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

The basic parameters (will be changed later):

```
#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

create_bag- function to create one snack package as matrix.

sample_MnM- function to create n bags from the create_bag function.

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

##       1  2  3  4  5  6
## [1,] 17 19 14 11 23 16

## [1] "3 bags:"

##       1 2 3 4 5 6
## Bag_1 3 2 2 0 2 2
## Bag_2 2 1 1 3 3 1
## Bag_3 5 3 0 2 0 1
```

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

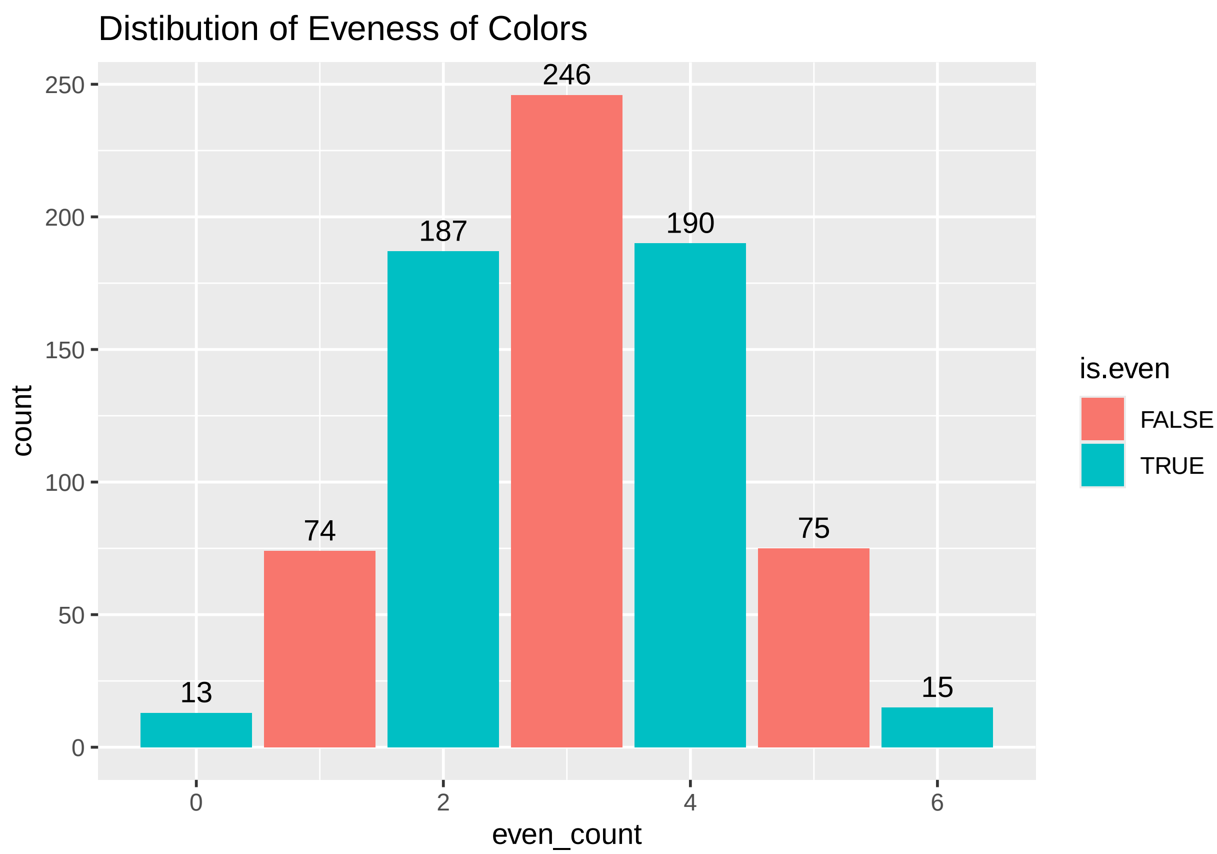
here are the first rows:

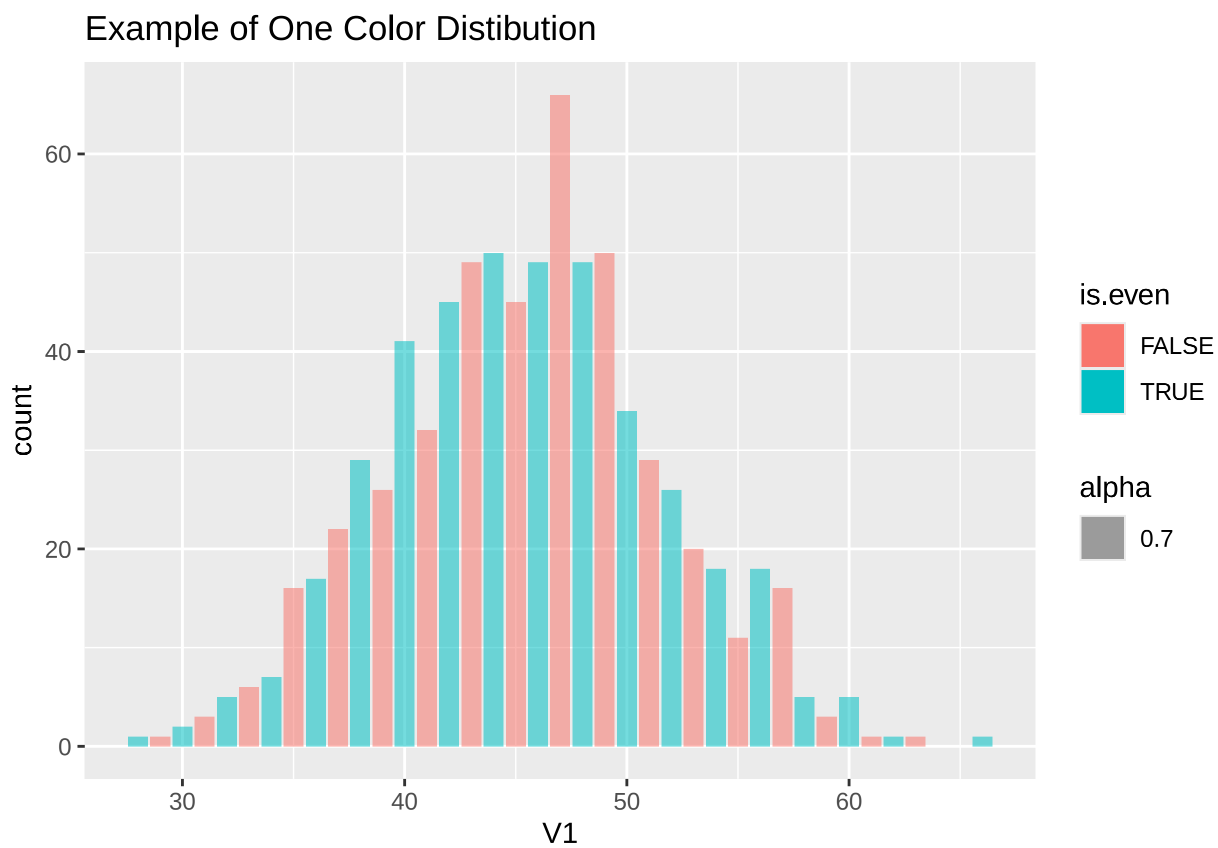
```
## # A tibble: 6 x 11
##      V1    V2    V3    V4    V5    V6 even_count even_evens Variance low_col
##   <int> <int> <int> <int> <int> <int>      <dbl> <lgl>         <dbl>   <dbl>
## 1    50    50    45    39    41    50          3 FALSE          24.6       0
## 2    49    41    45    49    40    50          2 TRUE           19.1       0
## 3    42    43    47    40    51    51          2 TRUE           22.3       0
## 4    42    40    50    45    44    54          5 FALSE          27.4       0
## 5    43    42    54    50    46    40          5 FALSE          28.2       0
## 6    43    52    60    38    37    44          4 TRUE           77.9       0
## # i 1 more variable: min <int>
```

plot the MnM sample sample

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

##    Min. 1st Qu. Median     Mean 3rd Qu. Max.      Var
## V1   28      41     46 45.43375      49   66 37.33228
## V2   27      41     46 45.87500      50   66 41.49625
## V3   28      41     45 45.65500      49   69 38.55917
## V4   28      42     45 45.69625      50   64 38.13916
## V5   23      41     46 45.73625      50   65 37.54612
## V6   30      42     46 46.09750      50   65 37.33967
```


Distibution of Eveness of Colors
250
200
150
100
50
0
count
246
187
190
74
75
13
15
0
2
4
6
even_count
is.even
FALSE
TRUE



Test Expected Value

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

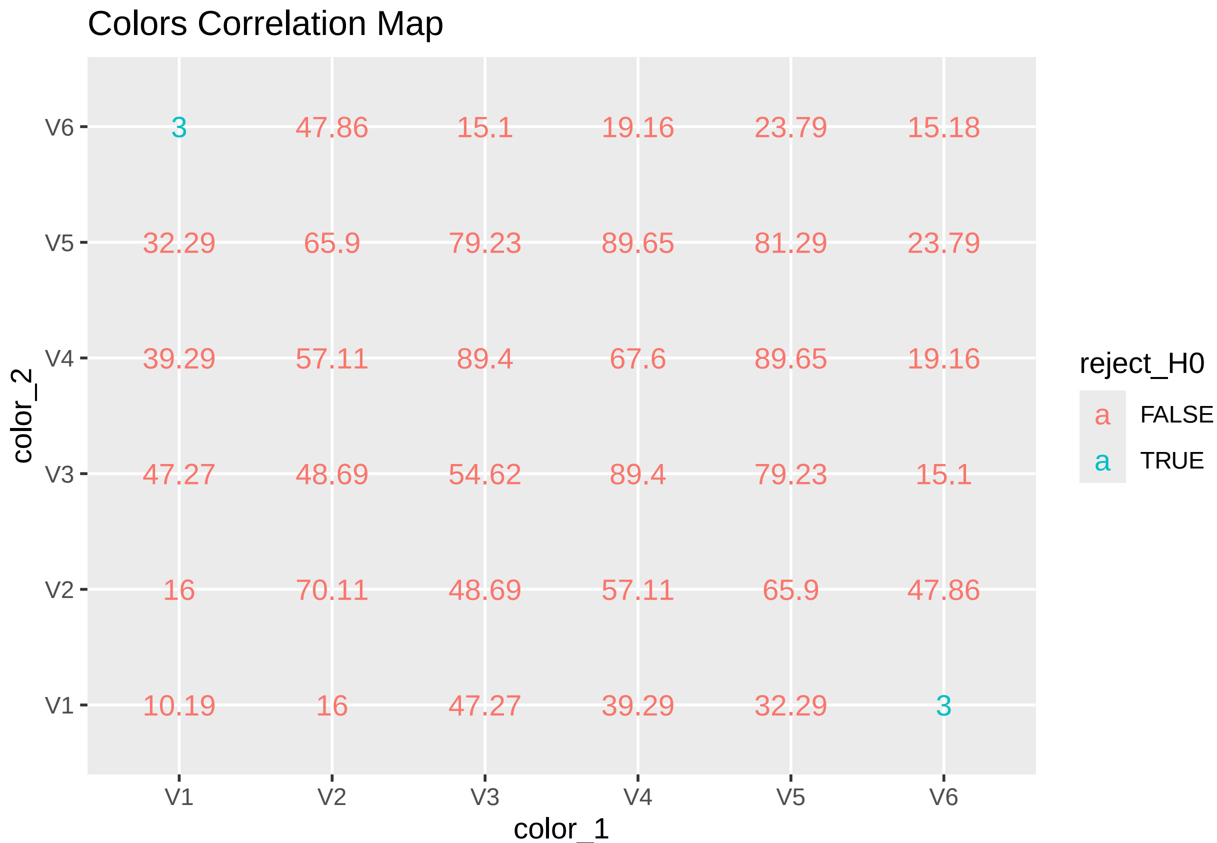
Here is the result, none of them bellow 5% P. value

```
##      V1      V2      V3      V4      V5      V6
## "10.2%" "70.1%" "54.6%" "67.6%" "81.3%" "15.2%"
```

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

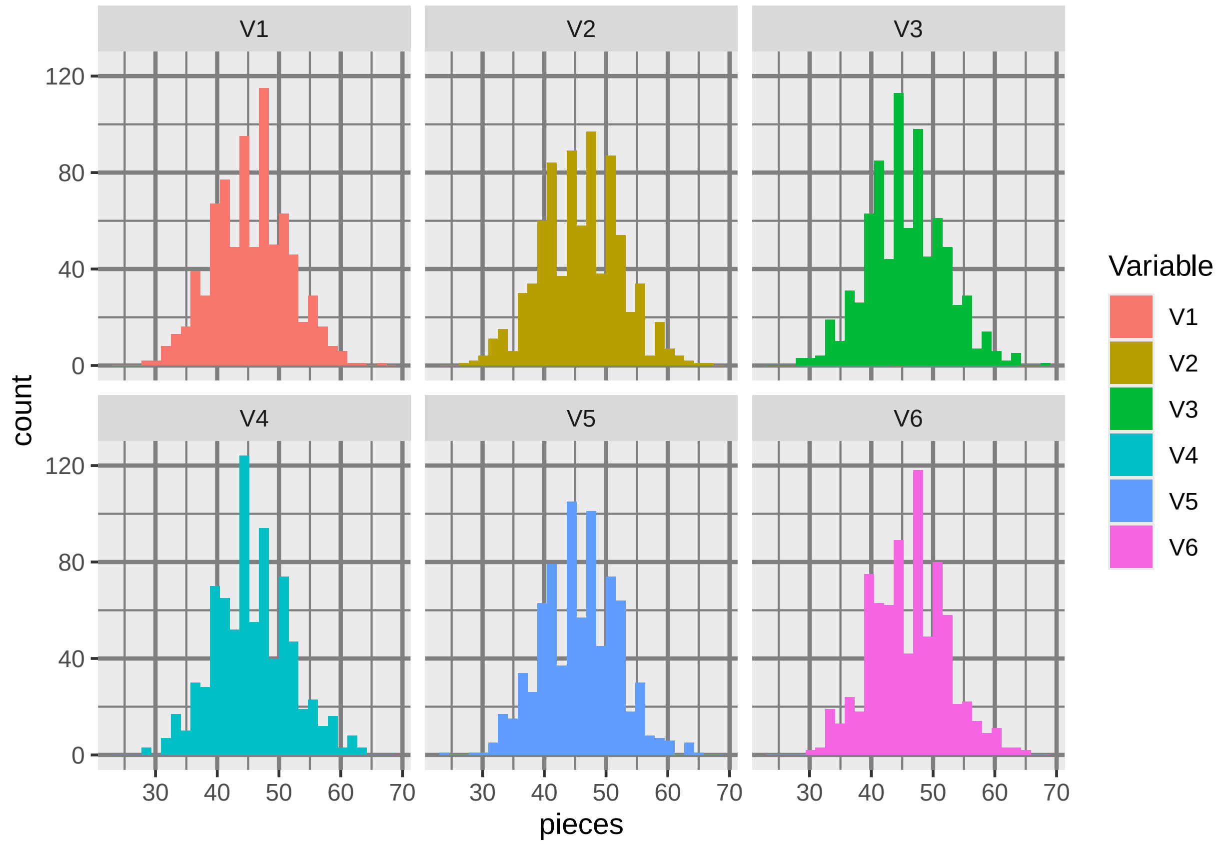
for each row i and column j, 1) if i==j, this it the check from before of the expected value to n_unit/n_color 2) if i!=j, this is two samples test of same expected value hypothesis

```
##        V1     V2     V3     V4     V5     V6
## V1 0.1019 0.1600 0.4727 0.3929 0.3229 0.0300
## V2 0.1600 0.7011 0.4869 0.5711 0.6590 0.4786
## V3 0.4727 0.4869 0.5462 0.8940 0.7923 0.1510
## V4 0.3929 0.5711 0.8940 0.6760 0.8965 0.1916
## V5 0.3229 0.6590 0.7923 0.8965 0.8129 0.2379
## V6 0.0300 0.4786 0.1510 0.1916 0.2379 0.1518
```



now here Is visualization of the actual data per color:

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



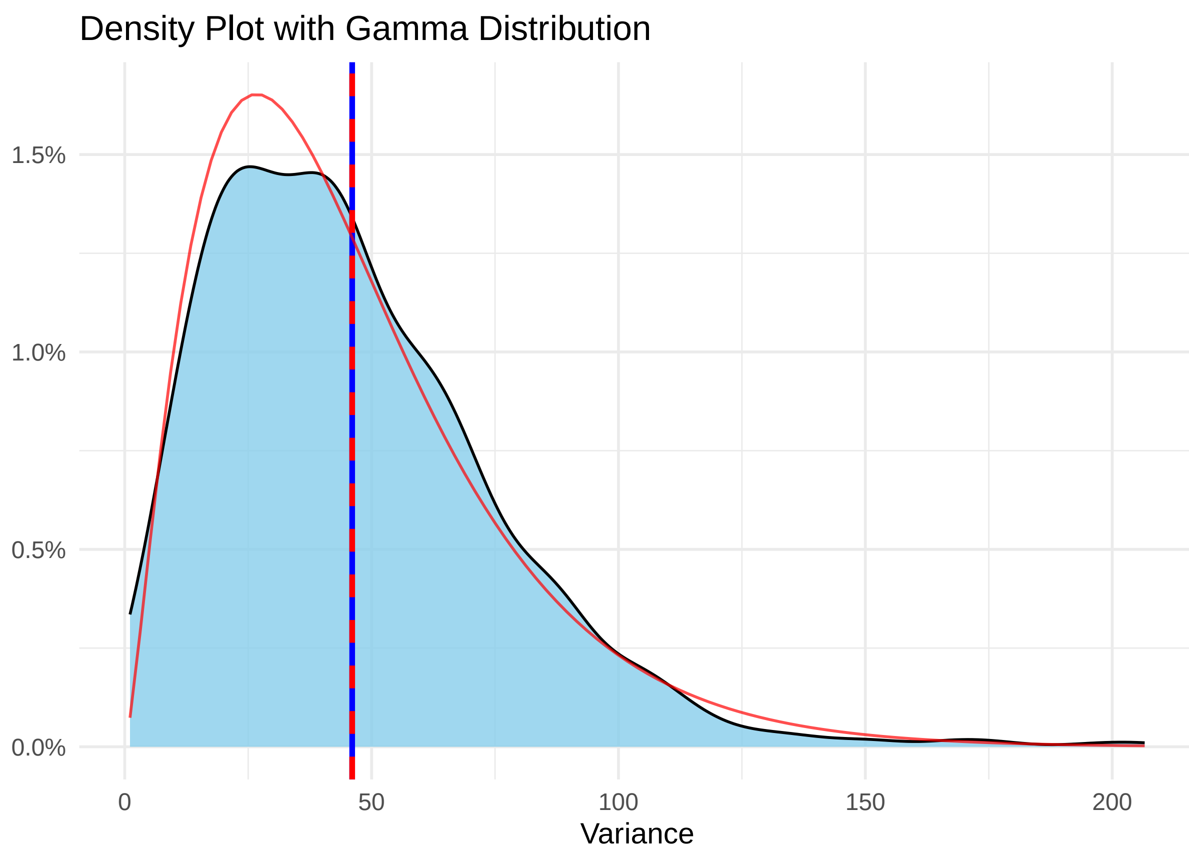
Variance Distribution Checking

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 2.377 and rate 0.052"
```



#use statistics to sample better low chance cases

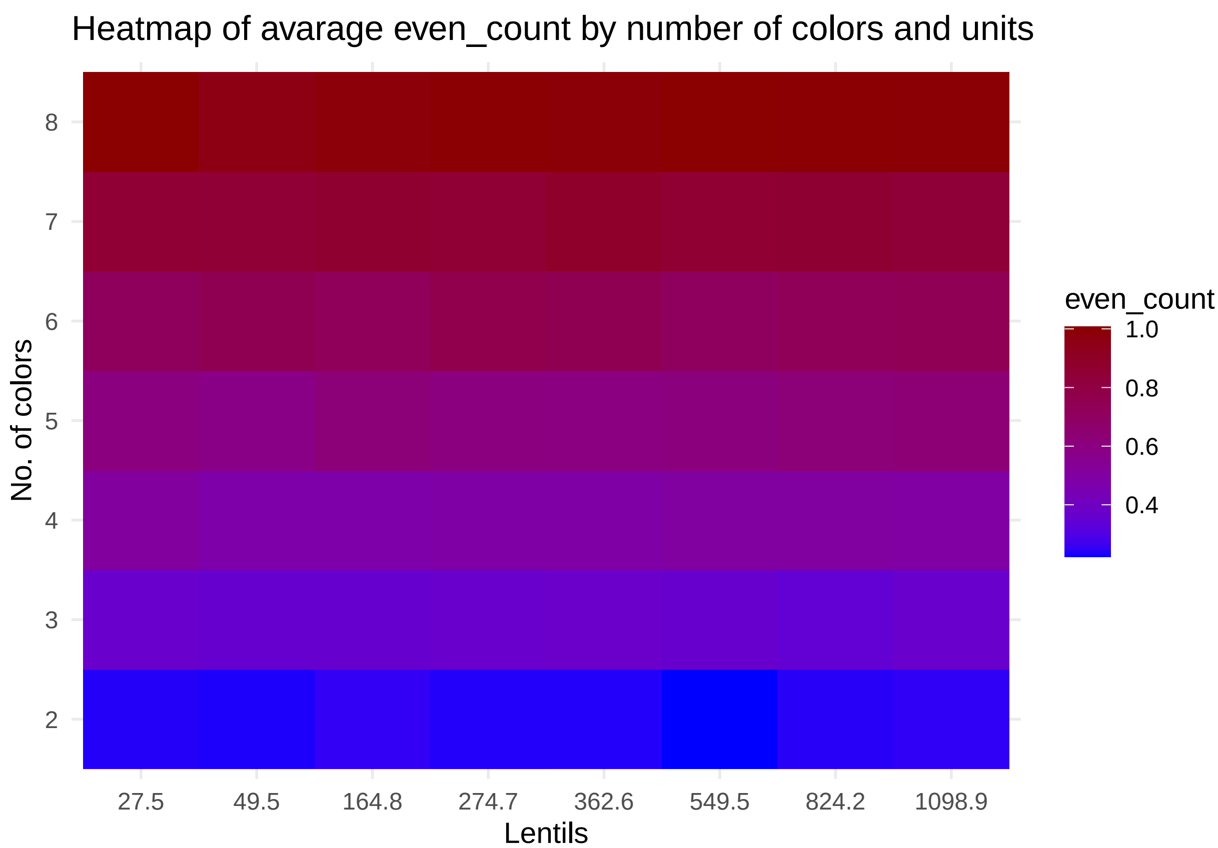
n*m types of snacks

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  164.8       8  0.9878667  0.1406024  18.357    0.044     0.066            9
## 2  362.6       6  0.7545714  0.1216619  57.154    0.048     0.002           40
## 3 1098.9       8  0.9979071  0.1218048 133.564    0.034     0.000          108
## 4  164.8       2  0.2536810  0.1263595  79.250    0.044     0.000           64
## 5  274.7       2  0.2377976  0.1306381 125.536    0.032     0.000          114
```

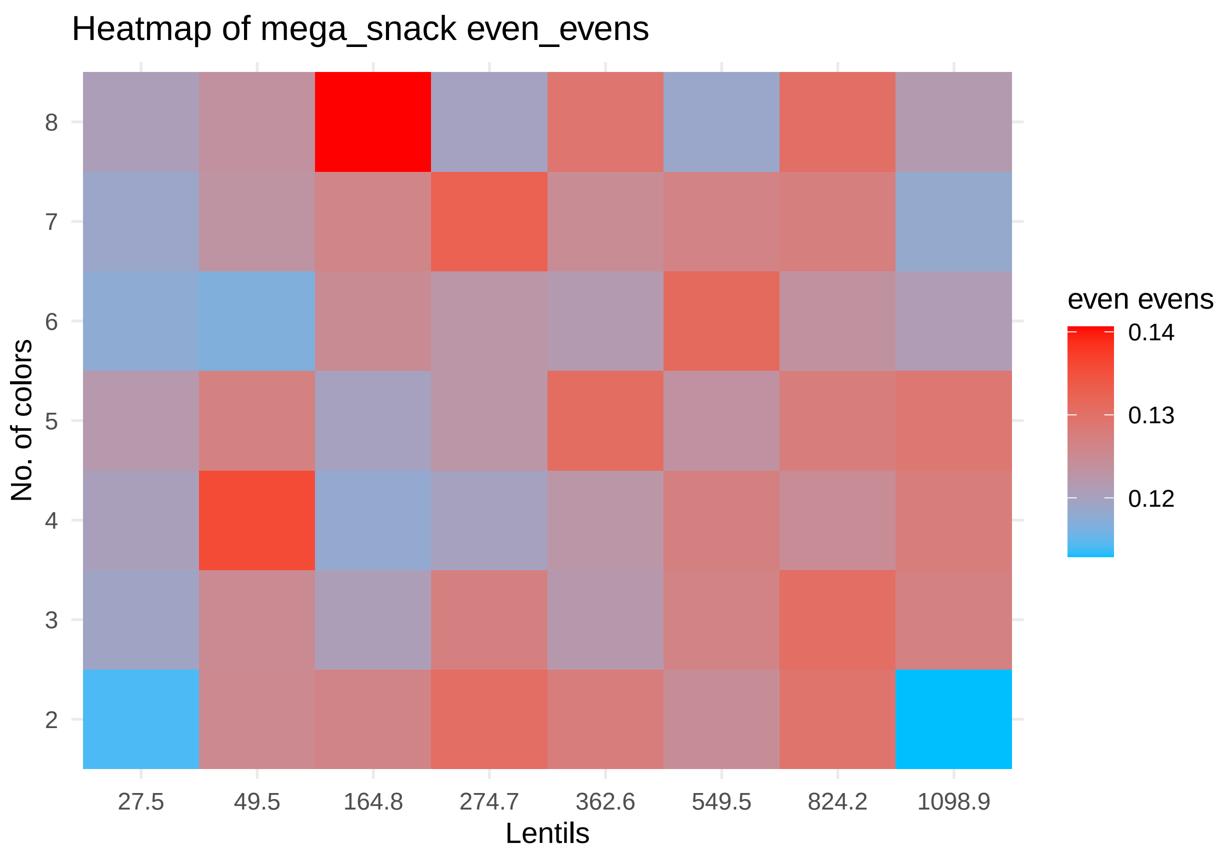



```
##
## Call:
## lm(formula = mega_snack_2$even_count ~ mega_snack_2$n_color +
##     mega_snack_2$n_unit + mega_snack_2$color_No2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52219 -0.17086 -0.01811  0.15970  0.50300
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.049e-01  2.674e-02  18.882   <2e-16 ***
## mega_snack_2$n_color        9.442e-04  4.237e-03   0.223    0.824
## mega_snack_2$n_unit        -9.761e-06  2.371e-05  -0.412    0.681
## mega_snack_2$color_No2TRUE  1.400e-02  1.712e-02   0.818    0.414
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2456 on 836 degrees of freedom
## Multiple R-squared:  0.001061,   Adjusted R-squared:  -0.002524
## F-statistic: 0.2959 on 3 and 836 DF,  p-value: 0.8284
```

```
##
## Call:
## lm(formula = mega_snack_2$even_count ~ mega_snack_2$n_color +
```

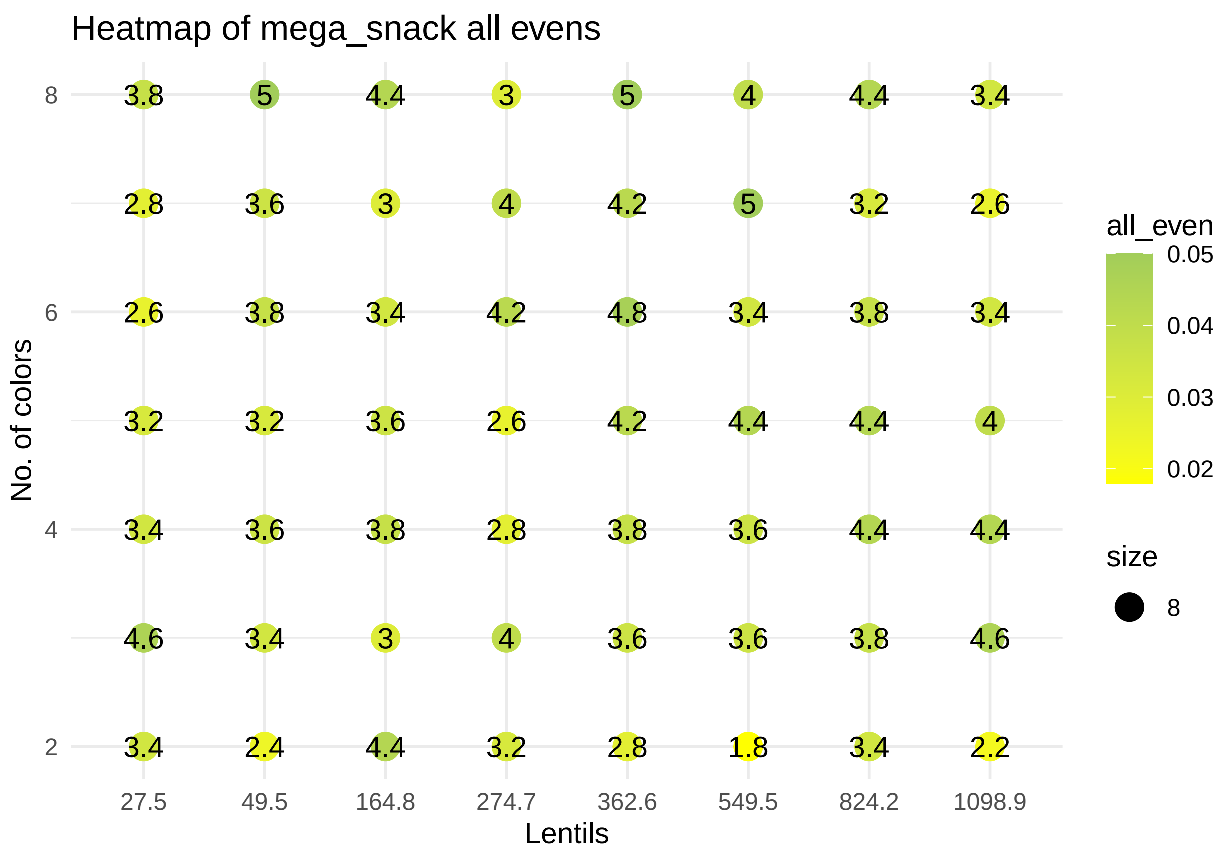
```
##     mega_snack_2$n_unit + mega_snack_2$color_No2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52219 -0.17086 -0.01811  0.15970  0.50300
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                5.049e-01  2.674e-02  18.882   <2e-16 ***
## mega_snack_2$n_color       9.442e-04  4.237e-03   0.223    0.824
## mega_snack_2$n_unit       -9.761e-06  2.371e-05  -0.412    0.681
## mega_snack_2$color_No2TRUE  1.400e-02  1.712e-02   0.818    0.414
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2456 on 836 degrees of freedom
## Multiple R-squared:  0.001061,   Adjusted R-squared:  -0.002524
## F-statistic: 0.2959 on 3 and 836 DF,  p-value: 0.8284
```

```r
mega_snack_1 %>%
  ggplot(aes(x = factor((round( n_unit,1) )), y = factor(n_color ), fill = even_evens )) +
  geom_tile() +
  scale_fill_gradient(low = "deepskyblue", high = "red")+
  labs(title = "Heatmap of mega_snack even_evens",
       x = "Lentils",
       y = "No. of colors",
       fill = "even evens") +
  theme_minimal()
```

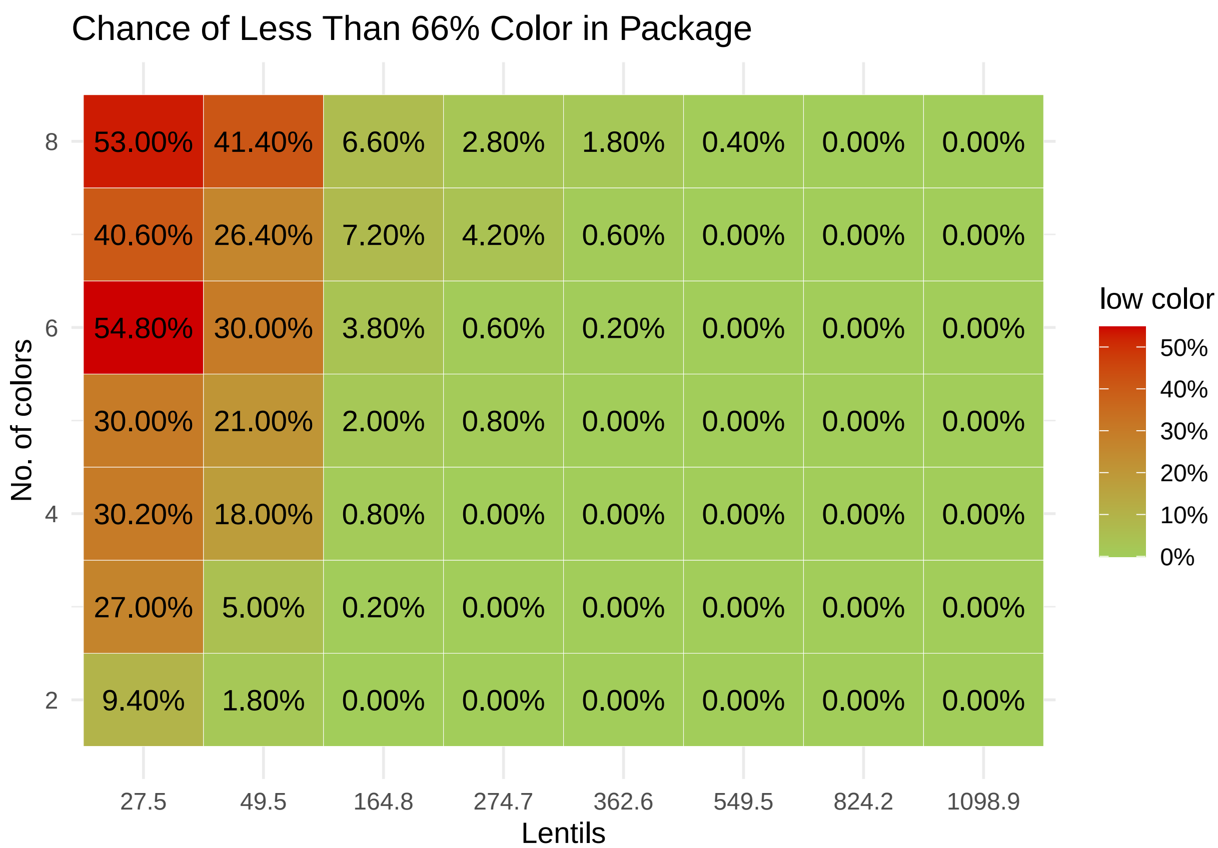


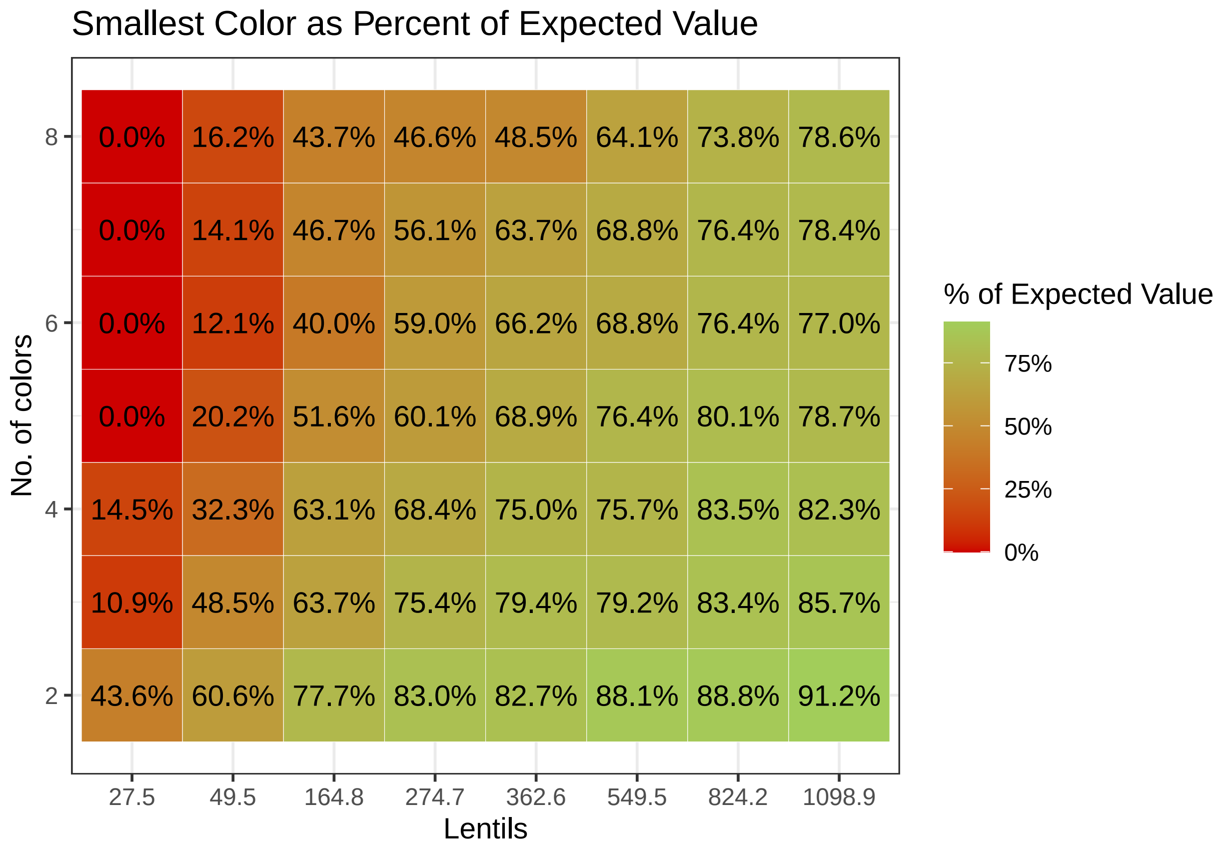
now let us see the probability of all even, and whether there is pattern.

```
mega_snack_1 %>%
  ggplot(aes(x = as.factor(round( n_unit,1) ), y = n_color , color = all_even,size = 8 )) +
  geom_point() +
  scale_color_gradient(low = "yellow", high = "darkolivegreen3")+
  labs(title = "Heatmap of mega_snack all evens",
       x = "Lentils",
       y = "No. of colors",
       color = "all_even") +
        geom_text(aes(label = all_even * 100), color = "black", size = 4)+
  theme_minimal()
```




```
mega_snack_1 %>%
  ggplot(aes(x = factor(round(n_unit, 1)), y = n_color, fill = low_color))+
  geom_tile(color = "white") +
    geom_text(aes(label = sprintf("%.2f%%", low_color * 100)), color = "black", size = 4)+
  scale_fill_gradient(low = "darkolivegreen3", high = "red3",labels = scales::percent)+
  labs(title = "Chance of Less Than 66% Color in Package",
       x = "Lentils",
       y = "No. of colors",
       fill = "low color")+
  theme_minimal()
```


Chance of Less Than 66% Color in Package
No. of colors
Lentils
27.5
49.5
164.8
274.7
362.6
549.5
824.2
1098.9
8
6
4
2
53.00% 41.40% 6.60% 2.80% 1.80% 0.40% 0.00% 0.00%
40.60% 26.40% 7.20% 4.20% 0.60% 0.00% 0.00% 0.00%
54.80% 30.00% 3.80% 0.60% 0.20% 0.00% 0.00% 0.00%
30.00% 21.00% 2.00% 0.80% 0.00% 0.00% 0.00% 0.00%
30.20% 18.00% 0.80% 0.00% 0.00% 0.00% 0.00% 0.00%
27.00% 5.00% 0.20% 0.00% 0.00% 0.00% 0.00% 0.00%
9.40% 1.80% 0.00% 0.00% 0.00% 0.00% 0.00% 0.00%
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.

Conclusions

Data Structure

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