Student homework, ggplot2

Dan Chitwood

April 3, 2016

This script is a collection of graphs created by the students of Bio5702, Current Approaches in Plant Research, a graduate course at Washington University in St. Louis. These graphs are explorations of the dataset "candy\_nutrition.txt", the nutrition label information from 75 different candy types. The point of this homework assignment was to become acquainted with R, basic statistics, data visualization, and ggplot2.

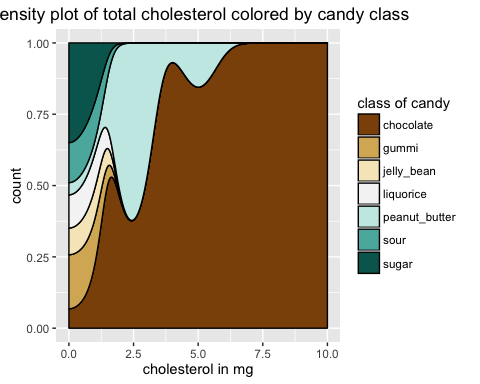
## Student #1

Begin homework here

library(ggplot2)  
data = read.table ("./candy\_nutrition.txt", header=T)

ggplot1, density plot of cholesterol content of different candy

d <- ggplot(data, aes(cholesterol\_mg, ..count.., fill = class))   
d + geom\_density(position = "fill") +  
 scale\_fill\_brewer(type="div", palette=1, name="class of candy") +  
 scale\_colour\_brewer(type="div", palette=1, name="class of candy") +  
 xlab(label="cholesterol in mg") +  
 ylab(label="count") +  
 ggtitle("Density plot of total cholesterol colored by candy class")

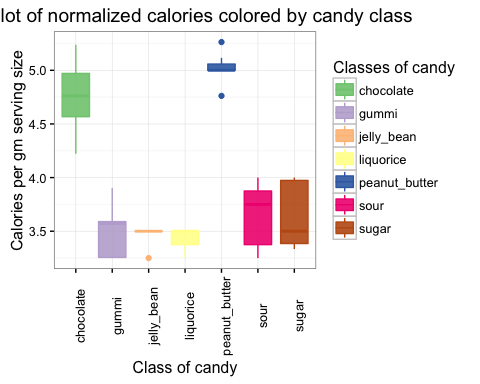


calories of candy is normalized to its serving\_size\_g

data$Normalized\_cal <- (data$calories/data$serving\_size\_g)

ggplot2, boxplot of normalized calories of different candy class

d <- ggplot(data, aes(x=class, y=Normalized\_cal, colour=class, fill=class))  
d + geom\_boxplot(alpha=0.9) +  
 scale\_fill\_brewer(type="qual", palette=1, name="Classes of candy") +  
 scale\_colour\_brewer(type="qual", palette=1, name="Classes of candy") +  
 xlab(label="Class of candy") +  
 ylab(label="Calories per gm serving size") +  
 ggtitle("Boxplot of normalized calories colored by candy class") +  
 theme\_bw() +  
 theme(axis.text.x=element\_text(angle=90))

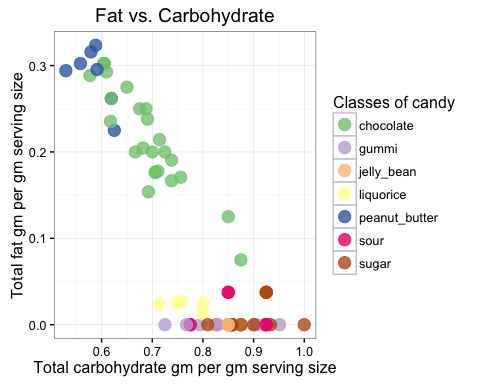


Normalized fat and carbohydrate

data$Normalized\_fat <- (data$total\_fat\_g/data$serving\_size\_g)  
data$Normalized\_carb <- (data$total\_carb\_g/data$serving\_size\_g)

ggplot3, scatterplot of total fat vs. total carbohydrate

d <- ggplot(data, aes(x=Normalized\_carb, y=Normalized\_fat, colour=class))  
d + geom\_point(size=4, alpha=0.8) +  
 scale\_fill\_brewer(type="qual", palette=1, name="Classes of candy") +  
 scale\_colour\_brewer(type="qual", palette=1, name="Classes of candy") +  
 xlab(label="Total carbohydrate gm per gm serving size") +  
 ylab(label="Total fat gm per gm serving size") +  
 ggtitle("Fat vs. Carbohydrate") +  
 theme\_bw()



## Student #2

Begin homework here

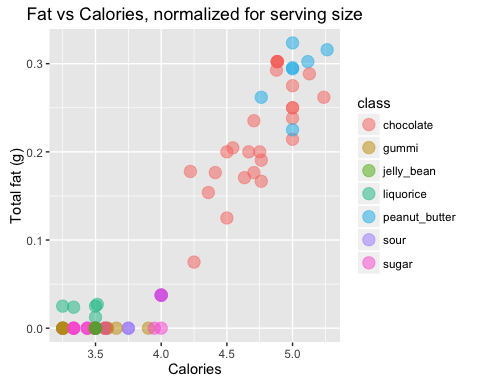
library(ggplot2)  
data = read.table ("./candy\_nutrition.txt", header=T)

normalized scatterplot of calories vs fat

data$Calories\_per\_serving\_size = data$calories / data$serving\_size\_g  
data$fat\_per\_serving\_size = data$total\_fat\_g / data$serving\_size\_g  
  
head(data)

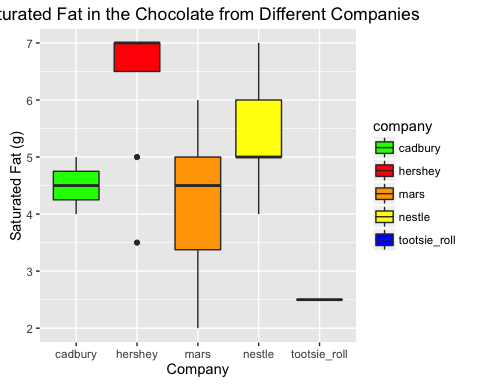
## id name company class serving\_size\_g  
## 1 id\_1 mini\_eggs cadbury chocolate 40  
## 2 id\_2 soft\_eating\_liquorice darrell\_lea liquorice 42  
## 3 id\_3 raspberries haribo sugar 39  
## 4 id\_4 candy\_corn nice gummi 41  
## 5 id\_5 crawlers\_minis trolli sour 40  
## 6 id\_6 strawberry\_shortcake\_mms mars chocolate 42  
## calories calories\_fat total\_fat\_g saturated\_fat\_g cholesterol\_mg  
## 1 190 70 8 5 5  
## 2 140 10 1 0 0  
## 3 140 0 0 0 0  
## 4 160 160 0 0 0  
## 5 130 0 0 0 0  
## 6 210 100 10 6 5  
## sodium\_mg total\_carb\_g dietary\_fiber\_g sugars\_g protein\_g  
## 1 30 28 0.5 27 2  
## 2 40 30 0.0 16 1  
## 3 0 36 0.0 29 1  
## 4 75 39 0.0 32 0  
## 5 35 31 0.0 24 1  
## 6 40 29 0.0 28 2  
## primary\_ingredient Calories\_per\_serving\_size fat\_per\_serving\_size  
## 1 chocolate 4.750000 0.20000000  
## 2 syrup 3.333333 0.02380952  
## 3 sugar 3.589744 0.00000000  
## 4 sugar 3.902439 0.00000000  
## 5 syrup 3.250000 0.00000000  
## 6 chocolate 5.000000 0.23809524

p = ggplot(data=data, aes(x=Calories\_per\_serving\_size, y=fat\_per\_serving\_size, colour=class))  
p + geom\_point(size=4, alpha=0.5)+ #alpha = transparency of the points  
 scale\_fill\_brewer(type="qual", palette=5, name="Classes of candy")+  
 xlab(label="Calories")+  
 ylab(label="Total fat (g)")+  
 ggtitle("Fat vs Calories, normalized for serving size")



comparison of saturated fat from different chocolate companies

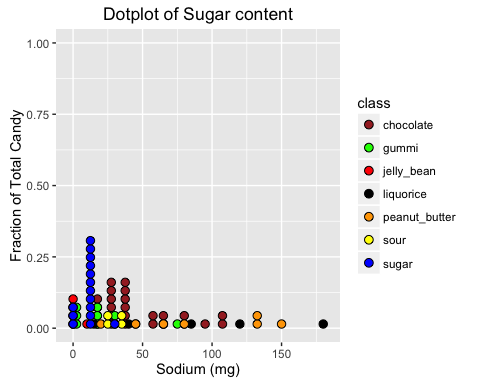
just\_chocolate = subset(data, class=="chocolate")  
  
p = ggplot(data=just\_chocolate, aes(x=company, y=saturated\_fat\_g, fill=company))  
p + geom\_boxplot()+  
 xlab(label="Company")+  
 ylab(label="Saturated Fat (g)")+  
 ggtitle("Saturated Fat in the Chocolate from Different Companies")+  
scale\_fill\_manual(values=c( "green","red", "orange", "yellow", "blue"))



dotplot histogram of sodium content

p = ggplot(data=data, aes(x=sodium\_mg, fill=class))  
p + geom\_dotplot()+  
 scale\_fill\_manual(values=c("brown", "green","red", "black", "orange", "yellow", "blue"))+  
 xlab(label="Sodium (mg)")+  
 ylab(label="Fraction of Total Candy")+  
 ggtitle("Dotplot of Sugar content")

## `stat\_bindot()` using `bins = 30`. Pick better value with `binwidth`.

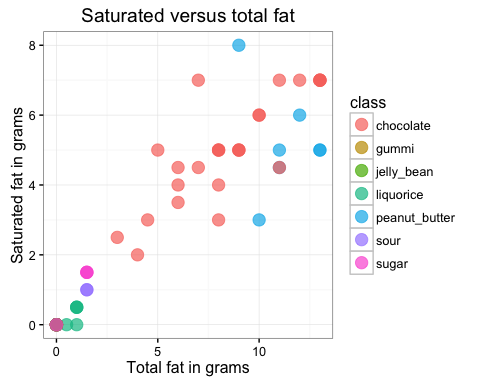


## Student #3

library(ggplot2)  
data <- read.table("./candy\_nutrition.txt", header=TRUE)

Let's look at saturated fat versus total fat to get an idea of what percentage saturated fat is in each candy

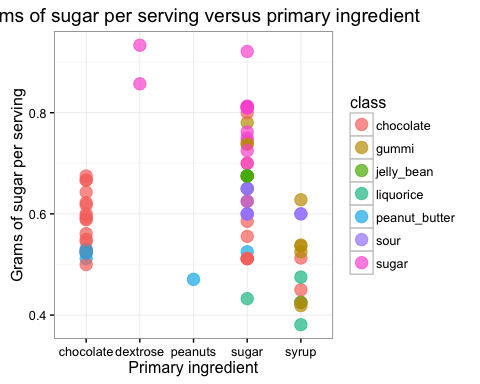
p <- ggplot(data=data, aes(x=total\_fat\_g, y=saturated\_fat\_g, colour=class))  
  
p + geom\_point(size=4, alpha=0.7) +   
    
  scale\_fill\_brewer(type="qual", palette=1, name="Classes of candy") +   
  xlab(label="Total fat in grams") +   
  ylab(label="Saturated fat in grams") +   
  ggtitle("Saturated versus total fat") +   
  theme\_bw()



It seems like this mostly follows a regression line of slope 1 so there is more-or-less a 1-to-1 correlation between saturated fat and total fat meaning that for most of these candies, half of the fat is unsaturated

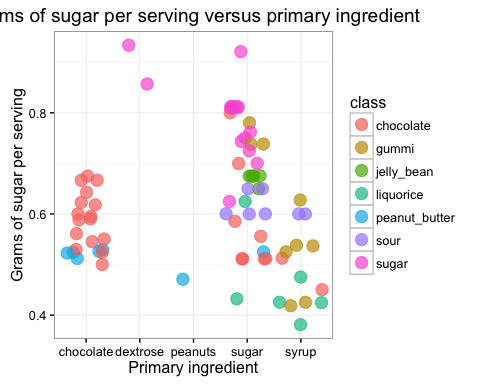
Let's see if there is any correlation between the primary ingredient and the total amount of sugar per serving size

p <- ggplot(data=data, aes(x=primary\_ingredient, y=(sugars\_g/serving\_size\_g), colour=class))  
  
p + geom\_point(size=4, alpha=0.7) +   
    
  scale\_fill\_brewer(type="qual", palette=1, name="Classes of candy") +   
  xlab(label="Primary ingredient") +   
  ylab(label="Grams of sugar per serving") +   
  ggtitle("Grams of sugar per serving versus primary ingredient") +   
  theme\_bw()



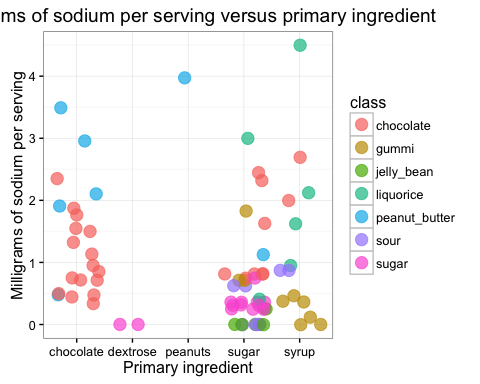
DC: Let's try the same graph with position="jitter"

p + geom\_point(size=4, alpha=0.7, position="jitter") +   
    
  scale\_fill\_brewer(type="qual", palette=1, name="Classes of candy") +   
  xlab(label="Primary ingredient") +   
  ylab(label="Grams of sugar per serving") +   
  ggtitle("Grams of sugar per serving versus primary ingredient") +   
  theme\_bw()



Now let's do the same but testing sodium

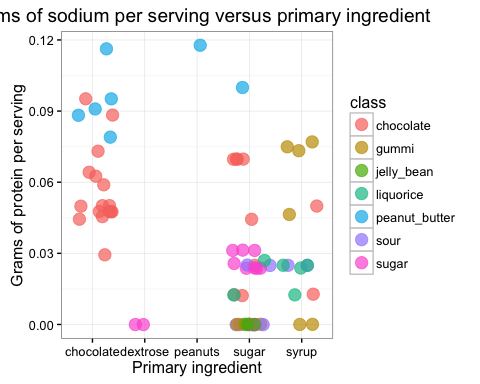
p <- ggplot(data=data, aes(x=primary\_ingredient, y=(sodium\_mg/serving\_size\_g), colour=class))  
  
p + geom\_point(size=4, alpha=0.7, position="jitter") +   
    
  scale\_fill\_brewer(type="qual", palette=1, name="Classes of candy") +   
  xlab(label="Primary ingredient") +   
  ylab(label="Milligrams of sodium per serving") +   
  ggtitle("Milligrams of sodium per serving versus primary ingredient") +   
  theme\_bw()



There doesn't seem to be much clustering of the data so there probably isn't a very strong correlation between milligrams of sodium per serving and the type of primary ingredient

What about protein?

p <- ggplot(data=data, aes(x=primary\_ingredient, y=(protein\_g/serving\_size\_g), colour=class))  
  
p + geom\_point(size=4, alpha=0.7, position="jitter") +   
    
  scale\_fill\_brewer(type="qual", palette=1, name="Classes of candy") +   
  xlab(label="Primary ingredient") +   
  ylab(label="Grams of protein per serving") +   
  ggtitle("Grams of sodium per serving versus primary ingredient") +   
  theme\_bw()



## Student #4

3 PLOTS ASSIGNMENT, due 5pm on 4-3-16

PLOT NUMBER ONE: a bar graph of mean calories per gram for each candy class

upload text document called candy\_nutrition.txt by clicking "import dataset" in the Environment window, and choosing From Local File, and finding candy\_nutrition.txt

rename candy\_nutrition.txt as "data"

library(ggplot2)  
data <- read.table("./candy\_nutrition.txt", header=TRUE)

create a new column in the data table that is calories per gram for every candy: divide calories per serving divided by # of grams in serving

data$cals\_per\_gram <- (data$calories / data$serving\_size\_g)

then reduce the number of decimal places in my new column down to hundredths, aka 2 decimal places

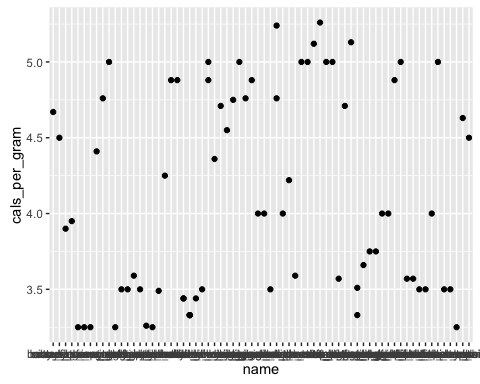
data$cals\_per\_gram <- round(data$cals\_per\_gram, 2)

install ggplots library, and upload the library into my current session

library(ggplot2)

then make a scatterplot where the x-axis is candy name, and the y-axis is the number of calories per gram of each candy

p <- ggplot(data, aes(name,cals\_per\_gram))  
p + geom\_point()



that's ugly and the x-axis labels are unreadable, so intead I'll make a plot where the x-axis is candy class, and y-axis is the mean calories per gram of that class

simply calculating the mean of the cals\_per\_gram sorted by candy class using tapply gives me a vector, not a data frame

mean\_cals\_per\_gram\_by\_class <- tapply(data$cals\_per\_gram, data$class, mean)

look at the output vector by executing this:

mean\_cals\_per\_gram\_by\_class

## chocolate gummi jelly\_bean liquorice peanut\_butter   
## 4.751923 3.504444 3.450000 3.431667 5.020000   
## sour sugar   
## 3.642857 3.616667

the vector looks like this, and lacks column labels which I need to have in order to generate a plot with the axes labeled properly chocolate gummi jelly\_bean liquorice peanut\_butter sour sugar 4.751923 3.504444 3.450000 3.431667 5.020000 3.642857 3.616667

can I use that "vector" to make a decent scatterplot?

p <- ggplot(mean\_cals\_per\_gram\_by\_class)   
p + geom\_point()

NOPE! Because this "vector" lacks column names and my ggplot command lacks definitions for the x and y axes, I can't make a plot.

Therefore, I need to turn that "vector" into a data.frame

cal\_density\_by\_class <- data.frame(mean\_cals\_per\_gram\_by\_class)

look at my new data frame

cal\_density\_by\_class

## mean\_cals\_per\_gram\_by\_class  
## chocolate 4.751923  
## gummi 3.504444  
## jelly\_bean 3.450000  
## liquorice 3.431667  
## peanut\_butter 5.020000  
## sour 3.642857  
## sugar 3.616667

that's nice, but it still doesn't have a column name for candy class instead of using "tapply", use "aggregate" to create a data table that puts candy class in its own column

aggregate\_test = aggregate(data$cals\_per\_gram, list(data$class), mean)

look at that new table

aggregate\_test

## Group.1 x  
## 1 chocolate 4.751923  
## 2 gummi 3.504444  
## 3 jelly\_bean 3.450000  
## 4 liquorice 3.431667  
## 5 peanut\_butter 5.020000  
## 6 sour 3.642857  
## 7 sugar 3.616667

it looks like this: Group.1 x 1 chocolate 4.751923 2 gummi 3.504444 3 jelly\_bean 3.450000 4 liquorice 3.431667 5 peanut\_butter 5.020000 6 sour 3.642857 7 sugar 3.616667

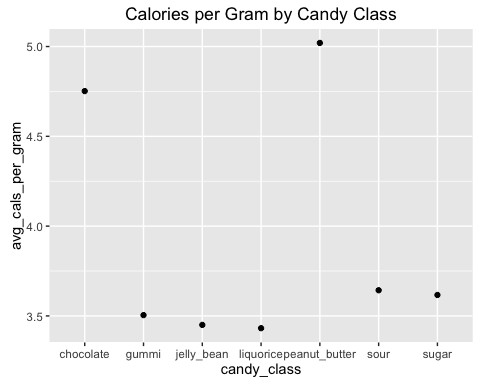
now it's in a nice data table format like an excel file, not just a list of numbers like the "vector" was.

now I need to rename the columns from "Group.1" and "x", to "candy\_class" and "mean\_cals\_per\_gram" respectively

names(aggregate\_test)[1] <- paste("candy\_class")  
names(aggregate\_test)[2] <- paste("avg\_cals\_per\_gram")

now that I have a dataframe with column names, I'll try turning it into a scatterplot, which ggplot2 calls geom\_point

p <- ggplot(aggregate\_test, aes(candy\_class, avg\_cals\_per\_gram))  
p + geom\_point() +   
 ggtitle(label="Calories per Gram by Candy Class")



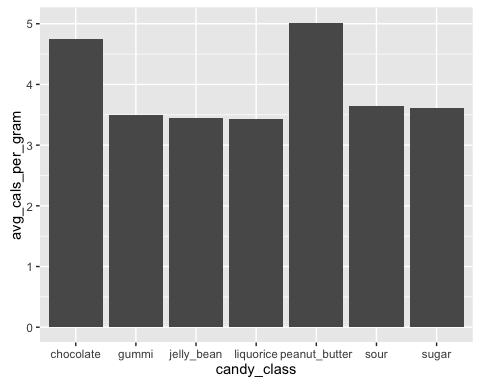
I tried to turn this into a bar plot which would have also made the y-axis range start at zero so the values don't look so skewed, but I was unsuccessful. Every attempt I made returned some error.

p <- ggplot(data=aggregate\_test, aes(candy\_class, avg\_cals\_per\_gram)) p + geom\_bar() Error: stat\_count() must not be used with a y aesthetic.

That's probably the simplest of my various attempts at achieving a bar plot, but that error message was the main problem I couldn't figure out how to fix.

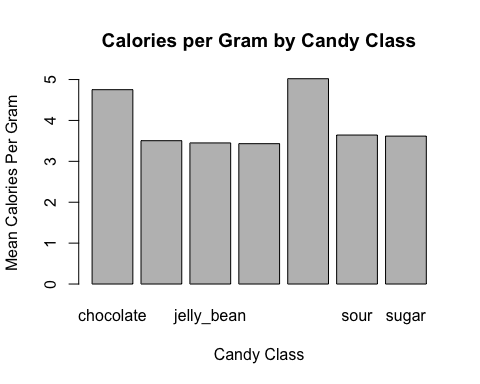
DC: Let me see if I can help out. You can read here, under ["Details"](http://docs.ggplot2.org/0.9.3.1/geom_bar.html) that by default geom\_bar takes count data through stat="bin". Let's change that to stat="identity":

p <- ggplot(data=aggregate\_test, aes(candy\_class, avg\_cals\_per\_gram))  
p + geom\_bar(stat="identity")



So I'll use the tips I found on Quick-R instead of ggplot

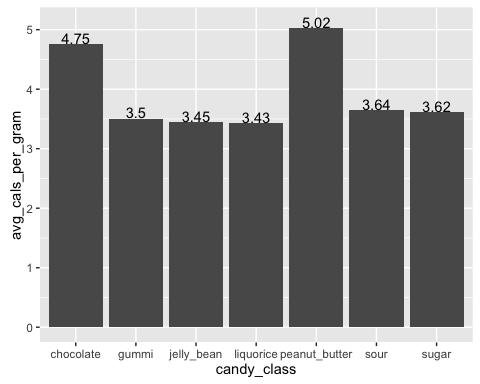
density <- table(aggregate\_test$candy\_class)  
barplot(density, main="Calories per Gram by Candy Class", height=aggregate\_test$avg\_cals\_per\_gram, names.arg=(aggregate\_test$candy\_class), xlab="Candy Class", ylab="Mean Calories Per Gram")



It would be nice to put the values at the ends of each bar, but I couldn't find any tips online for that other than "make a text box and manually enter coordinates until you figure out which coordinates place the text over the desired bar", which sounds sloppy

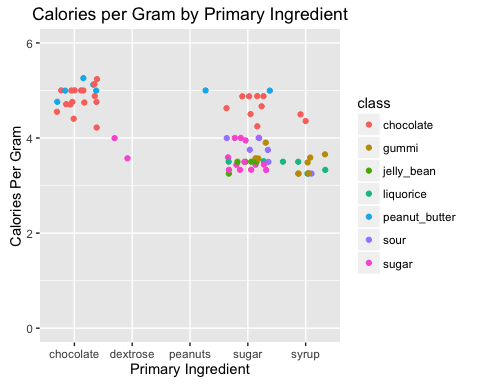
DC: Let me see if I can help out with that too! I might try geom\_text, let's give it a shot! I'll specify x=candy\_class and y=avg\_cals\_per\_gram values + 0.1 (so that it dodges the bars). Then, the labels are rounded avg\_cals\_per\_gram values.

p <- ggplot(data=aggregate\_test, aes(candy\_class, avg\_cals\_per\_gram))  
p + geom\_bar(stat="identity") + geom\_text(data=aggregate\_test, aes(x=candy\_class, y=avg\_cals\_per\_gram+0.1, label=round(avg\_cals\_per\_gram, digits=2)))



PLOT NUMBER TWO: x=primary ingredient, y=calories per gram, color=candy class

p <- ggplot(data, aes(primary\_ingredient, cals\_per\_gram))  
p + geom\_jitter(aes(colour = class)) +  
 ggtitle(label = "Calories per Gram by Primary Ingredient") +  
 xlab(label = "Primary Ingredient") +  
 ylab(label = "Calories Per Gram") +  
 ylim(0,6)



PLOT NUMBER THREE: stacked bars showing the % by weight of macronutrient types (total fat, total carbs, total protein) in a serving, sorted by candy class

Make new columns in the dataframe "data" for "percent by weight of fat", "percent by weight of carbs", and "percent by weight of protein"

data$percent\_fat\_by\_weight <- (data$total\_fat\_g / data$serving\_size\_g)  
data$percent\_carbs\_by\_weight <- (data$total\_carb\_g / data$serving\_size\_g)  
data$percent\_protein\_by\_weight <- (data$protein\_g / data$serving\_size\_g)

round each of them down to a tenth of a percent, then multiply by 100 to get actual percentage values

data$percent\_fat\_by\_weight <- round(data$percent\_fat\_by\_weight, 3) \* 100  
data$percent\_carbs\_by\_weight <- round(data$percent\_carbs\_by\_weight, 3) \* 100  
data$percent\_protein\_by\_weight <- round(data$percent\_protein\_by\_weight, 3) \* 100

that's good, but too many bars to plot on a single chart, so now I need to get the average percent macronutrient values for each candy class

mean\_fat\_percent <- tapply(data$percent\_fat\_by\_weight, data$class, mean)  
mean\_carbs\_percent <- tapply(data$percent\_carbs\_by\_weight, data$class, mean)  
mean\_protein\_percent <- tapply(data$percent\_protein\_by\_weight, data$class, mean)

Check to see if these look reasonable

mean\_fat\_percent

## chocolate gummi jelly\_bean liquorice peanut\_butter   
## 22.038462 0.000000 0.000000 1.883333 28.828571   
## sour sugar   
## 1.085714 0.760000

mean\_carbs\_percent

## chocolate gummi jelly\_bean liquorice peanut\_butter   
## 68.48846 84.02222 84.50000 78.26667 58.41429   
## sour sugar   
## 85.00000 89.29333

mean\_protein\_percent

## chocolate gummi jelly\_bean liquorice peanut\_butter   
## 5.142308 3.022222 0.000000 2.083333 9.814286   
## sour sugar   
## 1.428571 1.433333

Now I have vectors listing the mean macronutrient percentages for each candy class. Combine those three vectors into a single dataframe

mean\_macronutrient\_percents <- data.frame(mean\_fat\_percent, mean\_carbs\_percent, mean\_protein\_percent)

Again, I have the problem where the column listing class doesn't have a title

macronutrient\_aggregate = aggregate(data$percent\_fat\_by\_weight,percent\_carbs\_by\_weight,percent\_protein\_by\_weight, list(data$class), mean)

The aggregate command didn't work this time maybe because I'm trying to create a dataframe with more than two columns??? So I'll just copy and paste the dataframe into Excel, add the desired column title, copy that into a text file(because apparently Excel files can't upload into R properly), and upload the text file as a new dataframe.

Now I have a dataframe I can work with, so I'll try to make a stacked bar chart from those values

p <- ggplot(data = data, aes(x = class, y = percent\_fat\_by\_weight, percent\_carbs\_by\_weight, percent\_protein\_by\_weight))  
p + geom\_bar()

That ^ doesn't seem to be working, so I'll try to make my chart from the percent \_\_\_ by weight values I added to the original dataframe "data"

p <- ggplot(data = data, aes(x = class, y = percent\_fat\_by\_weight, percent\_carbs\_by\_weight, percent\_protein\_by\_weight))  
p + geom\_bar()

That ^ also doesn't work.

barplot(text\_mean\_macronutrient\_percents, main = "Macronutrient Percents by Candy Class", xlab = "Candy Class", ylab = "Percent")

Nope^

DC: Let's see what we can do to create this bar graph with stacked percentages

Let's create a smaller dataset with just the class of candy and the percents by weights of fat, carbs, and protein.

names(data)

## [1] "id" "name"   
## [3] "company" "class"   
## [5] "serving\_size\_g" "calories"   
## [7] "calories\_fat" "total\_fat\_g"   
## [9] "saturated\_fat\_g" "cholesterol\_mg"   
## [11] "sodium\_mg" "total\_carb\_g"   
## [13] "dietary\_fiber\_g" "sugars\_g"   
## [15] "protein\_g" "primary\_ingredient"   
## [17] "cals\_per\_gram" "percent\_fat\_by\_weight"   
## [19] "percent\_carbs\_by\_weight" "percent\_protein\_by\_weight"

sub\_data <- data[c(4,18:20)]

Next, let's use a function in another package created by Hadley Wickham (creator of ggplot2) called "reshape2", which allows us to reformat our data with ease. What we want to do is put the percents by weight of fat, carbs, and protein into a single column and then create a new column that says what they are (fat, carb, or protein). The function is called "melt", and it collapses your data except for the columns you specify with the argument "id". Let's try it!

names(sub\_data)

## [1] "class" "percent\_fat\_by\_weight"   
## [3] "percent\_carbs\_by\_weight" "percent\_protein\_by\_weight"

library(reshape2)  
melted\_data <- melt(sub\_data, id="class")  
  
names(melted\_data)

## [1] "class" "variable" "value"

head(melted\_data)

## class variable value  
## 1 chocolate percent\_fat\_by\_weight 20.0  
## 2 liquorice percent\_fat\_by\_weight 2.4  
## 3 sugar percent\_fat\_by\_weight 0.0  
## 4 gummi percent\_fat\_by\_weight 0.0  
## 5 sour percent\_fat\_by\_weight 0.0  
## 6 chocolate percent\_fat\_by\_weight 23.8

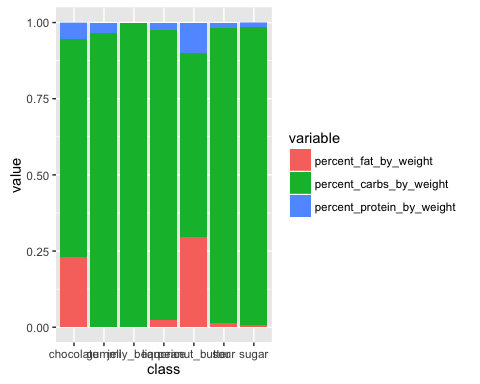
Great! Now let's try to create that bar plot

p <- ggplot(data = melted\_data, aes(x = class, y = value, fill= variable))  
p + geom\_bar(stat="identity")



These appear to be unequal because I believe we calcaulted percents fat, carbs, and protein for each candy type within each class of candy? If so, then let's try to normalize to get "percents" for each class

p <- ggplot(data = melted\_data, aes(x = class, y = value, fill= variable))  
p + geom\_bar(stat="identity", position="fill")



Also, see position="dodge"

p <- ggplot(data = melted\_data, aes(x = class, y = value, fill= variable))  
p + geom\_bar(stat="identity", position="dodge")

