My M&M OCD

Yoni

05 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?

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 (represented as factorial numbers), and see the results for many packages as a statistic data.

Parameters

The basic parameters (will be changed later):

```
#parameters
nn<- 500            #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 Sample

General Sample

create_bag is a function to create one snack package as matrix.

sample_MnM is a function to create n bags from the create_bag function.

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

##       1  2  3  4  5  6
## [1,] 18 20 22 14 15 11

## [1] "3 bags:"

##       1 2 3 4 5 6
## Bag_1 1 3 1 3 2 1
## Bag_2 3 1 1 2 2 1
## Bag_3 0 4 1 1 2 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-
6. min- the lowest color in each row

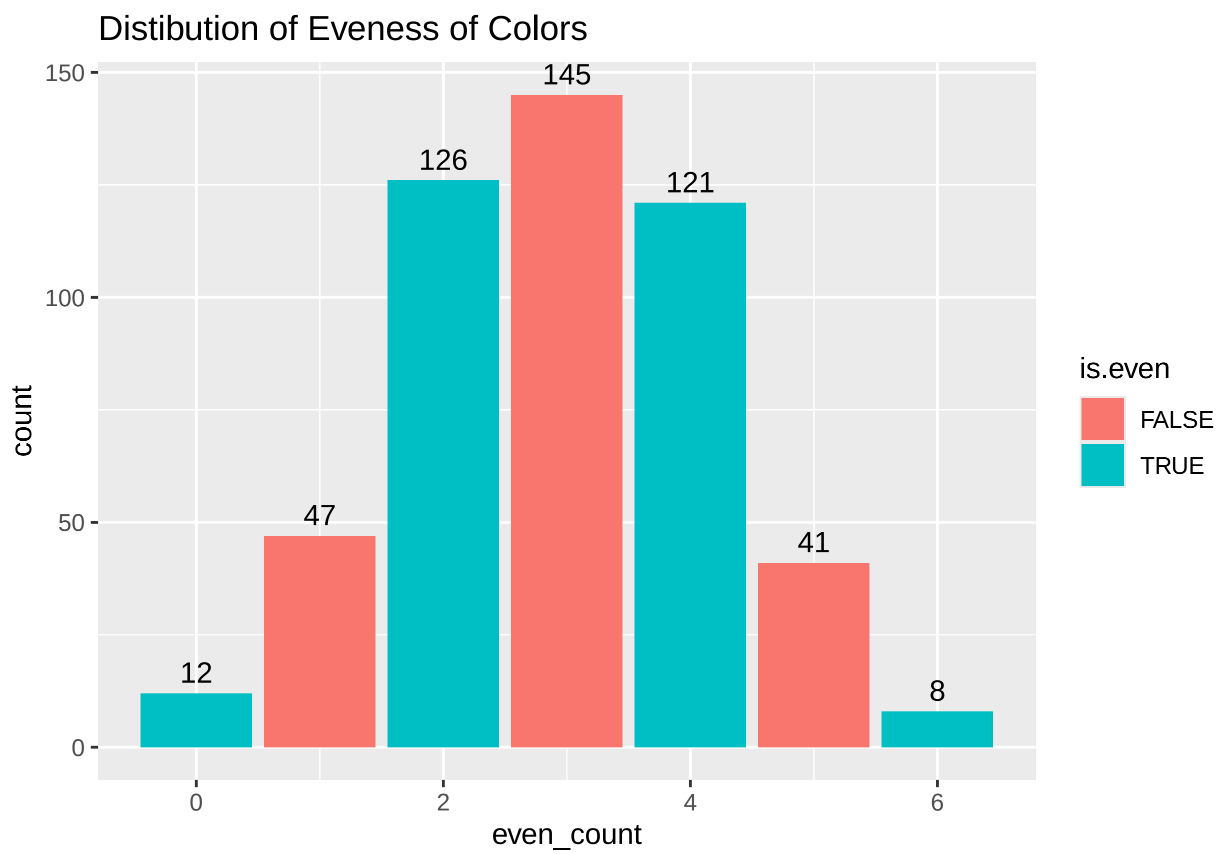
here are the first rows:

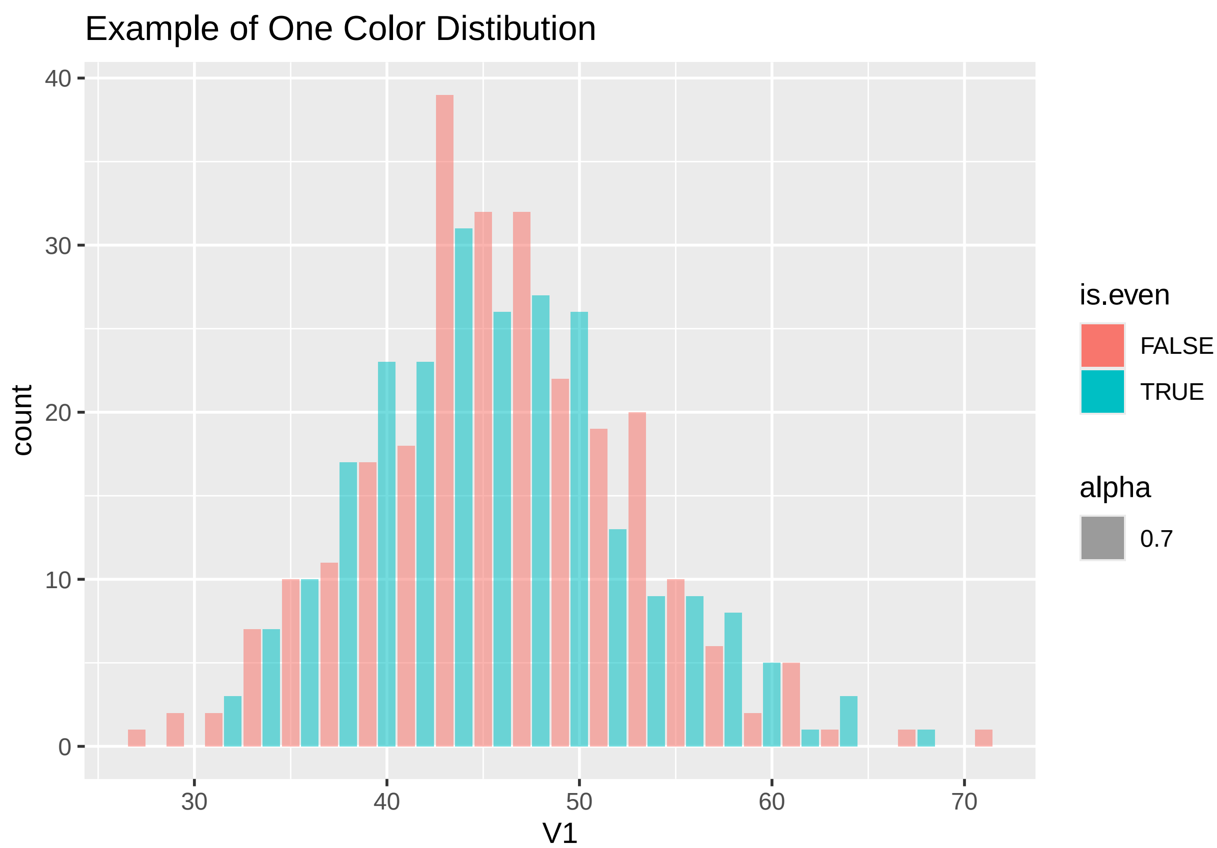
```
## # A tibble: 6 x 11
##      V1    V2    V3    V4    V5    V6 even_count even_evens Variance low_col
##   <int> <int> <int> <int> <int> <int>      <dbl> <lgl>         <dbl>   <dbl>
## 1    53    41    43    47    55    36          1 FALSE          53.0       0
## 2    46    36    45    53    47    48          3 FALSE          31.0       0
## 3    43    60    34    48    41    49          3 FALSE          77.4       0
## 4    44    50    48    38    50    44          6 TRUE           21.5       0
## 5    37    57    56    47    39    39          1 FALSE          80.2       0
## 6    43    50    47    39    47    49          1 FALSE          17.0       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   27      41     45 45.828      50   71 47.45332
## V2   30      42     46 45.988      50   64 36.50487
## V3   25      42     46 46.092      50   68 40.29212
## V4   27      41     45 45.154      49   68 39.38105
## V5   30      42     46 45.734      50   64 35.57840
## V6   29      42     46 45.670      49   63 35.40792
```


Distibution of Eveness of Colors
150
100
50
0
count
12
47
126
145
121
41
8
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.

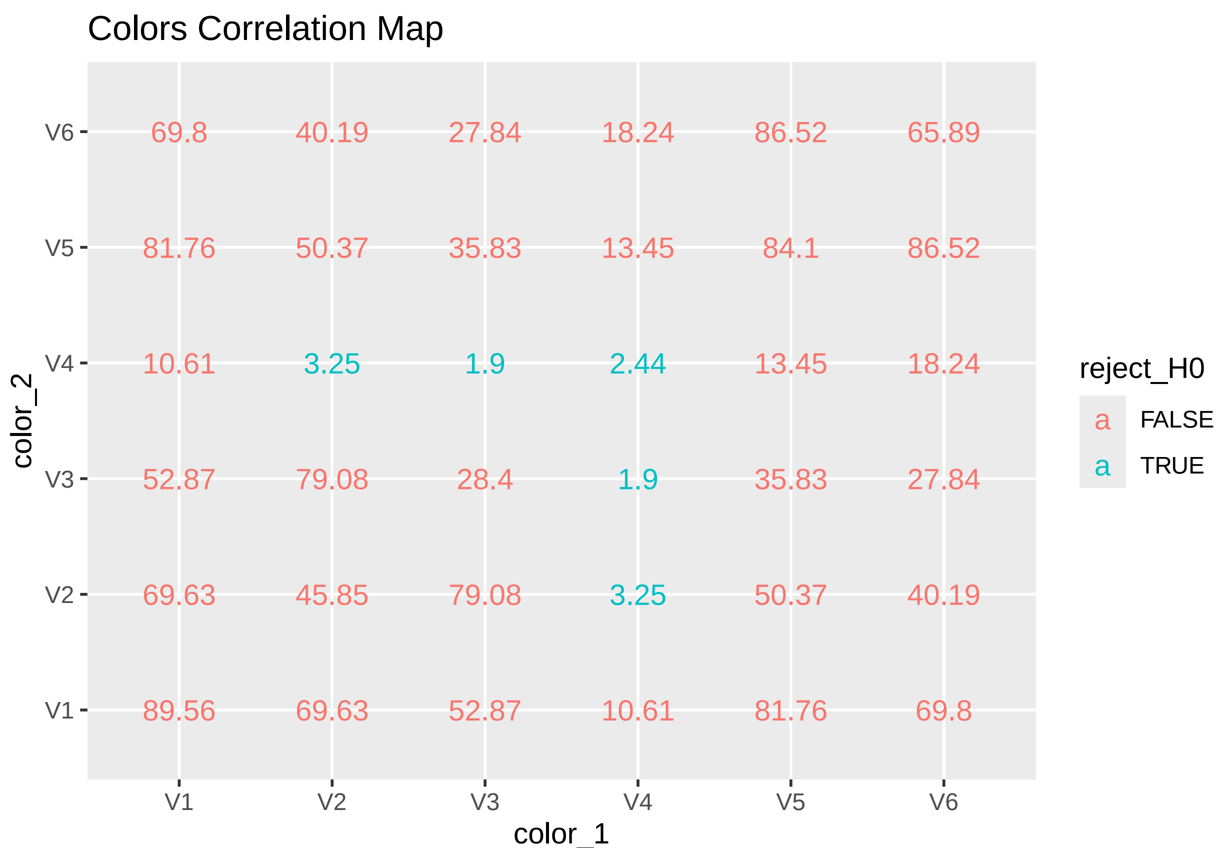
Here is the result, none of them bellow 5% P. value

```
##       V1       V2       V3       V4       V5       V6
## "89.6%" "45.9%" "28.4%"  "2.4%" "84.1%" "65.9%"
```

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

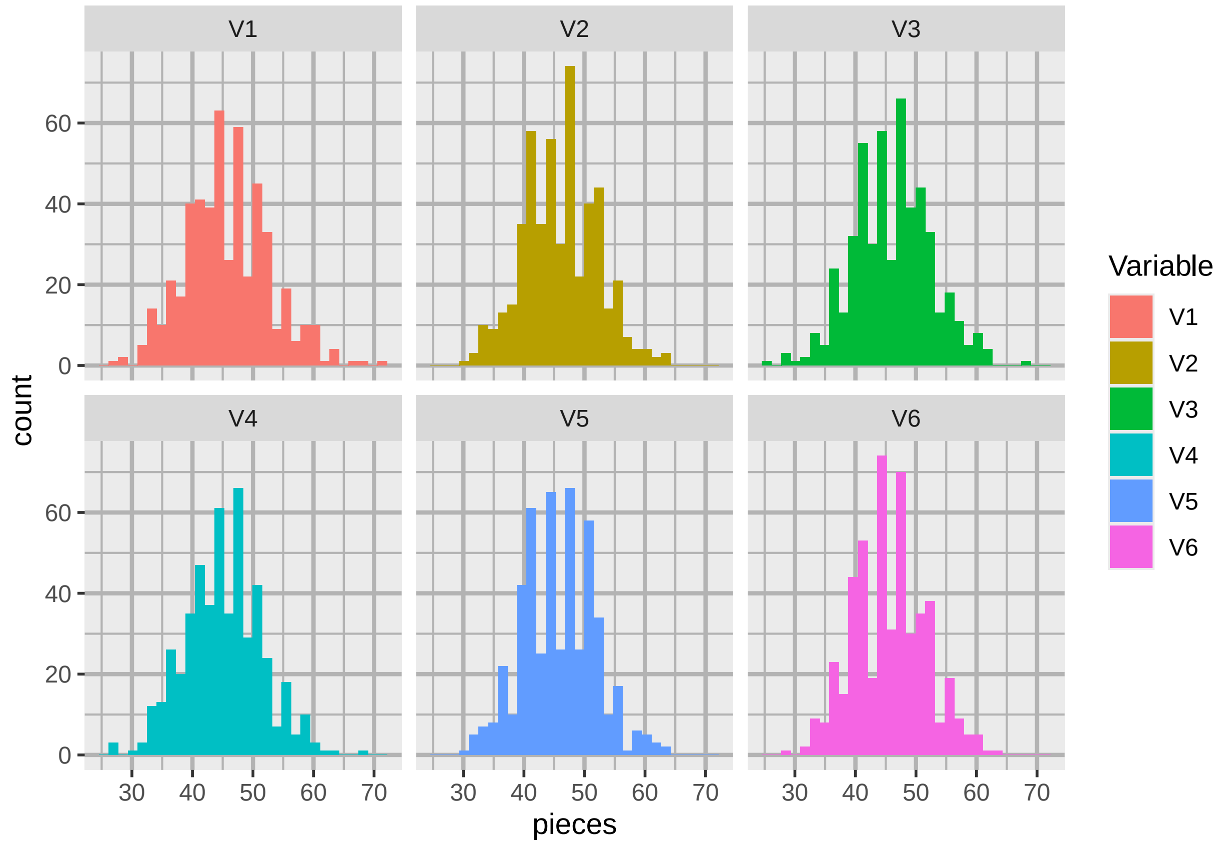
for each row i and column j, 1) if i==j, this it the check from before of the expected value to n_unit/n_color 2) if i!=j, this is two samples test of same expected value hypothesis

```
##        V1     V2     V3     V4     V5     V6
## V1 0.8956 0.6963 0.5287 0.1061 0.8176 0.6980
## V2 0.6963 0.4585 0.7908 0.0325 0.5037 0.4019
## V3 0.5287 0.7908 0.2840 0.0190 0.3583 0.2784
## V4 0.1061 0.0325 0.0190 0.0244 0.1345 0.1824
## V5 0.8176 0.5037 0.3583 0.1345 0.8410 0.8652
## V6 0.6980 0.4019 0.2784 0.1824 0.8652 0.6589
```



now here Is visualization of the actual data per color:

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



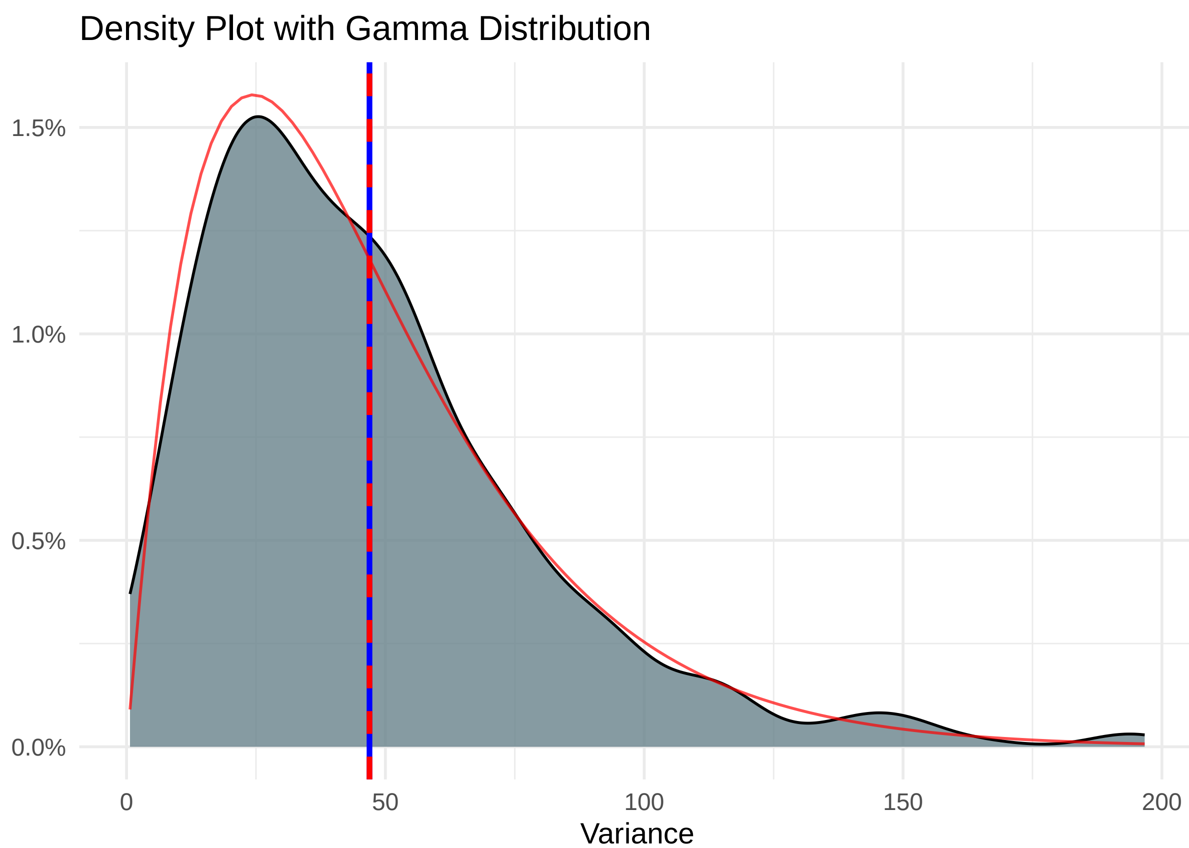
Variance Distribution Checking

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 2.088 and rate 0.044"
```



#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

```
mega_snack<- function(nn,n_unit,n_color)
  {
  m_sample<- length(n_unit)*length(n_color)
  nul_mat= matrix(nrow = m_sample, ncol = 6)
  res<- cbind(rep(n_unit,length(n_color)),sort(rep(n_color,length(n_unit))),
                                              nul_mat)
  for (i in 1:(dim(res)[1]))
  {
    #print(c(res[i,1],res[i,2]))
    low_color<- 0.666*res[i,1]/(res[i,2])
    small_sample<- sample_MnM(nn,res[i,1],res[i,2])
    small_sample<-
      small_sample %>% as_data_frame() %>%
      mutate(even_count= rowSums(across(everything()  , ~ .x %% 2 == 0))/n_color,
             #how many evens colors there are
             even_evens= (rowSums(across(c(1:n_color)  , ~ .x %% 2 == 1)) %% 2 ==0)/n_color,
             #are the uneven colors even
             var_col=    apply(across(c(1:n_color)), 1, var),
```

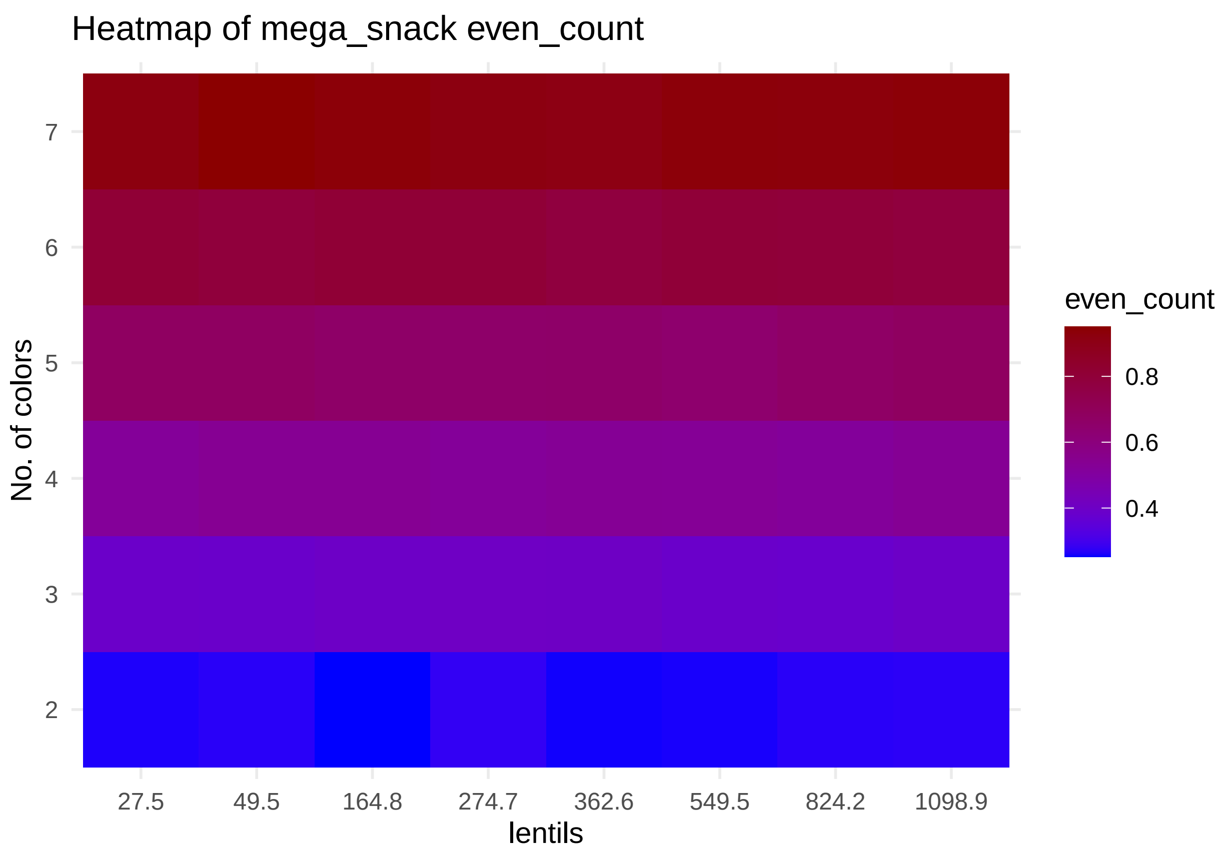
```
              #var of candy per color/ type
              min=        apply(across(c(1:n_color)), 1, min),
              #lowest value in color
              all_even=   rowSums(across(c(1:n_color)  , ~ .x %% 2 == 0))== n_color,
              low_col=    rowSums(across(c(1:n_color), ~ .x <= low_color ))>=1
             )
    res[i,3]<- mean(small_sample$even_count)
    res[i,4]<- mean(small_sample$even_evens)
    res[i,5]<- mean(small_sample$var_col)
    res[i,6]<- mean(small_sample$all_even)
    res[i,7]<- mean(small_sample$low_col)
    res[i,8]<- min(small_sample$min)
      }
  colnames(res)<- c("n_unit", "n_color", "even_count", "even_evens",
                    "var_col","all_even","low_color","smallest_col")
  res %>% as.data.frame()
  }
```

```
color_op<- 2:7
grams_op<- c(25,45,150,250,330,500,750,1000)
n_unit_op<- grams_op/gram
nn=900
```

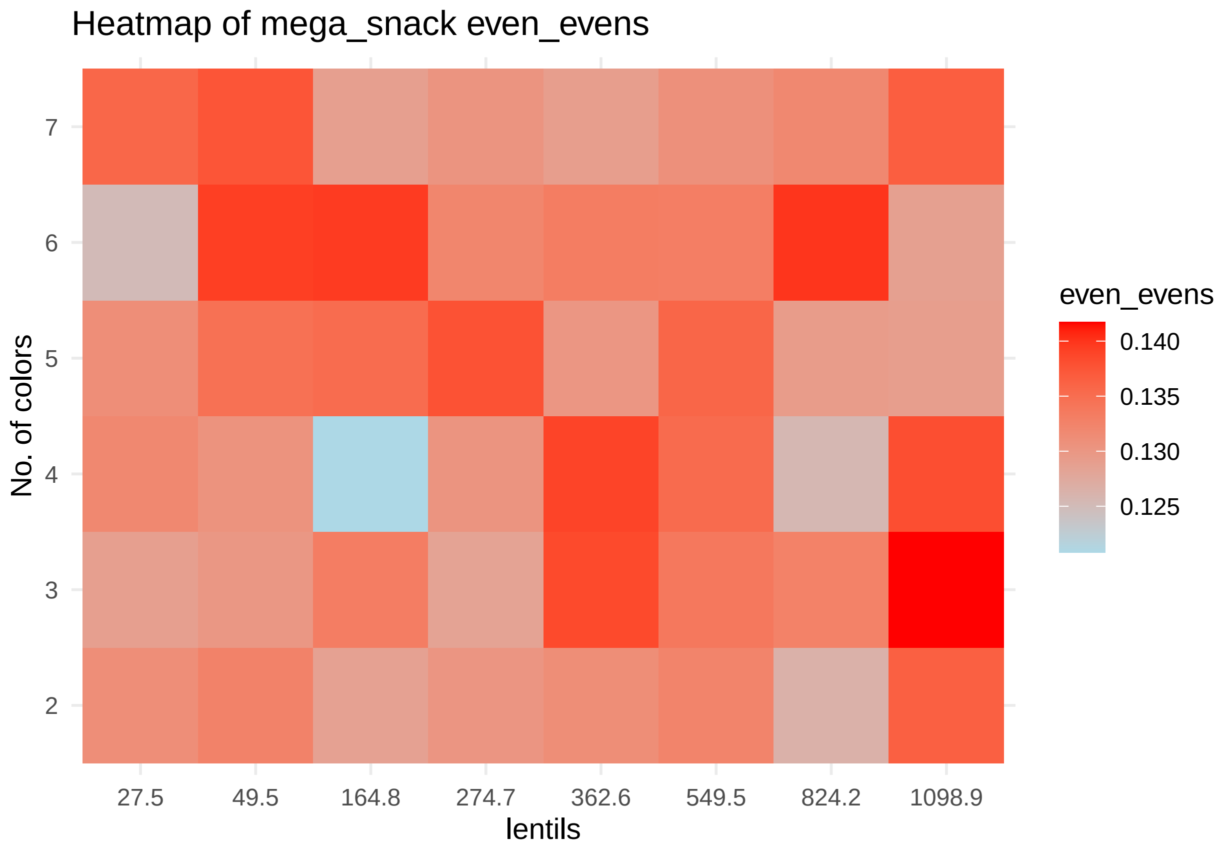
We will make the multiple sample. Here is some random rows:

```
##   n_unit n_color even_count even_evens   var_col   all_even  low_color
## 1   49.5       4  0.5364312  0.1305794  12.13389 0.04111111 0.18222222
## 2  824.2       7  0.9307143  0.1318968 116.43000 0.05111111 0.00000000
## 3  274.7       4  0.5216270  0.1303810  73.98333 0.04333333 0.00000000
## 4 1098.9       6  0.7863122  0.1287937 165.55833 0.03444444 0.00000000
## 5   27.5       2  0.2631746  0.1311799  12.71111 0.03777778 0.08111111
##   smallest_col
## 1            3
## 2           85
## 3           47
## 4          131
## 5            5
```

```
mega_snack_1 %>%
  ggplot(aes(x = factor((round( n_unit,1) )), y = factor(n_color ), fill = even_count )) +
  geom_tile() +
  scale_fill_gradient(low = "blue", high = "red4")+
  labs(title = "Heatmap of mega_snack even_count",
       x = "lentils",
       y = "No. of colors",
       fill = "even_count") +
  theme_minimal()
```

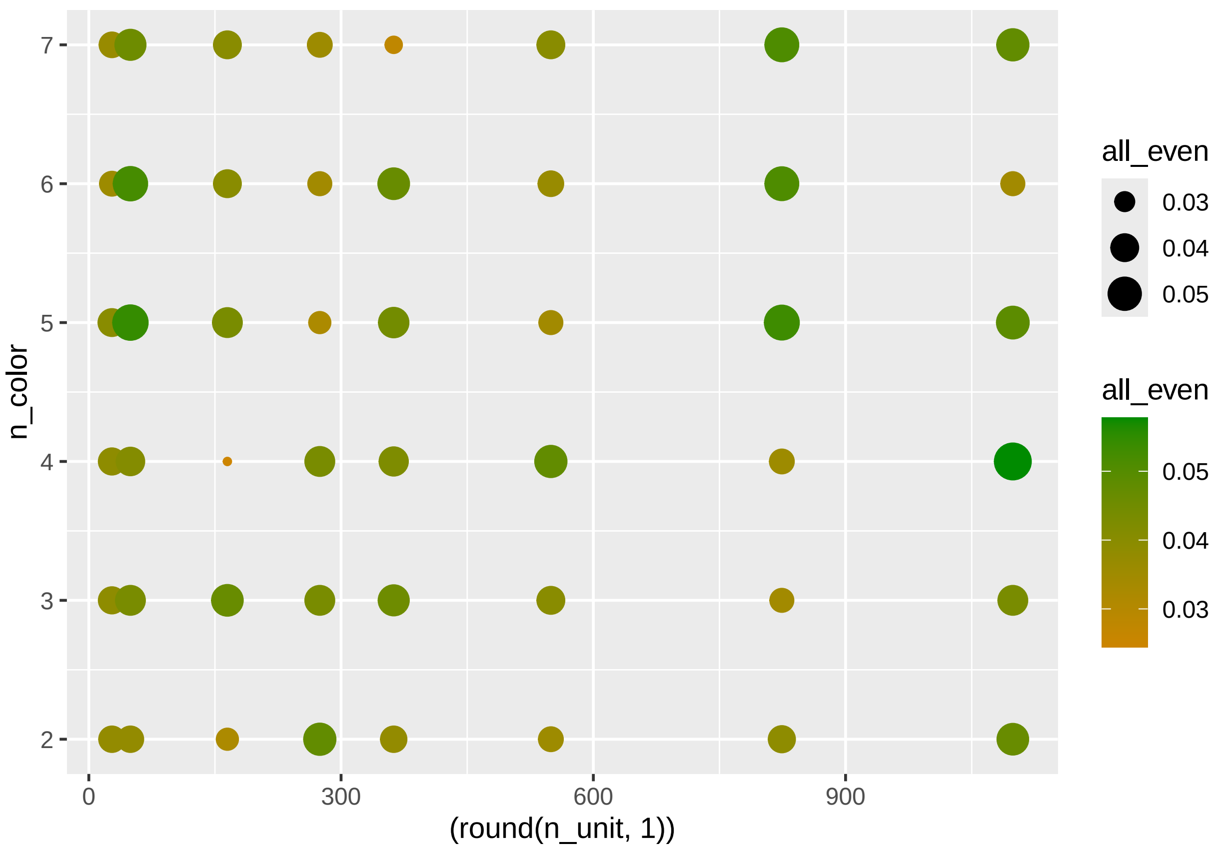



```r
mega_snack_1 %>%
  ggplot(aes(x = factor((round( n_unit,1) )), y = factor(n_color ), fill = even_evens )) +
  geom_tile() +
  scale_fill_gradient(low = "lightblue", high = "red")+
  labs(title = "Heatmap of mega_snack even_evens",
       x = "lentils",
       y = "No. of colors",
       fill = "even_evens")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



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

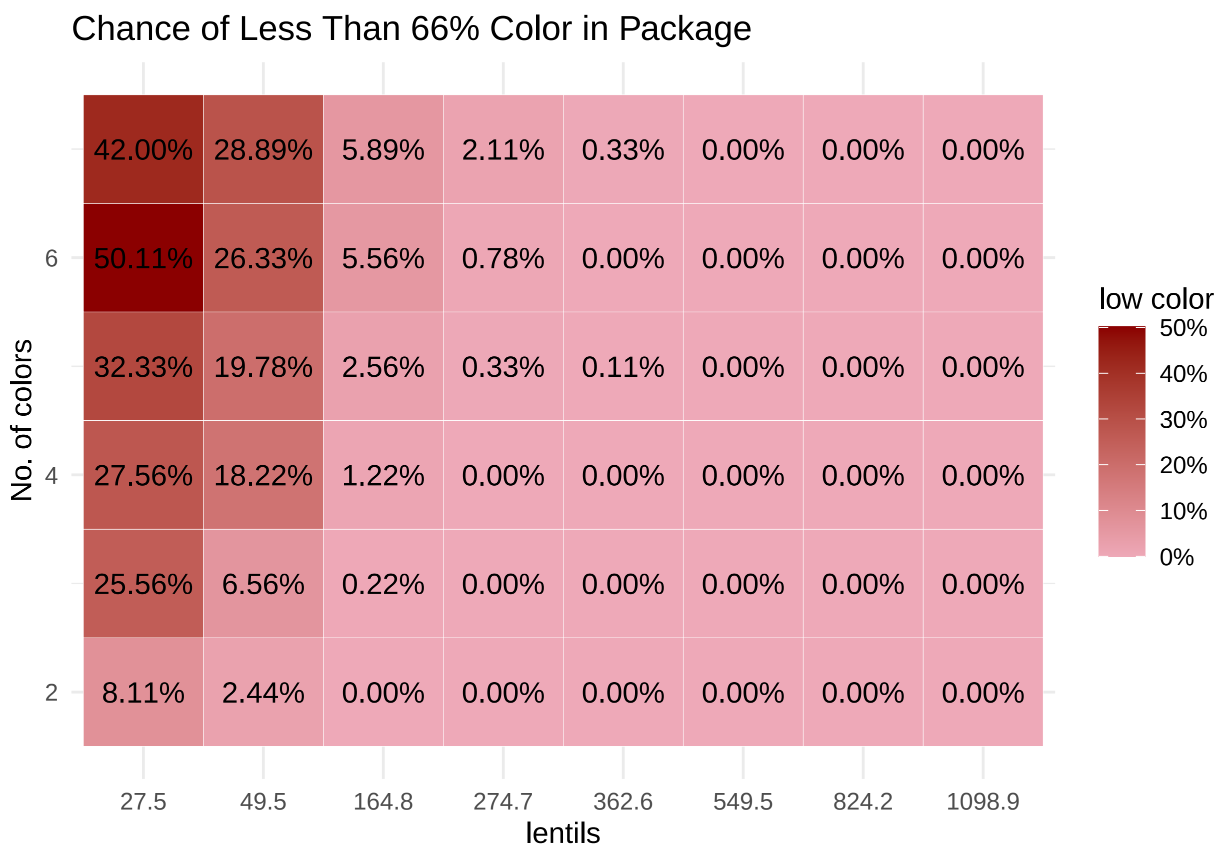
```
mega_snack_1 %>%
  ggplot(aes(x = (round( n_unit,1) ), y = n_color , color = all_even,size = all_even )) +
  geom_point() +
  scale_color_gradient(low = "orange3", high = "green4")
```

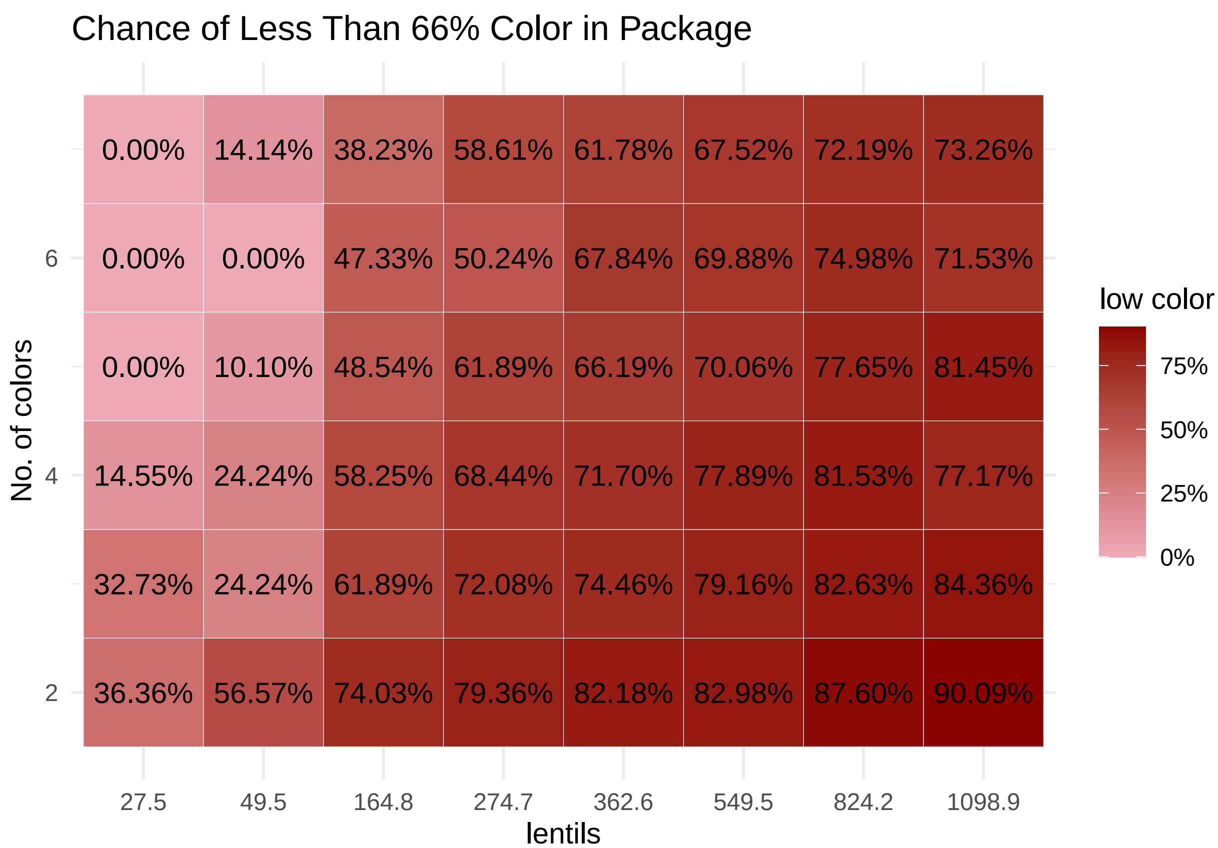



```
  labs(title = "Heatmap of mega_snack all evens",
       x = "lentils",
       y = "No. of colors",
       color = "all_even") +
  theme_minimal()
```

```
## NULL
```

```
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 = "pink2", high = "red4",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
42.00% 28.89% 5.89% 2.11% 0.33% 0.00% 0.00% 0.00%
50.11% 26.33% 5.56% 0.78% 0.00% 0.00% 0.00% 0.00%
32.33% 19.78% 2.56% 0.33% 0.11% 0.00% 0.00% 0.00%
27.56% 18.22% 1.22% 0.00% 0.00% 0.00% 0.00% 0.00%
25.56% 6.56% 0.22% 0.00% 0.00% 0.00% 0.00% 0.00%
8.11% 2.44% 0.00% 0.00% 0.00% 0.00% 0.00% 0.00%
No. of colors
6
4
2
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.