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,] 27 13 16 16 14 14
```

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

```
##       Red Blue Green Orange Yellow Brown
## Bag_1   3    2     3      2      0     1
## Bag_2   2    3     3      1      1     1
## Bag_3   2    0     2      1      2     3
```

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

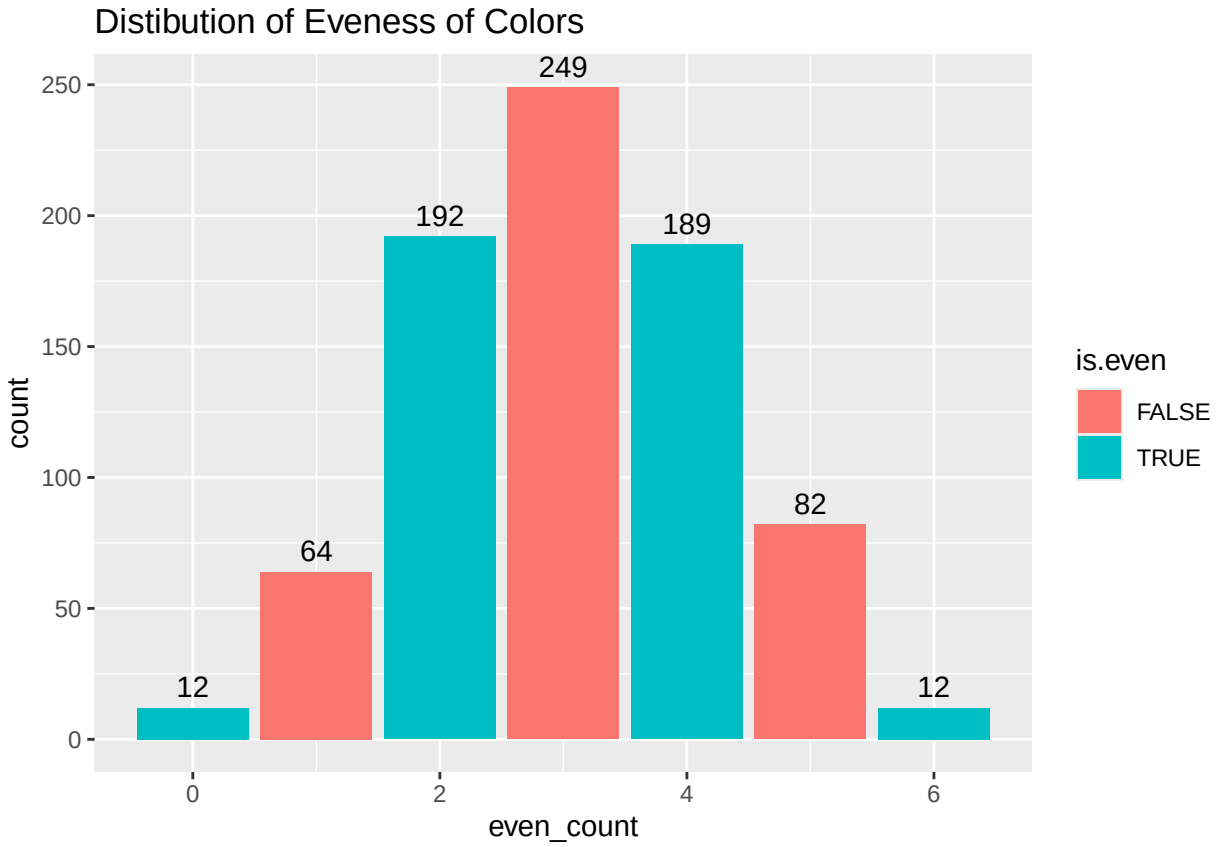
here are the first rows:

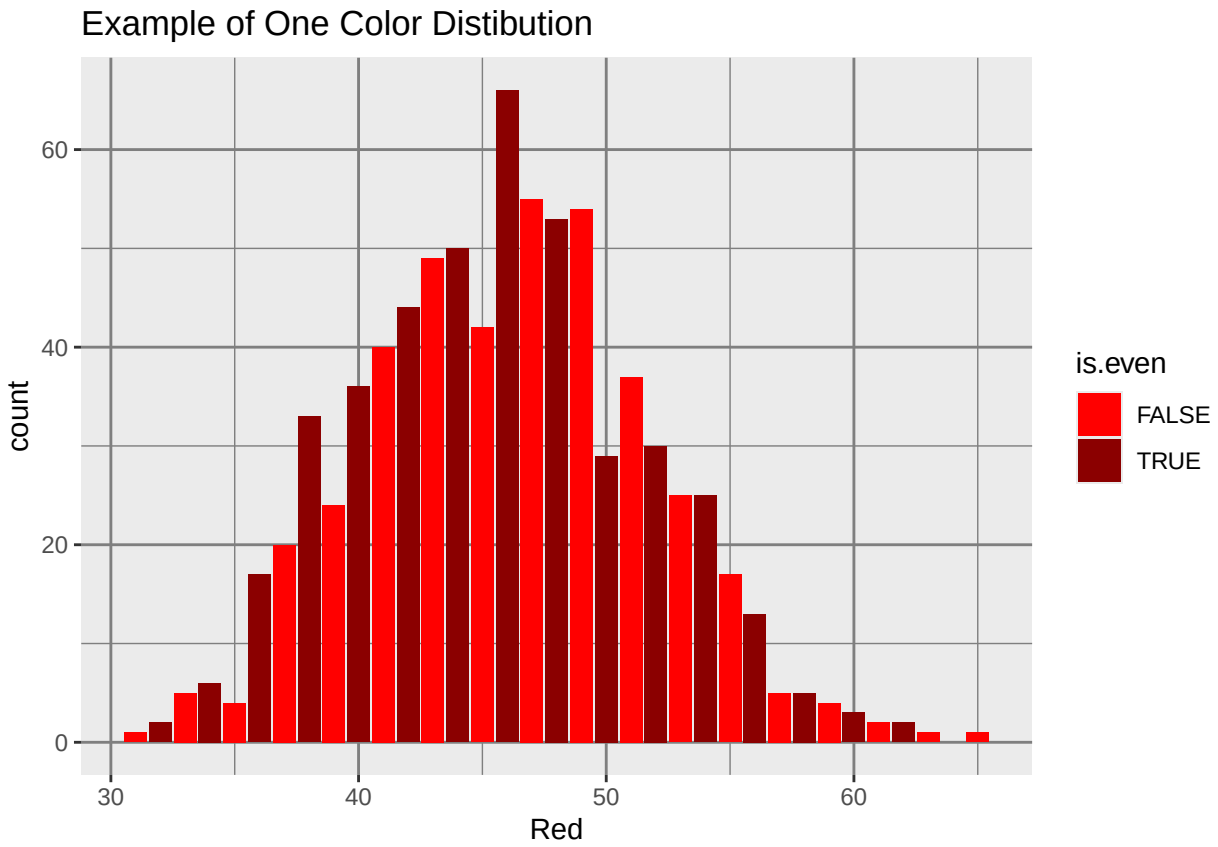
```
## # A tibble: 6 x 11
##     Red  Blue Green Orange Yellow Brown even_count even_evens low_col Variance
##   <int> <int> <int>  <int>  <int> <int>      <dbl> <lgl>        <dbl>    <dbl>
## 1    49    44    45     46     48    42          4 TRUE             0     6.67
## 2    57    43    52     45     33    45          1 FALSE            0    67.4
## 3    52    43    42     45     45    48          3 FALSE            0    13.4
## 4    48    51    48     31     48    49          3 FALSE            0    54.2
## 5    40    36    45     56     42    55          4 TRUE             0    66.7
## 6    46    50    47     39     30    62          4 TRUE             1   115.
## # 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       31      42     46 45.85250      49   65 32.55143
## Blue      29      42     46 46.11125      50   67 39.32553
## Green     25      41     45 45.25375      49   65 39.52126
## Orange    24      41     46 45.70625      50   68 38.09258
## Yellow    27      42     46 45.93000      50   64 37.57207
## Brown     27      41     46 45.64000      49   65 36.72380
```





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

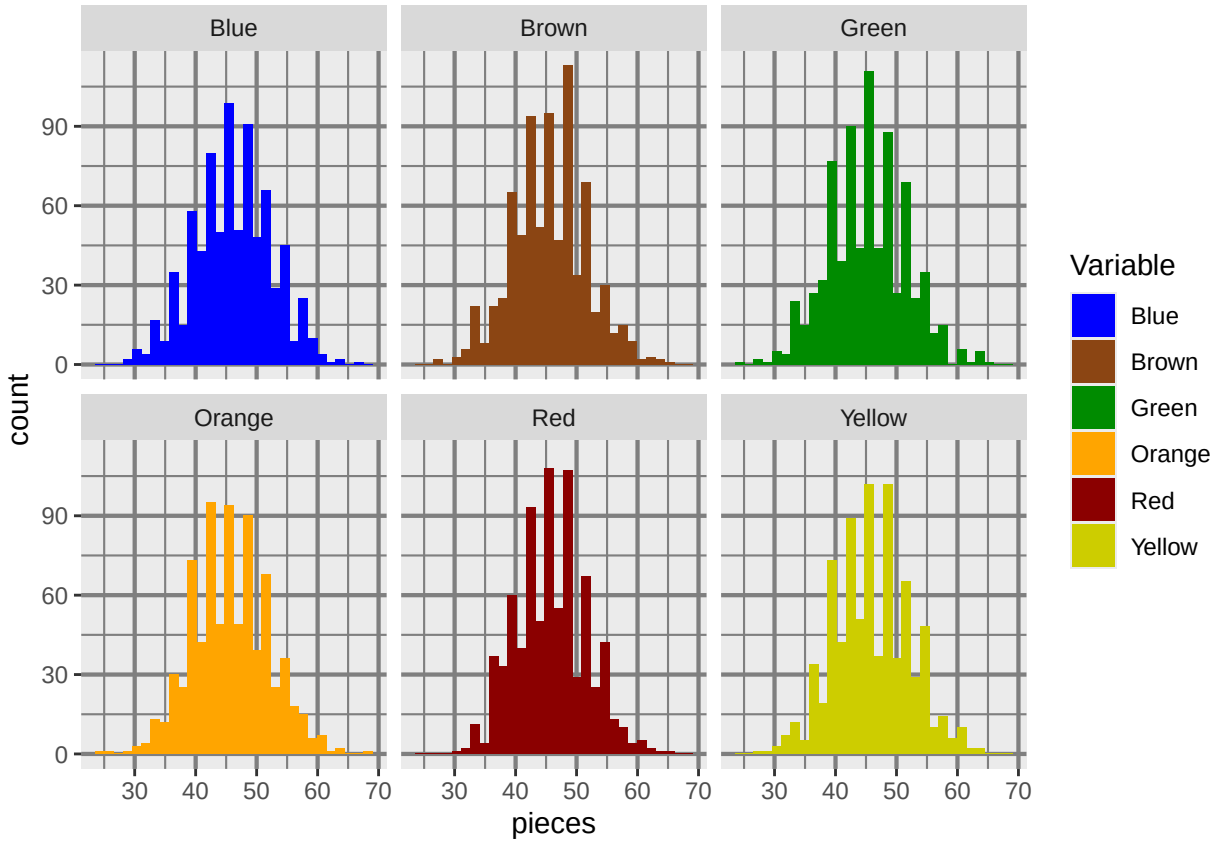
```
##     Red    Blue   Green  Orange  Yellow   Brown
## "74.8%" "14.5%"  "1.7%" "71.0%" "51.1%" "49.1%"
```

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`.
```



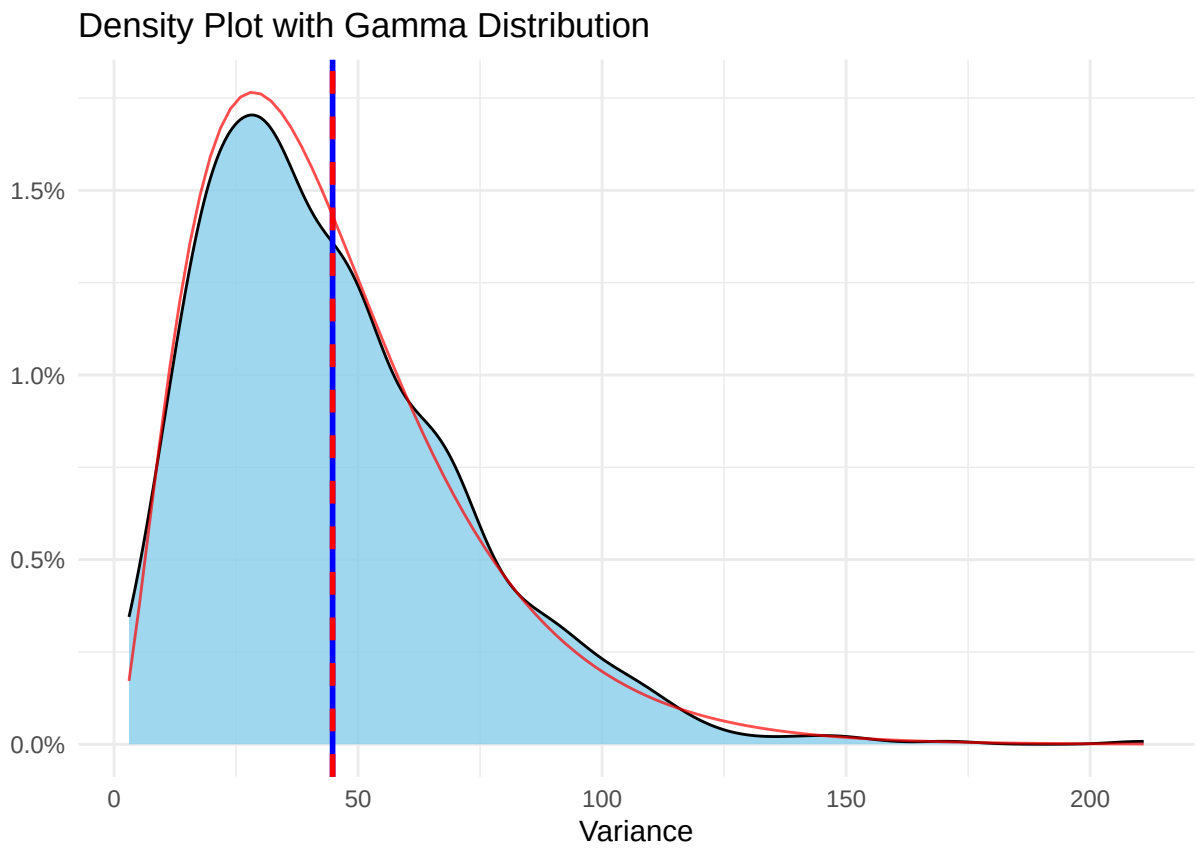
Variance Distribution Checking

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 2.745 and rate 0.061"
```



#use statistics to sample better low chance cases

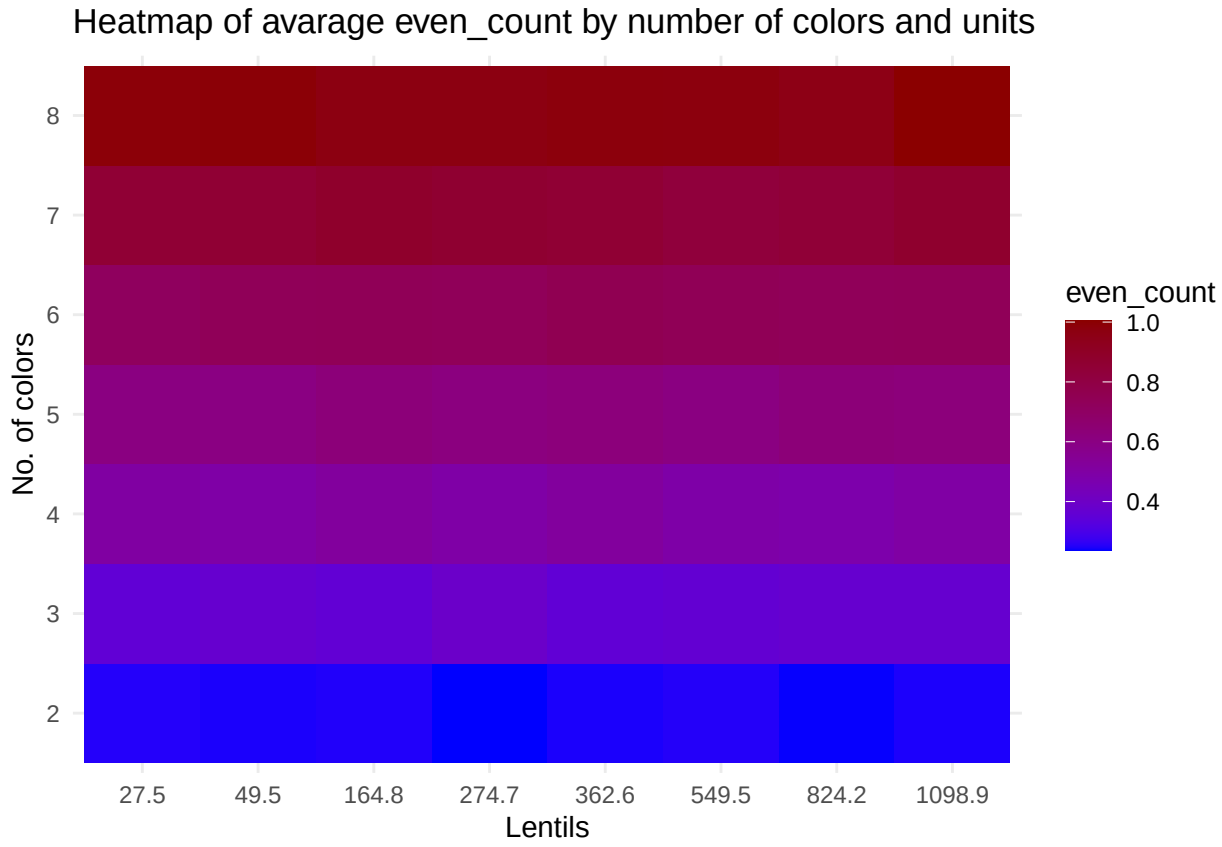
n*m types of snacks

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   27.5       5  0.6063024  0.1258500   5.444    0.038     0.340            0
## 2   27.5       8  0.9890024  0.1096714   3.187    0.032     0.544            0
## 3  274.7       2  0.2374000  0.1196786 133.245    0.032     0.000          110
## 4  824.2       2  0.2389762  0.1304405 444.128    0.034     0.000          364
## 5  164.8       8  0.9672786  0.1228119  21.022    0.038     0.098           10
```

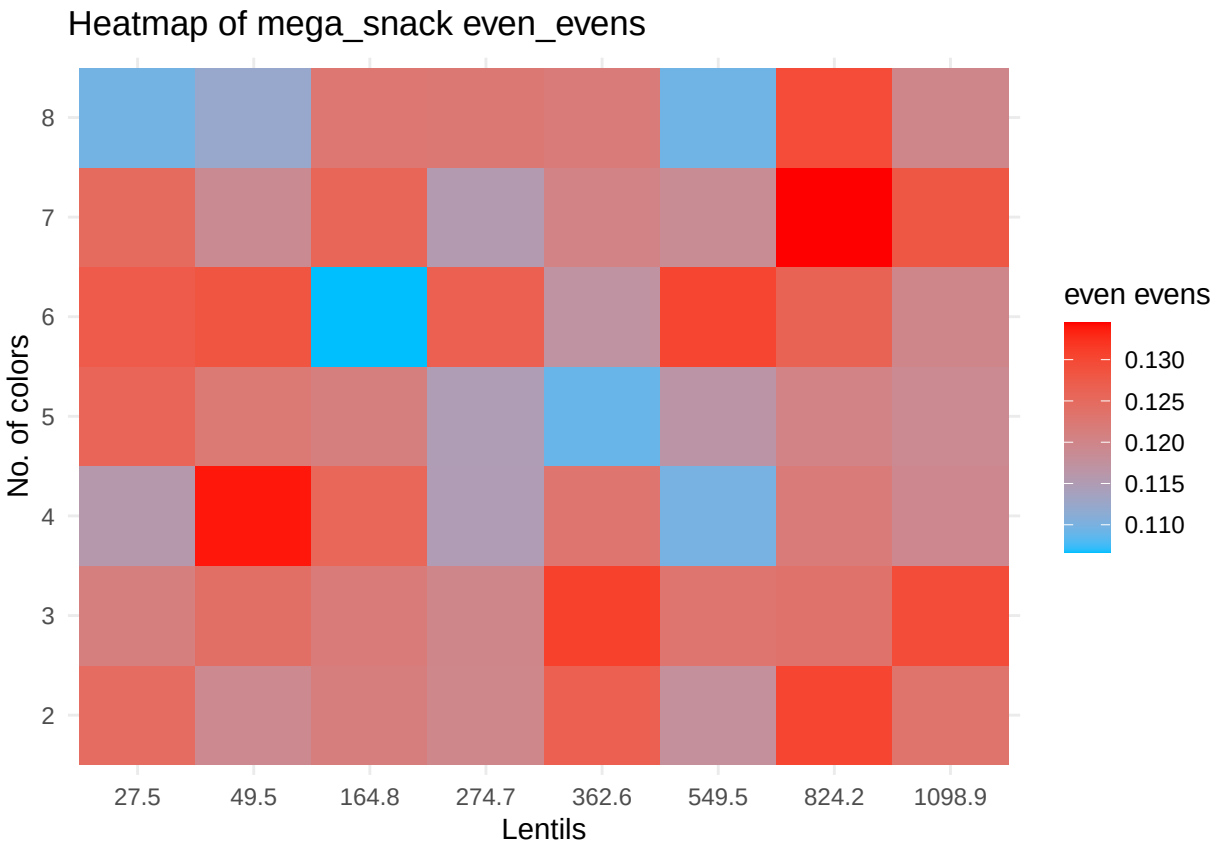



```
##
## Call:
## lm(formula = mega_snack_2$even_count ~ mega_snack_2$n_color +
##     mega_snack_2$n_unit + mega_snack_2$color_No2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.53105 -0.14581  0.00614  0.15720  0.57393
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 4.490e-01  2.659e-02  16.884  < 2e-16 ***
## mega_snack_2$n_color        1.314e-02  4.213e-03   3.119  0.00188 **
## mega_snack_2$n_unit        -5.611e-05  2.358e-05  -2.380  0.01754 *
## mega_snack_2$color_No2TRUE  1.248e-02  1.703e-02   0.733  0.46376
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2442 on 836 degrees of freedom
## Multiple R-squared:  0.0187, Adjusted R-squared:  0.01518
## F-statistic:  5.31 on 3 and 836 DF,  p-value: 0.001253

##
## 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
## -0.53105 -0.14581  0.00614  0.15720  0.57393
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 4.490e-01  2.659e-02  16.884  < 2e-16 ***
## mega_snack_2$n_color        1.314e-02  4.213e-03   3.119  0.00188 **
## mega_snack_2$n_unit        -5.611e-05  2.358e-05  -2.380  0.01754 *
## mega_snack_2$color_No2TRUE  1.248e-02  1.703e-02   0.733  0.46376
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2442 on 836 degrees of freedom
## Multiple R-squared:  0.0187, Adjusted R-squared:  0.01518
## F-statistic:  5.31 on 3 and 836 DF,  p-value: 0.001253
```

```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()
```



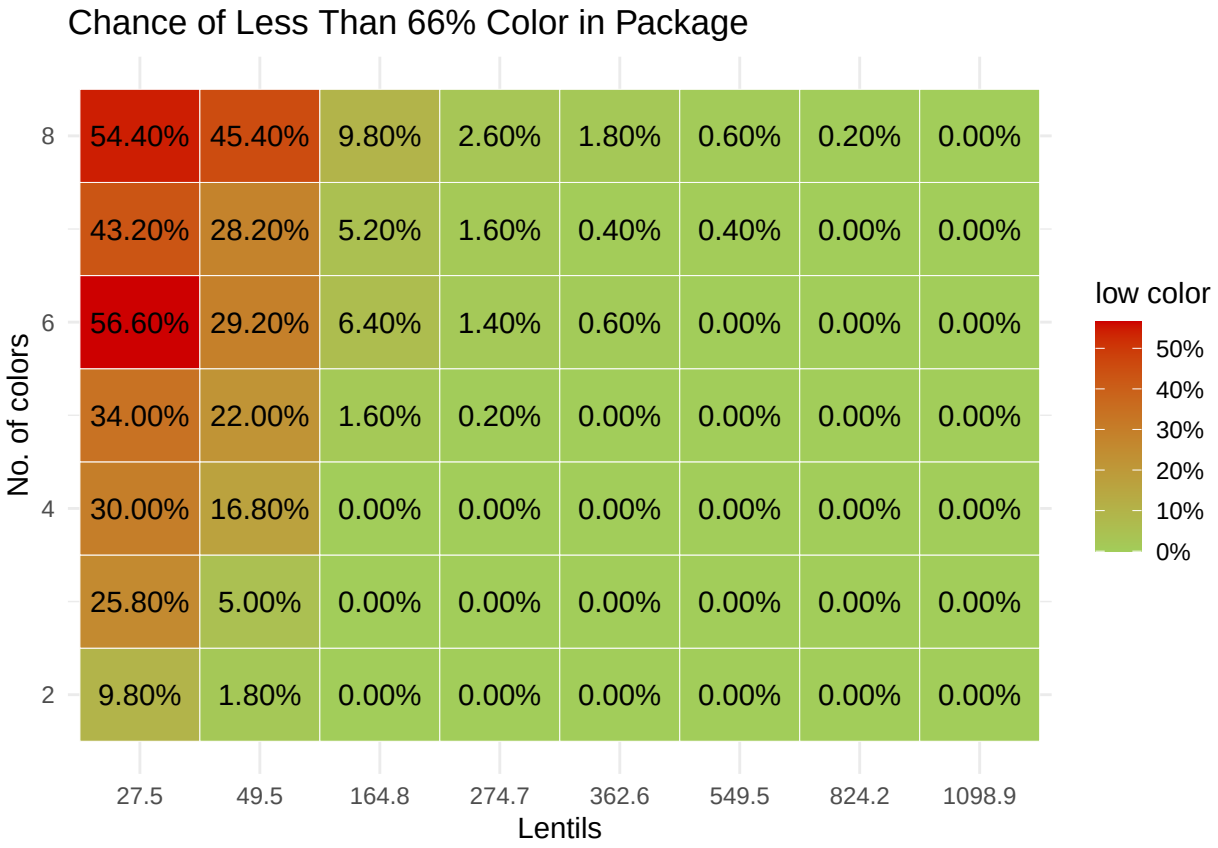
now let us see the probability of all even, and whether there is pattern.

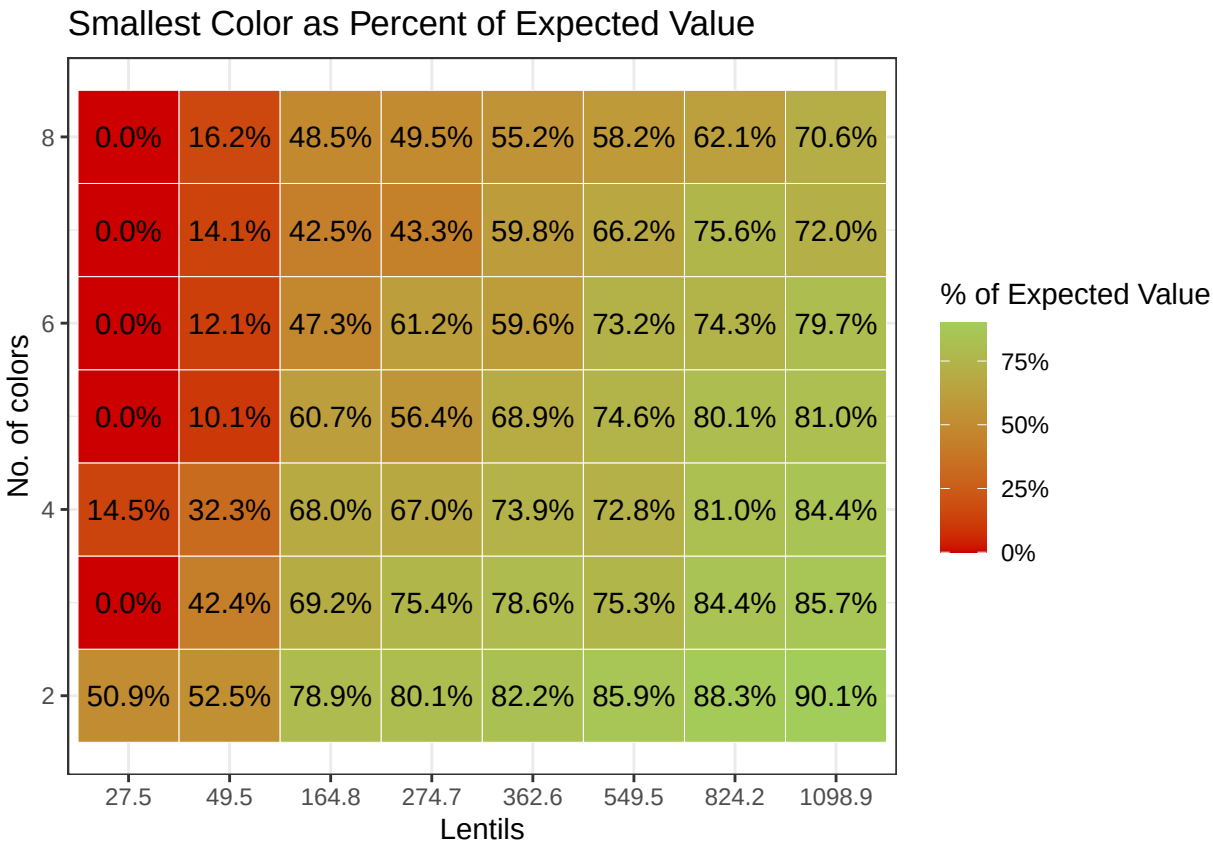
```
Heatmap_all_evens<-
  mega_snack_1 %>%
  ggplot(aes(x = as.factor(round( n_unit,1) ), y = n_color , color = all_even,size = 8 )) +
  geom_point() +
  scale_color_gradient(low = "yellow", high = "darkolivegreen3")+
  labs(title = "Heatmap of all evens",
       x = "Lentils",
       y = "No. of colors",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

ggsave(Heatmap_all_evens,
       filename = "Heatmap_all_evens.png",
       height = 4, width = 5, units = "in")
```

```
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()
```





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

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