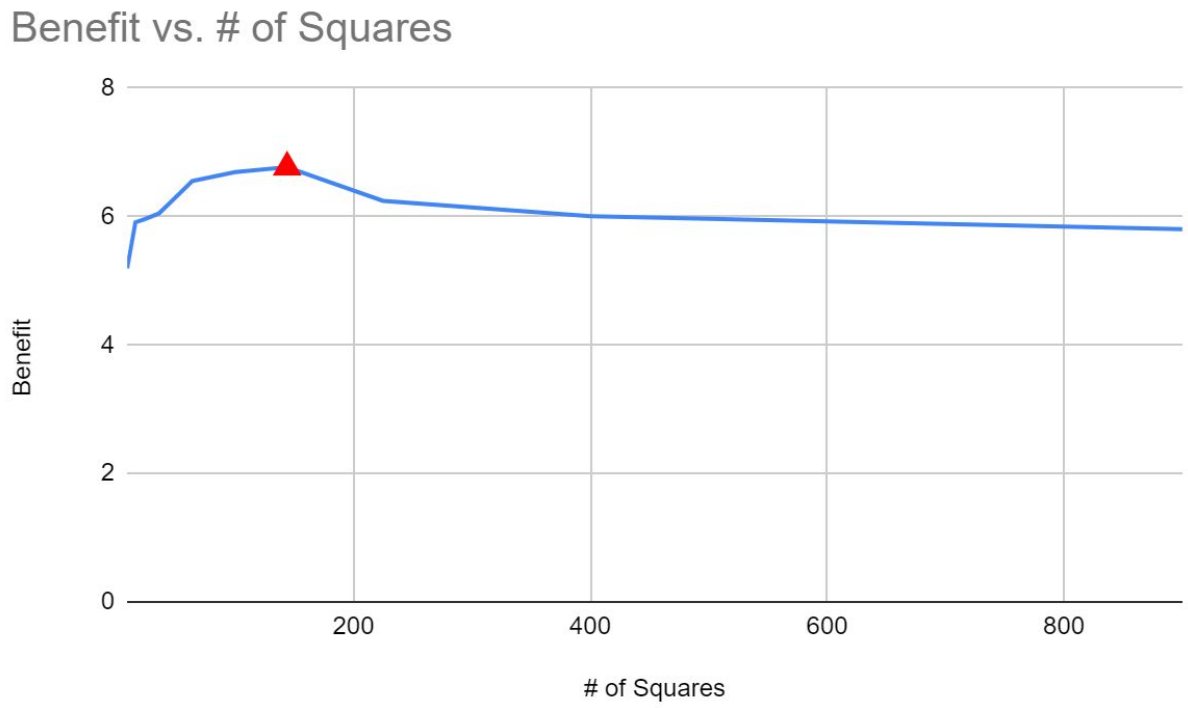


Correlation between the sample standard deviation of the *scores* of the light color scheme and the resulting computational time: 0.1304193135



Because the correlation is weak, there is evidence to suggest that computational time t and $(s_L + s_D)$ are independent. Therefore, the *benefit* that will be analyzed as the dependent variable will be $\frac{(\bar{x}_{s_l}+\bar{x}_{s_D})^2}{\bar{x}_t}$.

Using this statistic as the dependent variable, this image as a control, and using the cell size as the independent variable, the cell size can be optimized with $n = 30$ trials.



The red triangle indicates the cell size that optimizes the *benefit*, for this image, that cell size is 40, or a 12x12 grid, where the # of partitions/squares = 144.

Using this optimized cell size and optimized bin size, an ideal dark-themed and an ideal light-themed color scheme can be generated from this image:

Light-themed Color Scheme:

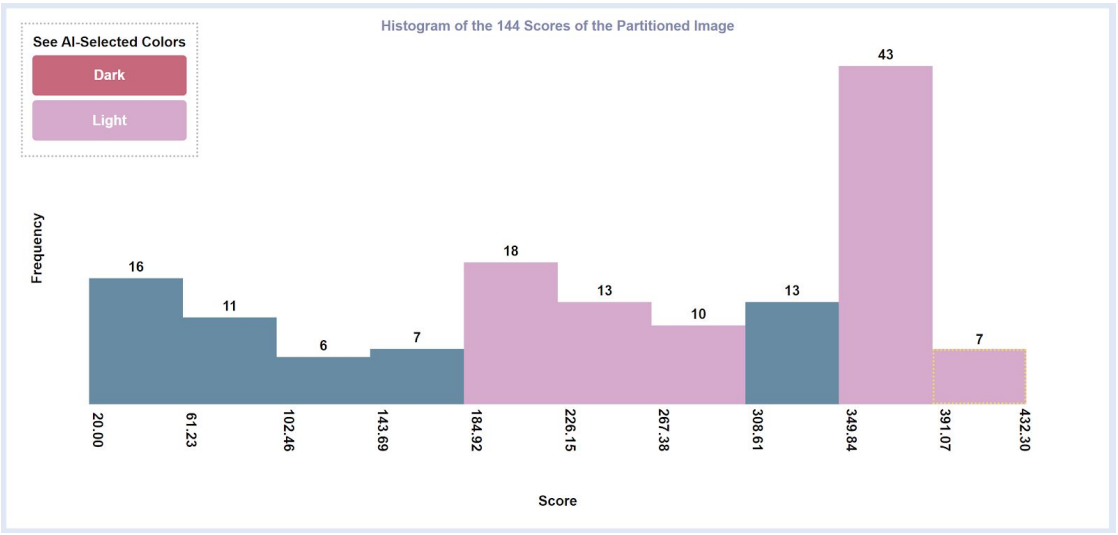
#6C7266

#818884

#9FA1A1

#B2C9E8

#DFE5F3



Dark-themed Color Scheme

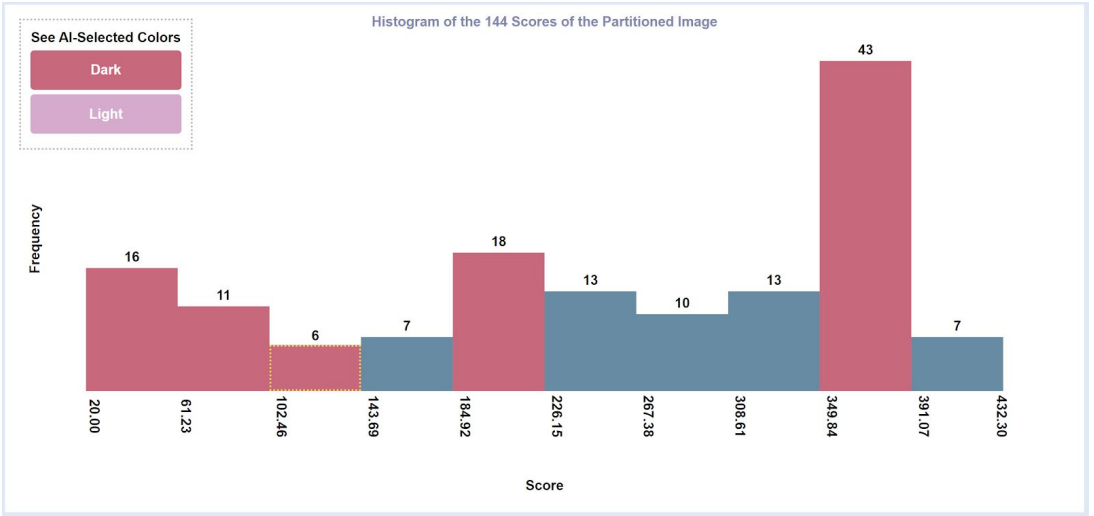
#0C110B

#332B19

#51543E

#717264

#C5D2EB



Conclusion: With the statistical applications applied in this study, color schemes can be generated through selection and trial. The critical aspect of this generation method is random selection. Random selection allows for color diversity. The theory behind this technique centers around the notion that multiple ideal color schemes can be generated from the same image. Whether any one color scheme from a specific generation is better than a color scheme from another is up to the eye of the beholder. This algorithm allows for several configurations of color values derived from the image. Moreover, random selection allows for the more common color values in an image to take higher precedence over the minority colors. The objective of this process lies in the representation: for the purposes of this study, a good color scheme with respect to the image at hand should be representative of the colors presented in that image.

The trial aspect in this selection-and-trial type of algorithmic generation comes from the statistical checkpoints put in place. Colors that look similar to each other with respect to the colors presented image as a whole will, theoretically, look good together. This is done by checking whether each prospective color to be added to the color scheme is within two standard deviations of the *anchor* color. This is where flexibility comes into play: simply change the *anchor* and all subsequent colors in the color scheme will be based around that color. The second statistical checkpoint comes in the evaluation of bin size. Yes, colors should be similar to each other, but simultaneously, however, they should not be too similar. Measuring how similar each color is relative to every other selected color is accomplished through bin size optimization. The bin size is optimized such that each bin realistically summarizes the target population. Assuring that each color is outside every other color already selected in the color scheme by at least one bin size satisfies this constraint. With these statistical constraints, combined with the power of random selection, color schemes generated per each image can look appealing while at the same time are representative of the image.

The final component of a successful color scheme generation comes from an optimized resolution, in other words, an optimized number or partitions or squares (cell size). In order for the implications of this application to be useful and practical, color scheme generations should be computed in a relatively efficient manner. This study found that there is no substantial gain by allowing for the number of partitions to lead into the thousands. By using the *benefit* as a means of determining which resolution optimizes efficiency, it was found that the net *benefit* caps out at around 100 to 144 partitions. In the images analyzed, three of the four images found that a cell size of 48 (where # of squares = 100) optimized *benefit*, whereas the other image found that a cell size of 40 (where # of squares = 144) optimized *benefit*. Therein lies a limitation with this study. The optimal resolution is not absolute; different images require different resolutions. The trend that this study unveiled was that a cell size that partitioned the image into a hundred or so squares was the most beneficial one. More data are required to unearth the true line of best fit. For the purposes of this study, however, a cell size of 48 will be kept as the universal resolution. This study only provides a baseline structure for statistical inference with color theory. More research will have to be done. This study has served its purpose. A practical color scheme

service, backed by statistical algorithms and research, can now be used by web designers and color enthusiasts.