My M&M OCD

Yoni

07 04, 2025

Intro

The goal of this simulation is to test the statistics of M&M and other stacks even Chocolate lentils by color, I wanted to know, if I eat m&m package 2 by 2, separated by color, what is the chance of my finishing the package without mixing any color in one bite.

In addition, here are some BI incite that needed to be checked:

1. What is the probability of M&M packages packaged fairly?
2. What is the probability of M&M packages packaged without one color?
3. How does the size of the package or number of colors affect this probability?

Method

The method is based of simulation of some M&M bags, according to the most common sizes. Each time we sample x lentils, name them by colors (V1,V2...), and see the results for many packages as a statistic data.

Parameters

Basic parameters:

```
#parameters
nn<- 800            #numbers of bags per sample
n_color<- 6         #unique colors of M&M
gram<- 0.91         #weight of one M&M
bag_g<- 250         #common weight of M&M package
n_unit<- bag_g/gram #M&M per package
av_per_color= n_unit/n_color
paste0("The avarage number of lentils per color is ", round(av_per_color,2))
```

```
## [1] "The avarage number of lentils per color is 45.79"
```

Creating of the Sample

General Sample

In order to test the theoretical data, we ned to simulate it using costumize functions. here are there:

- create_bag- function to create one snack package for chosen pachage size and number of colors.
- sample_MnM- function to create n bags from the create_bag function.

```
## [1] "One bag:"
```

```
##       1  2  3  4  5  6
## [1,] 22 17 13 14 15 19
```

```
## [1] "3 bags:"
```

```
##       Red Blue Green Orange Yellow Brown
## Bag_1   1    1     2      4      1     2
## Bag_2   1    3     1      1      1     3
## Bag_3   1    2     2      2      1     2
```

Preview Graph

Now will be creating nn bugs of M&M

columns:

1. V1:V6- the number of lentils per color
2. even_count- how many evens colors there are
3. even_evens- are the uneven colors even
4. Variance- variance of lentils per color
5. low_col- sum true if one color's count is lower than $\frac{2}{3}$ of expected value
6. min- the lowest color in each row

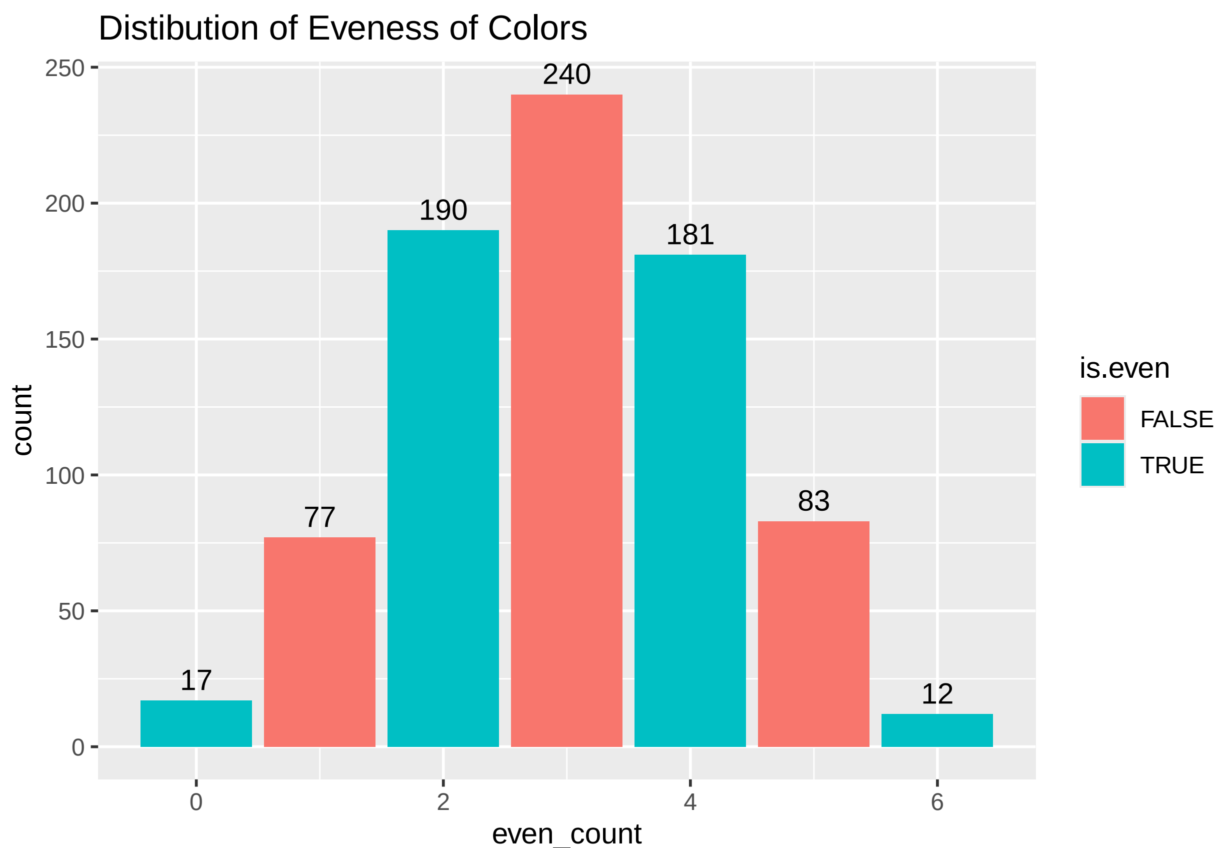
here are the first rows:

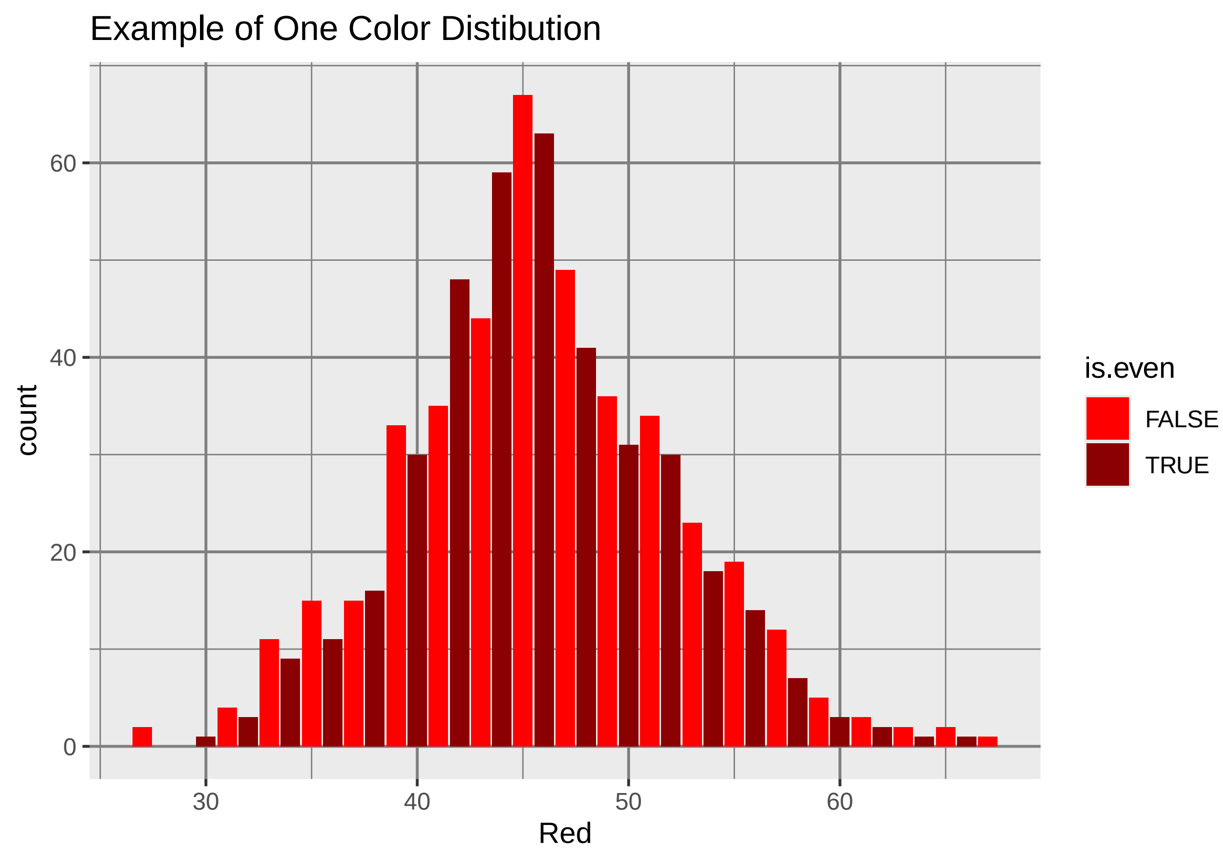
```
## # A tibble: 6 x 11
##     Red  Blue Green Orange Yellow Brown even_count even_evens low_col Variance
##   <int> <int> <int>  <int>  <int> <int>      <dbl> <lgl>        <dbl>    <dbl>
## 1    56    51    38     51     39    40          3 FALSE            0     59.8
## 2    46    45    43     52     38    51          3 FALSE            0     27.0
## 3    49    49    54     42     41    40          3 FALSE            0     31.8
## 4    37    48    47     38     54    51          3 FALSE            0     47.8
## 5    43    52    45     48     42    45          3 FALSE            0     13.4
## 6    59    45    37     38     55    41          1 FALSE            0     84.2
## # i 1 more variable: min <int>
```

plot the M&M sample sample

```
## [1] "summary of all colors Distibution:"
```

```
##         Min. 1st Qu. Median     Mean 3rd Qu. Max.      Var
## Red       27      42     45 45.77500      50   67 39.42365
## Blue      26      41     45 45.63750      50   65 38.28895
## Green     26      42     46 45.83625      50   65 38.34236
## Orange    28      42     46 45.88250      50   71 38.97992
## Yellow    25      41     45 45.68500      50   65 39.72043
## Brown     29      41     45 45.68375      50   63 39.14517
```





Statisics Checking of the Simullation

Test Expected Value

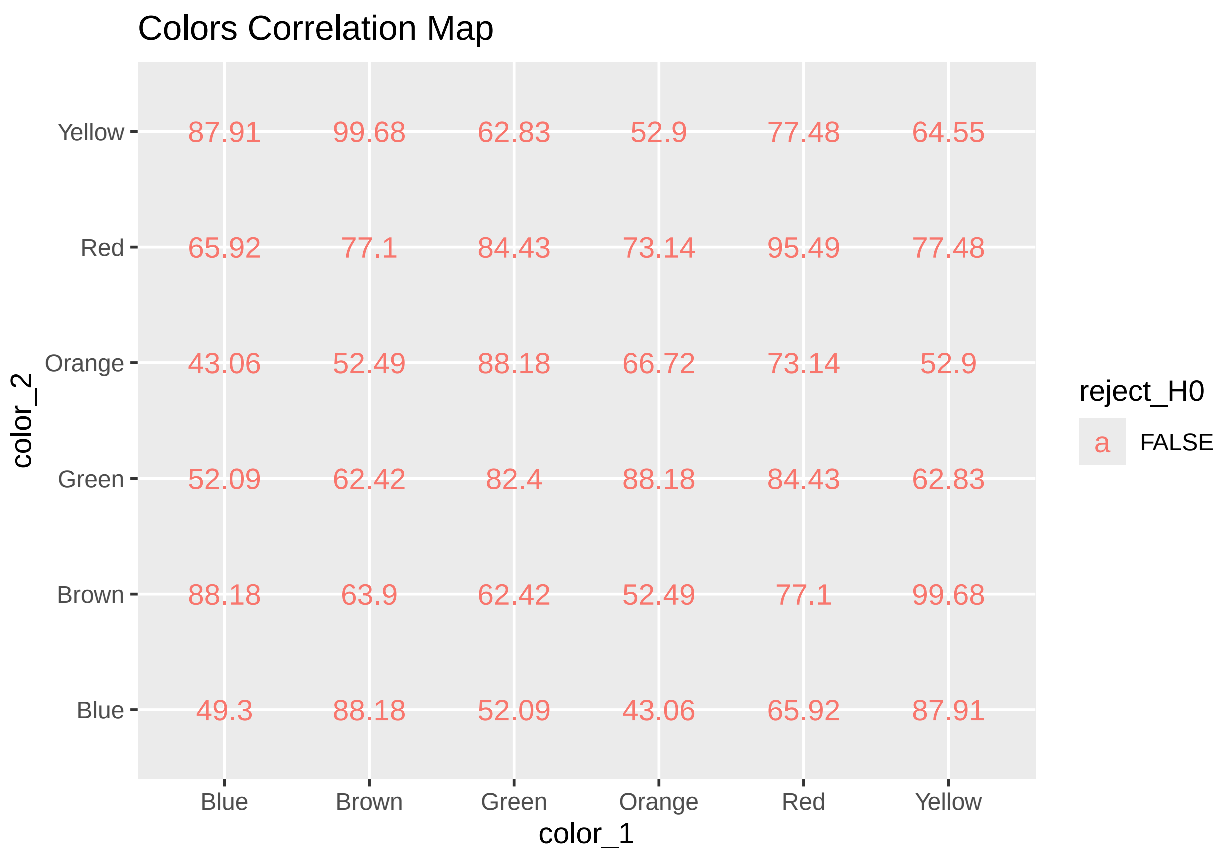
to see is the μ of the lentils per color are fair, we will test it per column with t.test for each color.

Here is the result, none of them bellow 5% P. value

```
##       Red      Blue     Green    Orange    Yellow     Brown
## "95.49%" "49.30%" "82.40%" "66.72%" "64.55%" "63.90%"
```

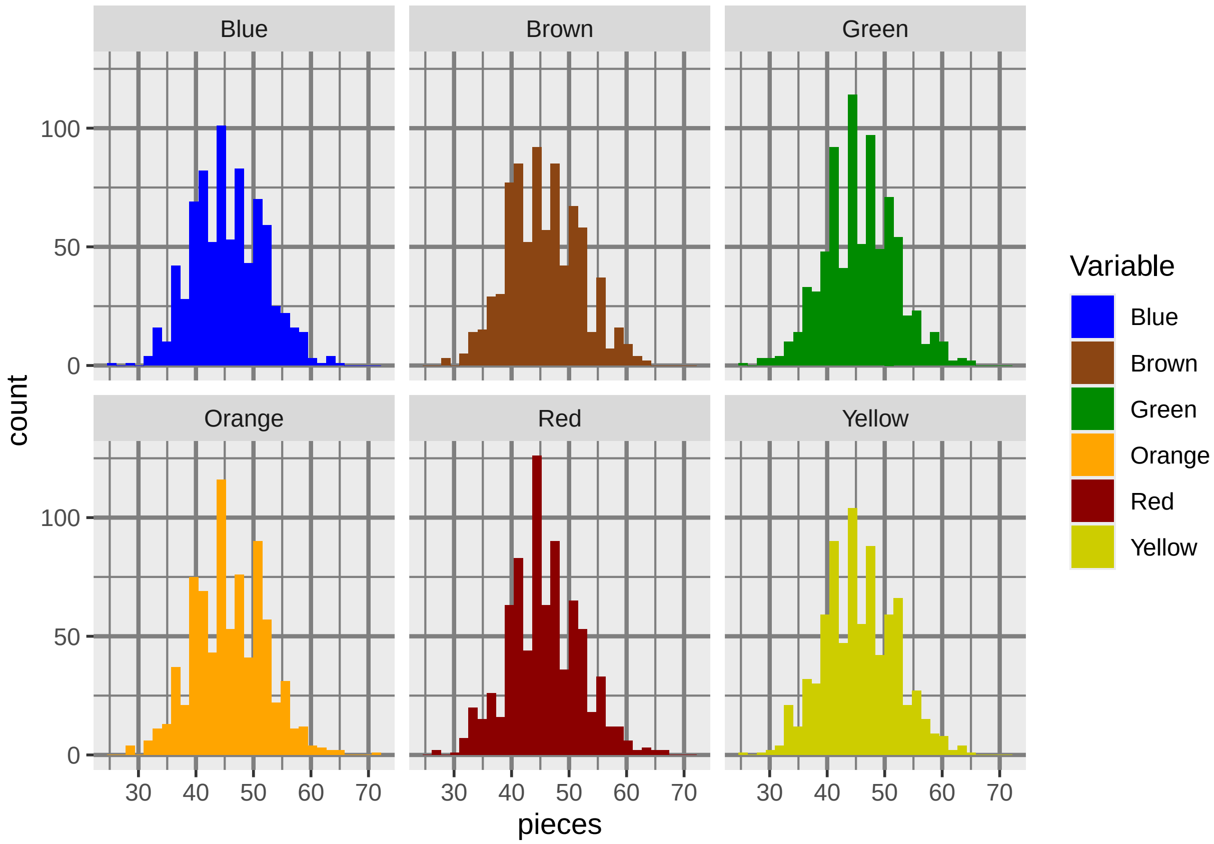
Now we will do the same checking for 2 samples, to see whether there is correlation between each 2 colors distribution.

for each row i and column j, 1) if i==j, this it the check from before of the expected value to n_unit/n_color 2) if i!=j, this is two samples test of same expected value hypothesis



Now here Is visualization of the actual data per color

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



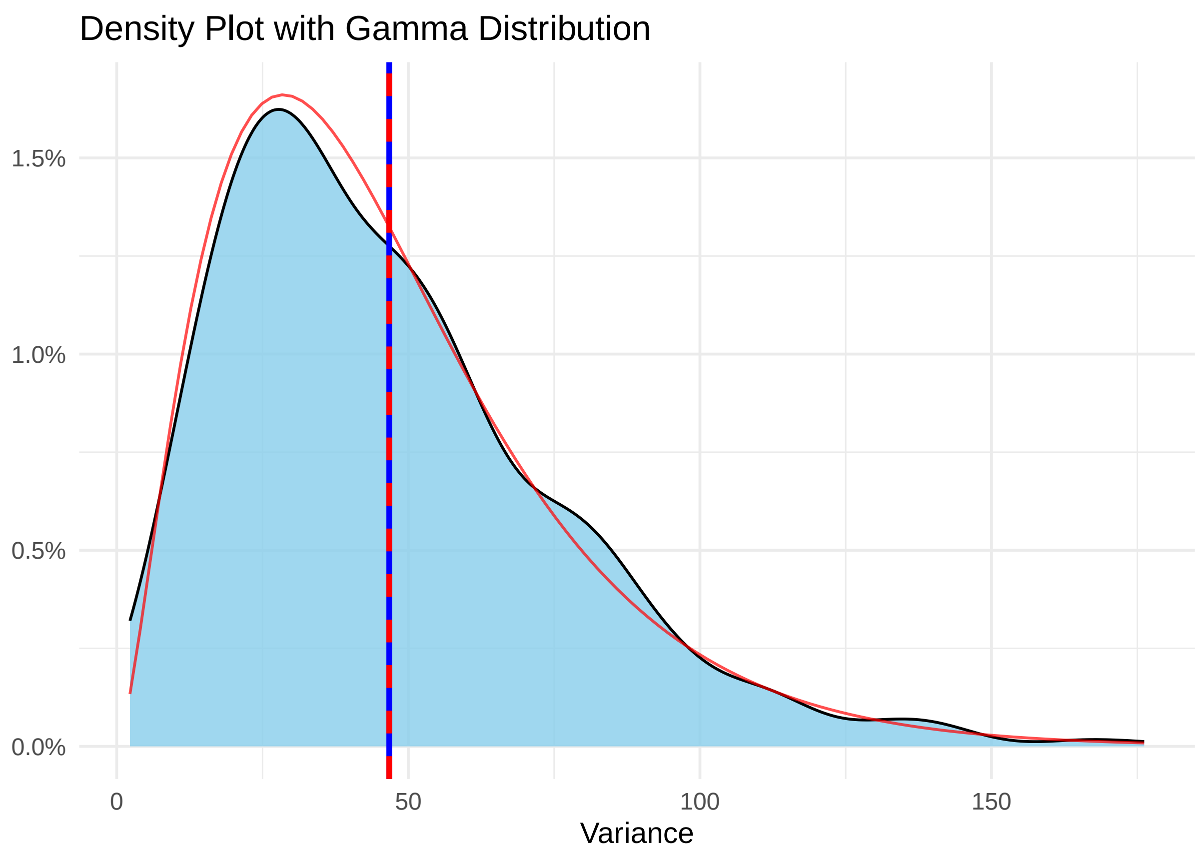
Variance Distribution Checking

We know that the distribution of variance is approximately Gamma distribution:

$$f(x) = \frac{1}{(\Gamma(\alpha)\theta^{\alpha})} x^{\alpha-1} e^{-x/\theta}$$

We can see that the variance distribution is Gamma like with shape and rate as seen below

```
## [1] "The parameters of the gamma shaped variance is shape 2.565 and rate 0.055"
```



#use statistics to sample better low chance cases

n*m types of snacks

We will create a function that create sample for each number of colors and package size we want, and then calculate some interesting parameters

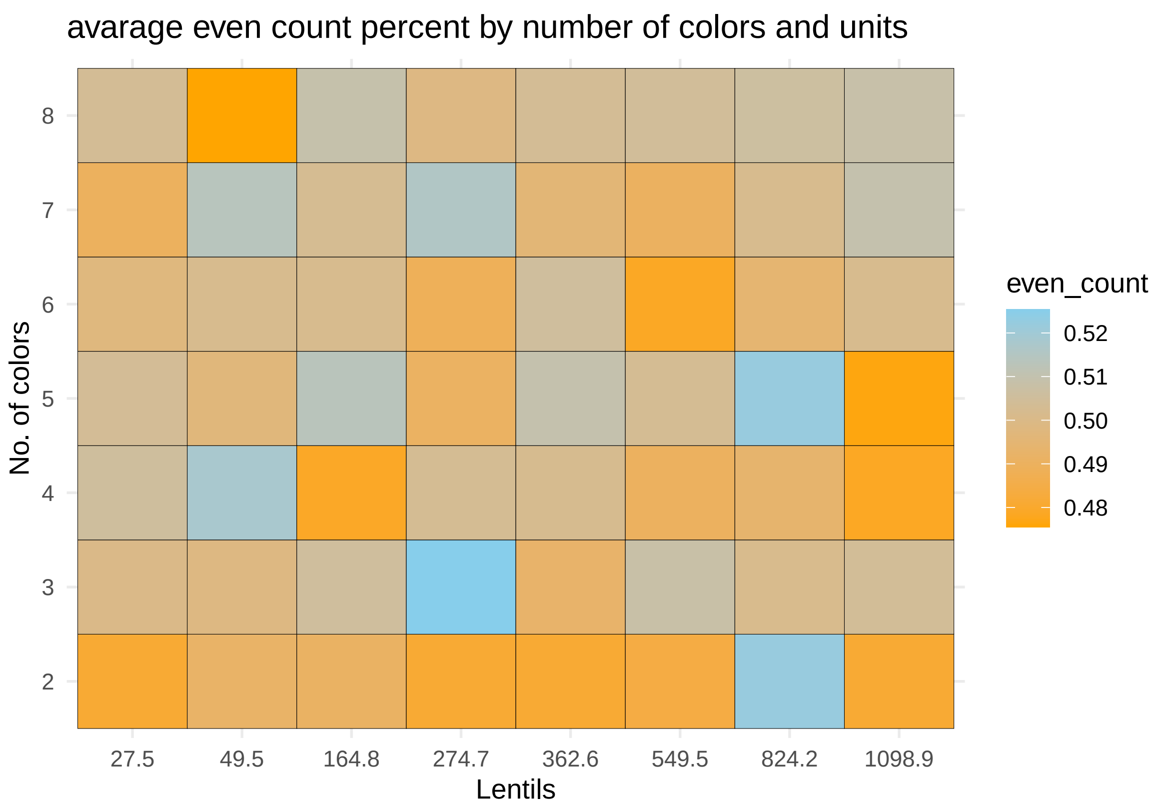
```
color_op<- 2:8
grams_op<- c(25,45,150,250,330,500,750,1000)
n_unit_op<- grams_op/gram
nn=500
```

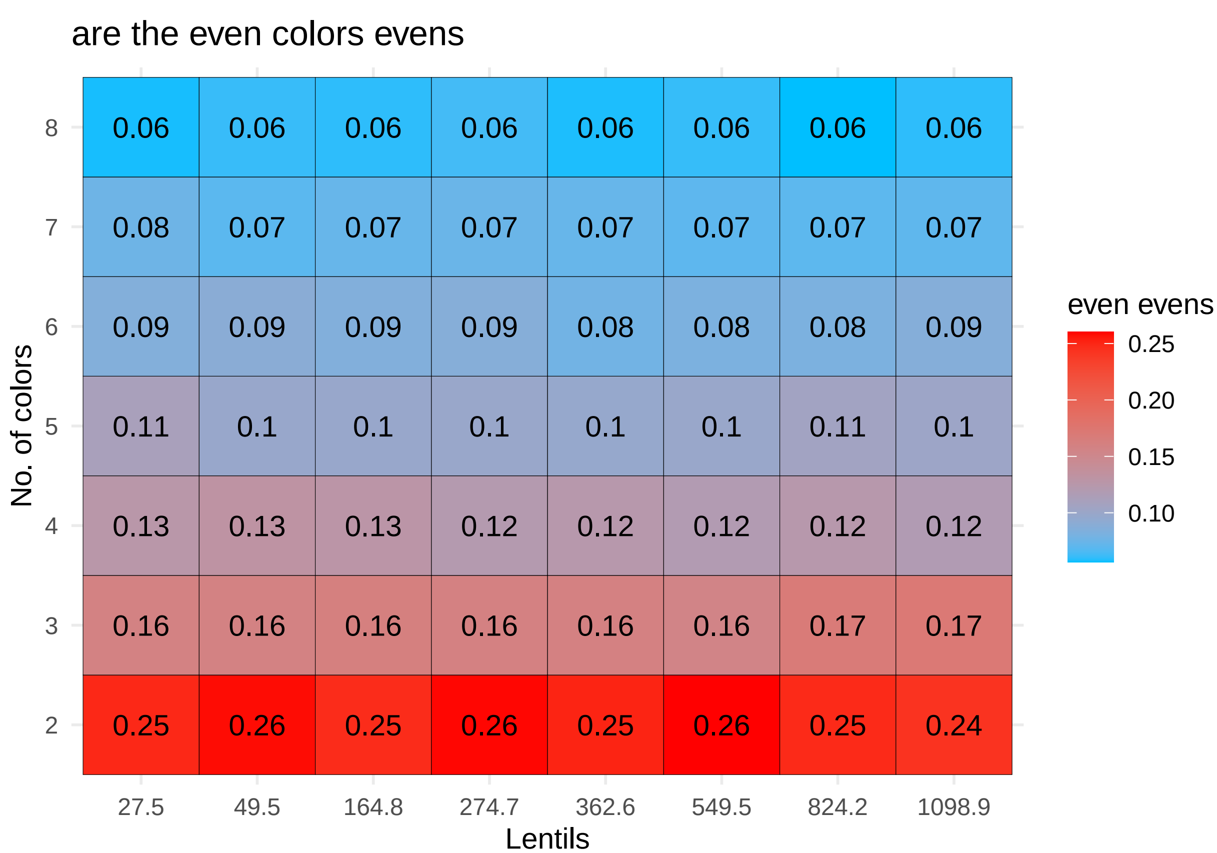
We will make the multiple sample. Here is some random rows:

```
##   n_unit n_color even_count even_evens var_col all_even low_color smallest_col
## 1 1098.9       8  0.5085000 0.05975000 132.053        0     0.000           96
## 2  362.6       7  0.4957143 0.07342857  49.611        0     0.002           33
## 3  549.5       7  0.4902857 0.07028571  93.030        0     0.002           50
## 4  274.7       4  0.5030000 0.12150000  67.737        0     0.000           46
## 5  362.6       6  0.5053333 0.07800000  55.754        0     0.002           38
```

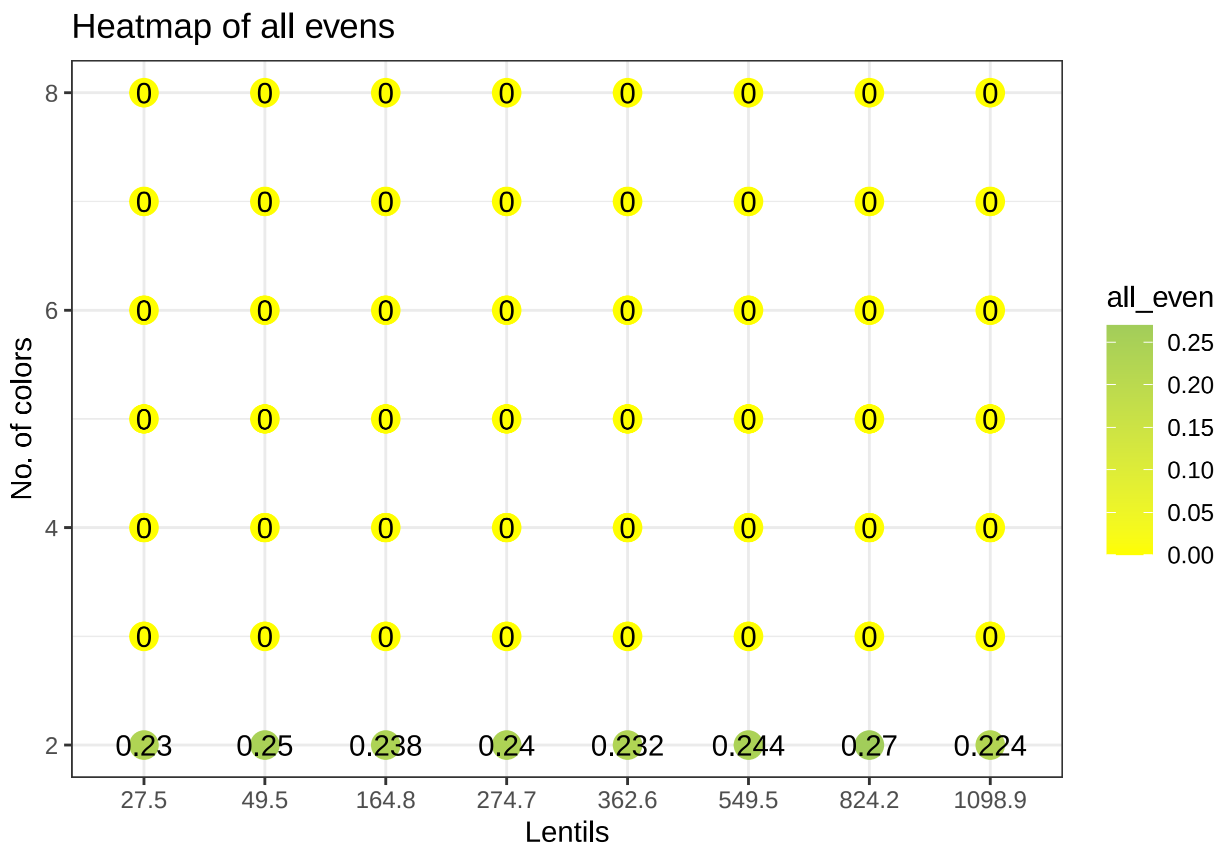
Deep Insight on the Data

here are some insights:

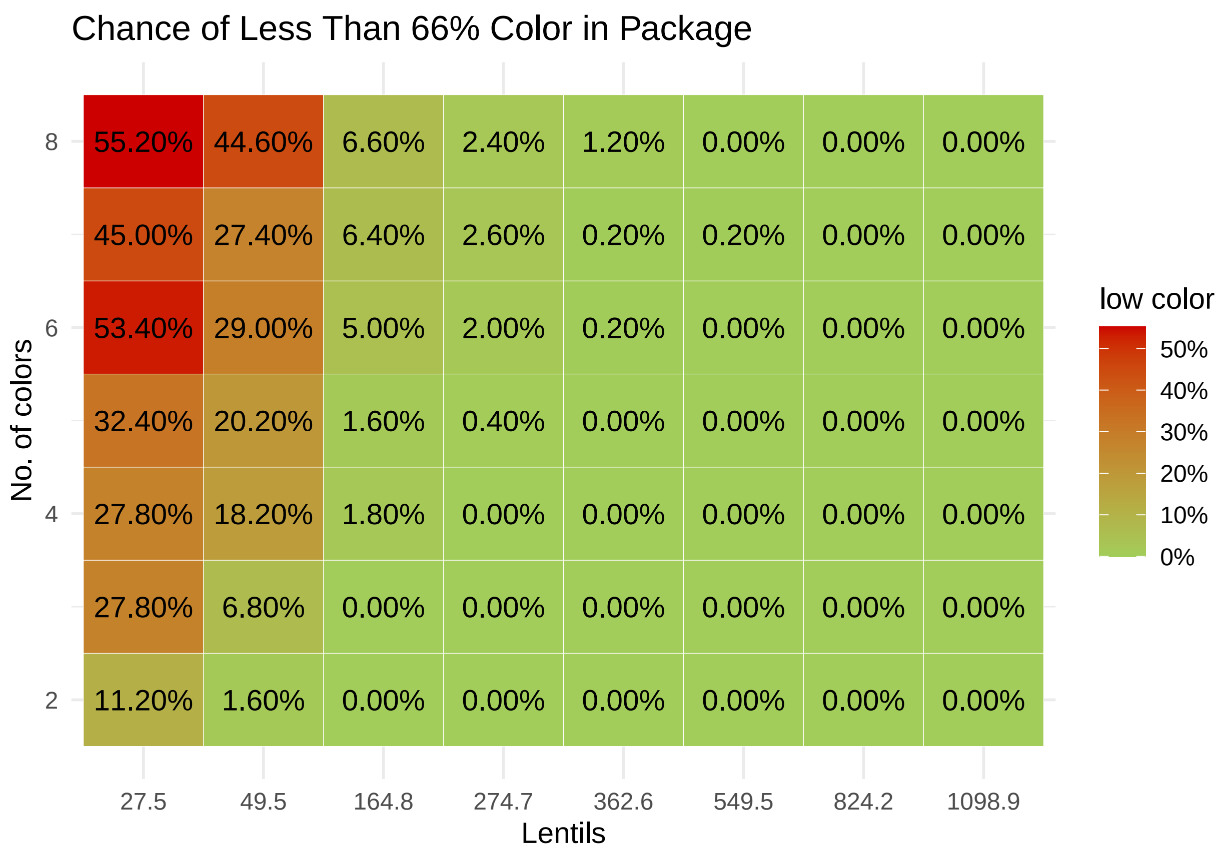

avarage even count percent by number of colors and units
No. of colors
8
7
6
5
4
3
2
27.5
49.5
164.8
274.7
362.6
549.5
824.2
1098.9
Lentils
even_count
0.52
0.51
0.50
0.49
0.48


are the even colors evens
No. of colors
8 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06
7 0.08 0.07 0.07 0.07 0.07 0.07 0.07 0.07
6 0.09 0.09 0.09 0.09 0.08 0.08 0.08 0.09
5 0.11 0.1 0.1 0.1 0.1 0.1 0.11 0.1
4 0.13 0.13 0.13 0.12 0.12 0.12 0.12 0.12
3 0.16 0.16 0.16 0.16 0.16 0.16 0.17 0.17
2 0.25 0.26 0.25 0.26 0.25 0.26 0.25 0.24
27.5
49.5
164.8
274.7
362.6
549.5
824.2
1098.9
Lentils
even evens
0.25
0.20
0.15
0.10

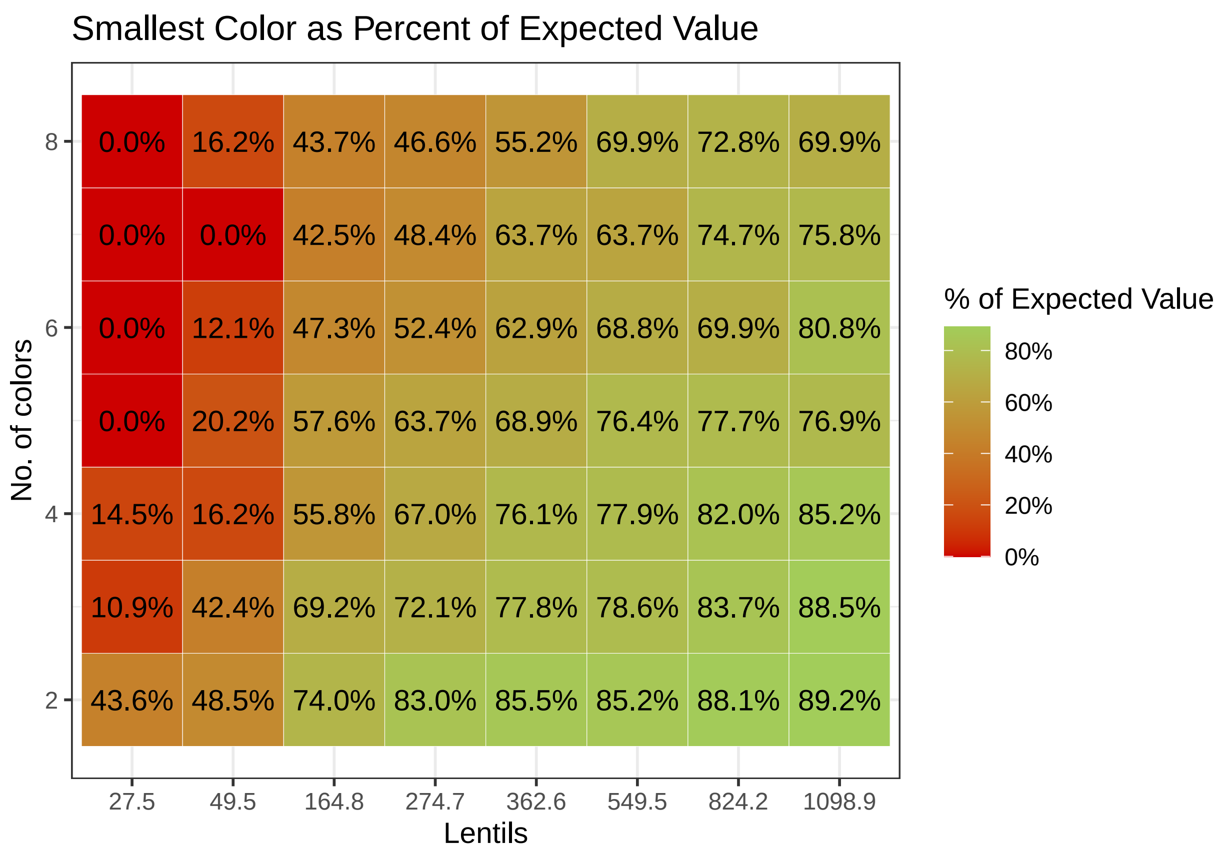
now let us see the probability of all even, and whether there is pattern.




```
## <Guides[1] ggproto object>
##
## colour : "none"
```



Chance of Less Than 66% Color in Package
No. of colors
8
6
4
2
55.20%
44.60%
6.60%
2.40%
1.20%
0.00%
0.00%
0.00%
45.00%
27.40%
6.40%
2.60%
0.20%
0.20%
0.00%
0.00%
53.40%
29.00%
5.00%
2.00%
0.20%
0.00%
0.00%
0.00%
32.40%
20.20%
1.60%
0.40%
0.00%
0.00%
0.00%
0.00%
27.80%
18.20%
1.80%
0.00%
0.00%
0.00%
0.00%
0.00%
27.80%
6.80%
0.00%
0.00%
0.00%
0.00%
0.00%
0.00%
11.20%
1.60%
0.00%
0.00%
0.00%
0.00%
0.00%
0.00%
27.5
49.5
164.8
274.7
362.6
549.5
824.2
1098.9
Lentils
low color
50%
40%
30%
20%
10%
0%



As we can see, only the small package (less than 50 lentils) have high probability of at least one color to appear severely lower.

Therefore, splitting package by color on the big ones should be relatively even.

using regression for correlation check

```
##
## Call:
## lm(formula = mega_snack_2$even_count ~ mega_snack_2$n_color +
##     mega_snack_2$n_unit + mega_snack_2$color_No2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49747 -0.16013  0.01086  0.17234  0.51266
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 4.952e-01  2.692e-02  18.398   <2e-16 ***
## mega_snack_2$n_color       -6.378e-04  4.265e-03  -0.150    0.881
## mega_snack_2$n_unit         3.817e-06  2.387e-05   0.160    0.873
## mega_snack_2$color_No2TRUE -5.403e-03  1.724e-02  -0.313    0.754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2472 on 836 degrees of freedom
```

```
## Multiple R-squared:  0.0001749,  Adjusted R-squared:  -0.003413
## F-statistic: 0.04874 on 3 and 836 DF,  p-value: 0.9858
```

```
##
## Call:
## lm(formula = mega_snack_2$even_count ~ mega_snack_2$n_color +
##     mega_snack_2$n_unit + mega_snack_2$color_No2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49747 -0.16013  0.01086  0.17234  0.51266
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 4.952e-01  2.692e-02  18.398   <2e-16 ***
## mega_snack_2$n_color       -6.378e-04  4.265e-03  -0.150    0.881
## mega_snack_2$n_unit         3.817e-06  2.387e-05   0.160    0.873
## mega_snack_2$color_No2TRUE -5.403e-03  1.724e-02  -0.313    0.754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2472 on 836 degrees of freedom
## Multiple R-squared:  0.0001749,  Adjusted R-squared:  -0.003413
## F-statistic: 0.04874 on 3 and 836 DF,  p-value: 0.9858
```

Conclusions

Data Structure

The simulation created

Main Q: Eating M&M by Two

Although there is no clear pattern to the right M&M package for all the colors to have even count, maybe different approach can find a clear reason for more or less couples of M&M.

The general probability of all colors to be even is 4% for small 50g package 2.8% for big 1000g package, and overall 2.8%, which is less than I expected.

Summery