ECON 220 Lab
(Week 3)

Introduction to Python (Part II)

Justin S. Eloriaga

Outline

- Data Collection and Input
- Generating Variables
- Loops

(Fun Size) M&Ms

- I have an unhealthy obsession with sweets. I hope I don't get diabetes soon
- One of the things I love are M&M's!
  - I can eat the whole Costco tin in one sitting (send help…)
  - But for this class, let us use the Fun Size packs to get our point across.
- Each M&M pack can have at most 6 colors

Image obtained from Walgreens

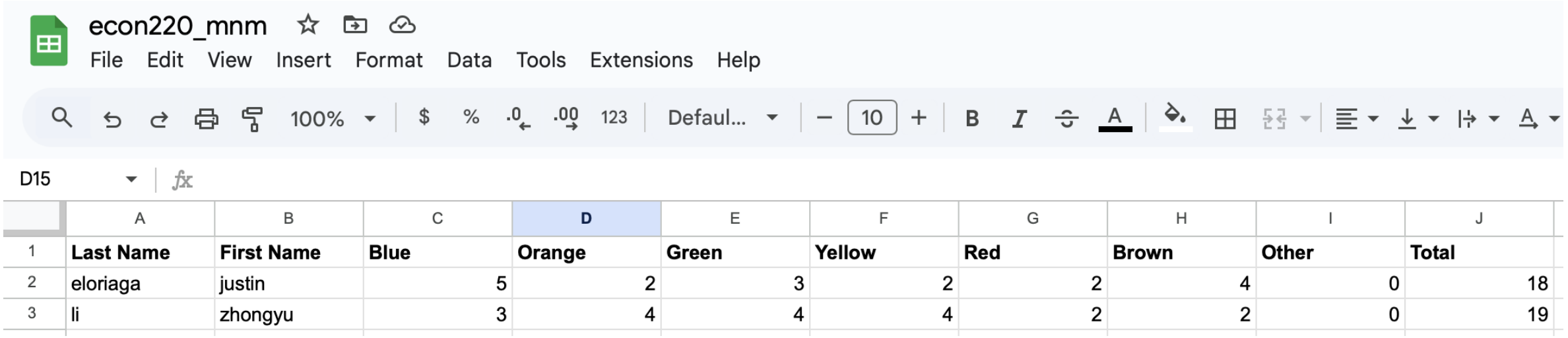
According to Google...

- **Plain M&Ms:** 24% blue, 20% orange, 16% green, 14% yellow, 13% red, 13% brown

- Let us test this claim, and in the process, learn a couple of key things in making a dataset and designing a function and doing loops.
- Plus, you get a fun size m&m on me, lol

Building our M&M Dataset

- Go to the link on Canvas or click here: https://shorturl.at/Yuj9o



	A	B	C	D	E	F	G	H	I	J
1	**Last Name**	**First Name**	**Blue**	**Orange**	**Green**	**Yellow**	**Red**	**Brown**	**Other**	**Total**
2	eloriaga	justin	5	2	3	2	2	4	0	18
3	li	zhongyu	3	4	4	4	2	2	0	19

- For the next 3 minutes, count the number of **Blue**, **Orange**, **Green**, **Yellow**, **Red**, and **Brown** M&M's you got in the fun size pack. The sheet will automatically total it.

Generating Variables

- Suppose we want to generate a variable that represents the proportion of Blue m&m's per pack collected.

$$Blue\ M\&M\ Proportion = \frac{\#\ of\ Blue\ M\&M\ Per\ Pack}{Total\ M\&M\ Per\ Pack}$$

```
data['blue_prop'] = data['Blue'] / data['Total']
```

^What you used to create that variable

Name of newly created variable

Using a (for) Loop

- Suppose we want to create a similar proportion variable but now for every other color (including other) in our dataset. To do this quickly, we can use a **loop**!

```
colors = ['Blue', 'Orange', 'Green', 'Yellow', 'Red', 'Brown', 'Other']
for color in colors:
    data[color + '_prop'] = data[color] / data['Total']
```

- Steps. (1) Create a list called colors. (2) Start the loop with for XXXX in the list name. (3) Specify what you want to create (i.e. similar formula to the variable creation command.

Using a (for) Loop

	Lastname	Firstname	Blue	Orange	Green	Yellow	Red	Brown	Other	Total	Blue_prop	Orange_prop	Green_prop	Yellow_prop	Red_prop	Brown_prop	Other_prop
0	MKFHJ	NOWCP	5	2	5	0	3	0	3	18	0.277778	0.111111	0.277778	0.000000	0.166667	0.000000	0.166667
1	TFFEJ	ELPOT	4	3	3	2	1	5	0	18	0.222222	0.166667	0.166667	0.111111	0.055556	0.277778	0.000000
2	TCPZF	IOLRT	4	0	3	4	1	2	2	16	0.250000	0.000000	0.187500	0.250000	0.062500	0.125000	0.125000
3	TBIBN	WFRCH	2	4	2	4	0	2	4	18	0.111111	0.222222	0.111111	0.222222	0.000000	0.111111	0.222222
4	RJXIG	WLTCX	0	2	1	4	3	5	4	19	0.000000	0.105263	0.052632	0.210526	0.157895	0.263158	0.210526
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	JEBTM	DSQKW	2	3	0	5	4	4	1	19	0.105263	0.157895	0.000000	0.263158	0.210526	0.210526	0.052632
96	VWXNM	VAXKJ	1	0	4	3	3	0	5	16	0.062500	0.000000	0.250000	0.187500	0.187500	0.000000	0.312500
97	THFHV	XJSDM	2	3	3	1	0	4	4	17	0.117647	0.176471	0.176471	0.058824	0.000000	0.235294	0.235294
98	YAHAO	OWITZ	5	3	0	3	1	3	3	18	0.277778	0.166667	0.000000	0.166667	0.055556	0.166667	0.166667
99	DVZLI	DCTRX	1	2	2	4	5	4	1	19	0.052632	0.105263	0.105263	0.210526	0.263158	0.210526	0.052632

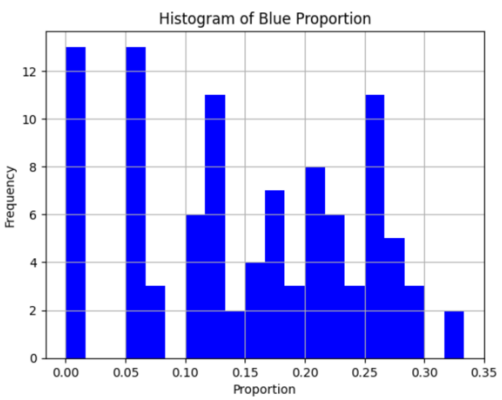
NEWLY CREATED VARIABLES which are all the proportions!

Generating a Histogram

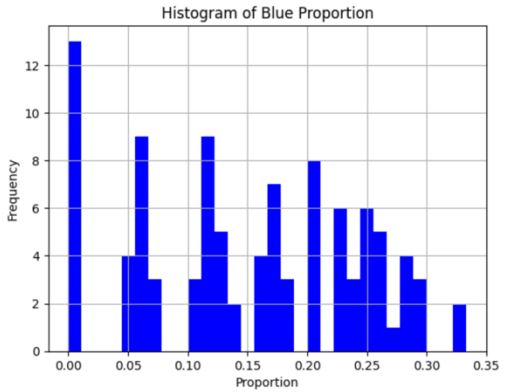
- Histograms are useful graphs for observing distributions.

```
plt.figure()
data['Blue_prop'].hist(bins=10, color='blue')
plt.title('Histogram of Blue Proportion')
plt.xlabel('Proportion')
plt.ylabel('Frequency')
plt.show()
```

The choice of bins matters!!

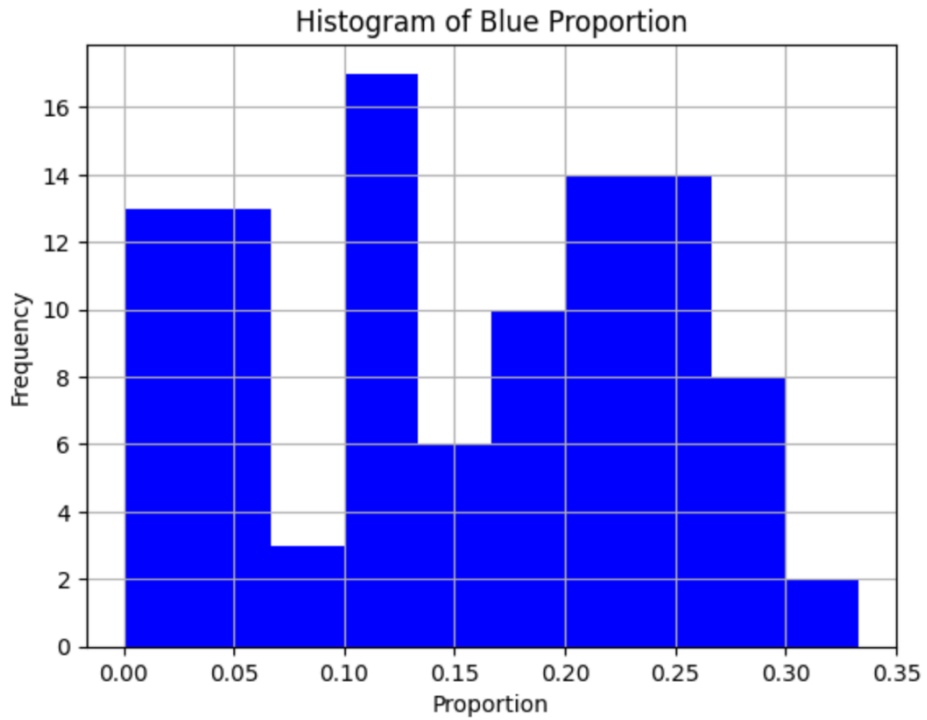


Bins = 20



Bins = 30

Bins = 10



Mean and Variance of the Proportions

```
mean_proportions = data[['Blue_prop', 'Orange_prop', 'Green_prop', 'Yellow_prop', 'Red_prop', 'Brown_prop', 'Other_prop']].mean()
mean_proportions
```

✓ 0.0s

```
Blue_prop      0.149946
Orange_prop    0.154594
Green_prop     0.150901
Yellow_prop    0.136330
Red_prop       0.122674
Brown_prop     0.143452
Other_prop     0.142102
dtype: float64
```

- **Plain M&Ms**: 24% blue, 20% orange, 16% green, 14% yellow, 13% red, 13% brown

So, what do we find?

```
variance_proportions = data[['Blue_prop', 'Orange_prop', 'Green_prop', 'Yellow_prop', 'Red_prop', 'Brown_prop', 'Other_prop']].var()
variance_proportions
```

✓ 0.0s

```
Blue_prop      0.008736
Orange_prop    0.009640
Green_prop     0.010006
Yellow_prop    0.009629
Red_prop       0.009744
Brown_prop     0.009180
Other_prop     0.010818
dtype: float64
```