Lab 2

By Joshua Williams

```
FakeData <- read.csv(datasource1, stringsAsFactors=TRUE)
ClassSurveys <- read.csv(datasource2, stringsAsFactors=TRUE)
ClassSurveysSP2021 <- read.csv(datasource3, stringsAsFactors=TRUE)
```

Question 01

Answer the following two questions based on what you have done in class this far into the semester.

1. **What are the appropriate numerical and graphical summaries of quantitative variables?**

- There are several possible ways to measure numerical data, and each of them reveals slightly different information. Boxplots give high-level summaries of aggregated data. Histograms and density plots help show a dataset's distribution. Time plots can show how an explanatory variable changes with respect to an observational variable. Like time plots, Scatter plots show relationships possible relationships between two variables. They can be cleaner than a line plot because they don't assume that the data has an order to it as time plots do.

2. **What are the appropriate numerical and graphical summaries of qualitative variables?**

- Stacked bar graphs can show the proportion each block takes up in a given factor. Side-by-side bar groups and pie charts can reveal how a group is partitioned into sub-groups.

Question 02

Last week you learned how to compute the mean and standard deviation of a list of quantitative values. The syntax will be very similar for these other tools. For example, based on last week's lab you could compute the mean of the Age variable in FakeData using the command mean(`FakeData$Age`). Almost all of the standard numerical summaries only require replacing the function mean, with a new statistical word as follows:

1. **In order to compute the median, replace mean with median and input median(`FakeData$Age`). Double check that you obtain 56.**

```
mean(FakeData$Age)
```

```
## [1] 56.12874
```

```
median(FakeData$Age)
```

```
## [1] 56
```

2. **In order to compute the quartiles, R uses a more general function for computing any percentile, using the quantile function. You may either specify the percent cutoff you want, e.g. .25 for the first quartile, or specify none and obtain the five number summary. Try entering `quantile(FakeData$Age, 0.25)`. You should obtain 49. What is the command to obtain the 3rd quartile?**

```
quantile(FakeData$Age,0.25)
```

```
## 25%
##  49
```

```
quantile(FakeData$Age)
```

```
##   0%  25%  50%  75% 100%
##   41   49   56   63   70
```

3. **Computing the IQR is as easy as can be. The IQR of the age should be 14 when you compute it in R.**

```
IQR(FakeData$Age)
```

```
## [1] 14
```

4. **The maximum and minimum values can be computed using the max and min functions respectively. Implement each of these on Age.**

```
min(FakeData$Age)
```

```
## [1] 41
```

```
max(FakeData$Age)
```

```
## [1] 70
```

5. **Compute the mean, median, standard deviation and IQR for the height variable in FakeData.**

```
mean(FakeData$Height)
```

```
## [1] 64.15569
```

```
median(FakeData$Height)
```

```
## [1] 64
```

```
sd(FakeData$Height)
```

```
## [1] 4.603793
```

```
IQR(FakeData$Height)
```

```
## [1] 8
```

Question 03

Bar graphs and pie charts

In order to create a bar graph or pie chart in R, you must first tabulate your counts and then convert those to percents. Once you have saved this table you input it into one of the cleverly named functions such as barplot or pie for creating the respective plots. Next you will create a bar graph summarizing the Favorite Breakfast Column. Here is the process broken down into steps:

1. **Create a table of counts**

```
tble <- table(FakeData$Breakfast)
tble <- sort(tble)
df3<- as.data.frame(tble)
df3$df3 <- as.factor(df3$Var1)
colnames(df3) <- c('group', 'frequency')
#This organizes the data from smallest to largest
df3$frequency <- sort(df3$frequency,decreasing=FALSE)
#This gets rid of an unecessary column in the data frame
df3<-df3[1:2]
df3
```

```
##      group frequency
## 1    Grain        11
## 2     Meat        25
## 3     None        27
## 4    Fruit        61
## 5 Multiple       104
## 6    Dairy       106
```

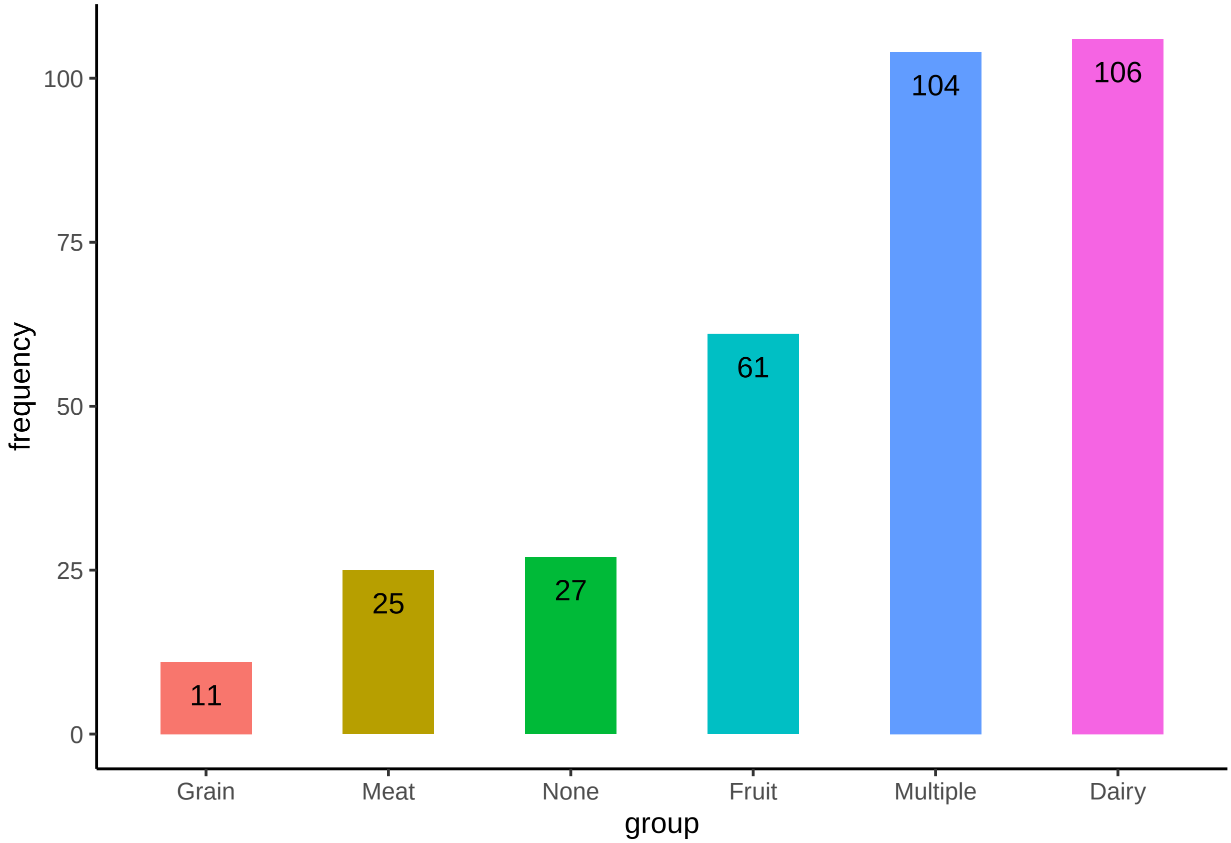
3.1 Create a Bar Graph

You may have to resize the plot window to get the full bar labels to display. Frequently we may also want to control the colors of the bars on the bar graph to make the groups stand out. The only problem with this plot is that it would be helpful to color the boxes by group so that the relationships between groups are easier to identify. The barplot command has a color option to pass in a list of colors to be used for distinguishing groups. Before we do this we will need to know how many groups of the Breakfast variable there are. We can find this in the previous graph without the color, or by using the summary command and counting how many groups of this variable are summarized. In this case there are 6, so we will need to specify a list of 6 colors. R has several ways of making lists of colors, which are called color palettes, e.g. `rainbow`, `cm.colors`, `topo.colors`, `heat.colors`, `terrain.colors`, `etc`. The syntax for generating a list of 6 colors (do not enter this yet) might looks like `heat.colors(6)`. Now to create the colored paragraph, enter the command

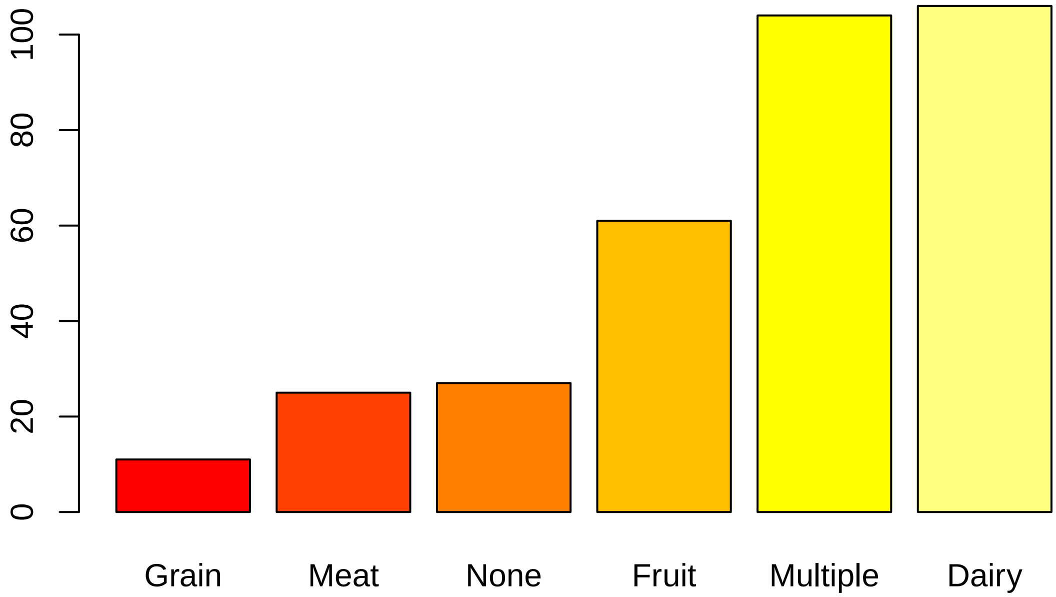
```
tble <- table(FakeData$Breakfast)
tble <- sort(tble)
df3<- as.data.frame(tble)
df3$df3 <- as.factor(df3$Var1)
colnames(df3) <- c('group', 'frequency')
#This organizes the data from smallest to largest
df3$frequency <- sort(df3$frequency,decreasing=FALSE)
#This gets rid of an unnecessary column in the data frame
df3<-df3[1:2]
df3
```

```
##      group frequency
## 1    Grain        11
## 2     Meat        25
## 3     None        27
## 4    Fruit        61
## 5 Multiple       104
## 6    Dairy       106
```

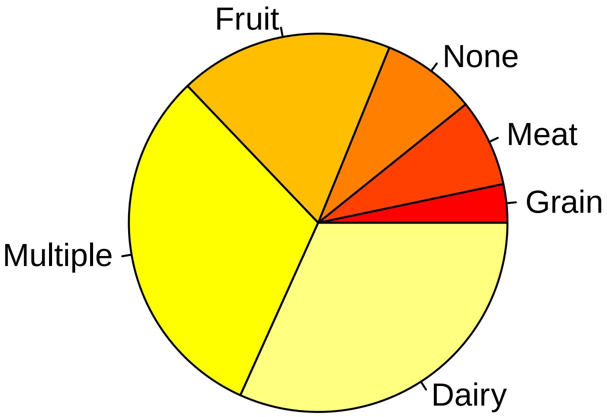
```
ggplot(data = df3)+
  geom_bar(mapping=aes(x=group,y=frequency,fill=group),
           width=0.5,
           stat='identity',
           show.legend=FALSE)+
  geom_text(mapping =
              aes(x=group,y=frequency,label=frequency),
            check_overlap=TRUE,
            nudge_y=-5)+
  theme_classic()
```



```
tble <- sort(tble)
barplot(tble, col = heat.colors(6))
```



```
pie(tble, col = heat.colors(6))
```



### **3.2 Percents on the bar graph** Frequently one wishes to display the bar graph with percents instead of counts. This will just require updating the table and redrawing the plot. Since we want proportions, simply enter the command

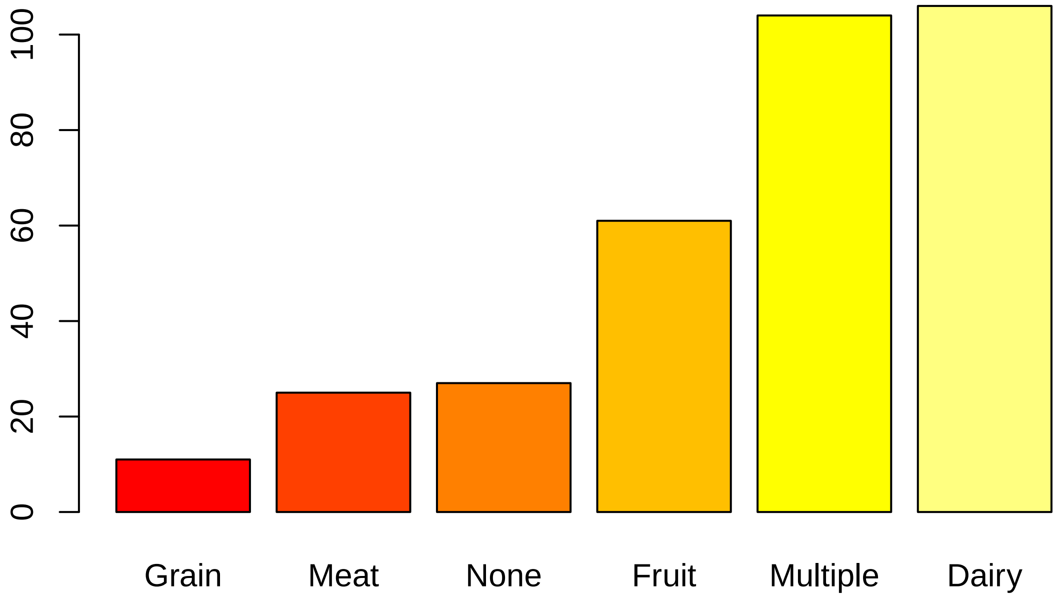
```
proptble <- prop.table(tble)
```

which computes the percentages for the table and stores them under the variable proptable. Display the new table proptble. You can recreate the bar graph or the pie chart by simply replacing the old table tble with the new table proptble and executing the command as before. Construct both of these plots and use a different color palette from those listed above, or any other you find online.

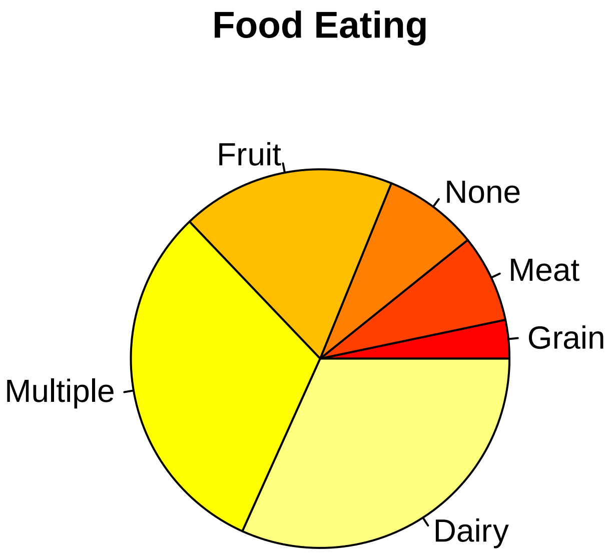
```
proptble <- prop.table(tble)
proptble
```

```
##
##      Grain       Meat       None      Fruit   Multiple      Dairy
## 0.03293413 0.07485030 0.08083832 0.18263473 0.31137725 0.31736527
```

```
barplot(tble, col = heat.colors(6))
```



```
pie(tble, col = heat.colors(6))+
  title('Food Eating')
```

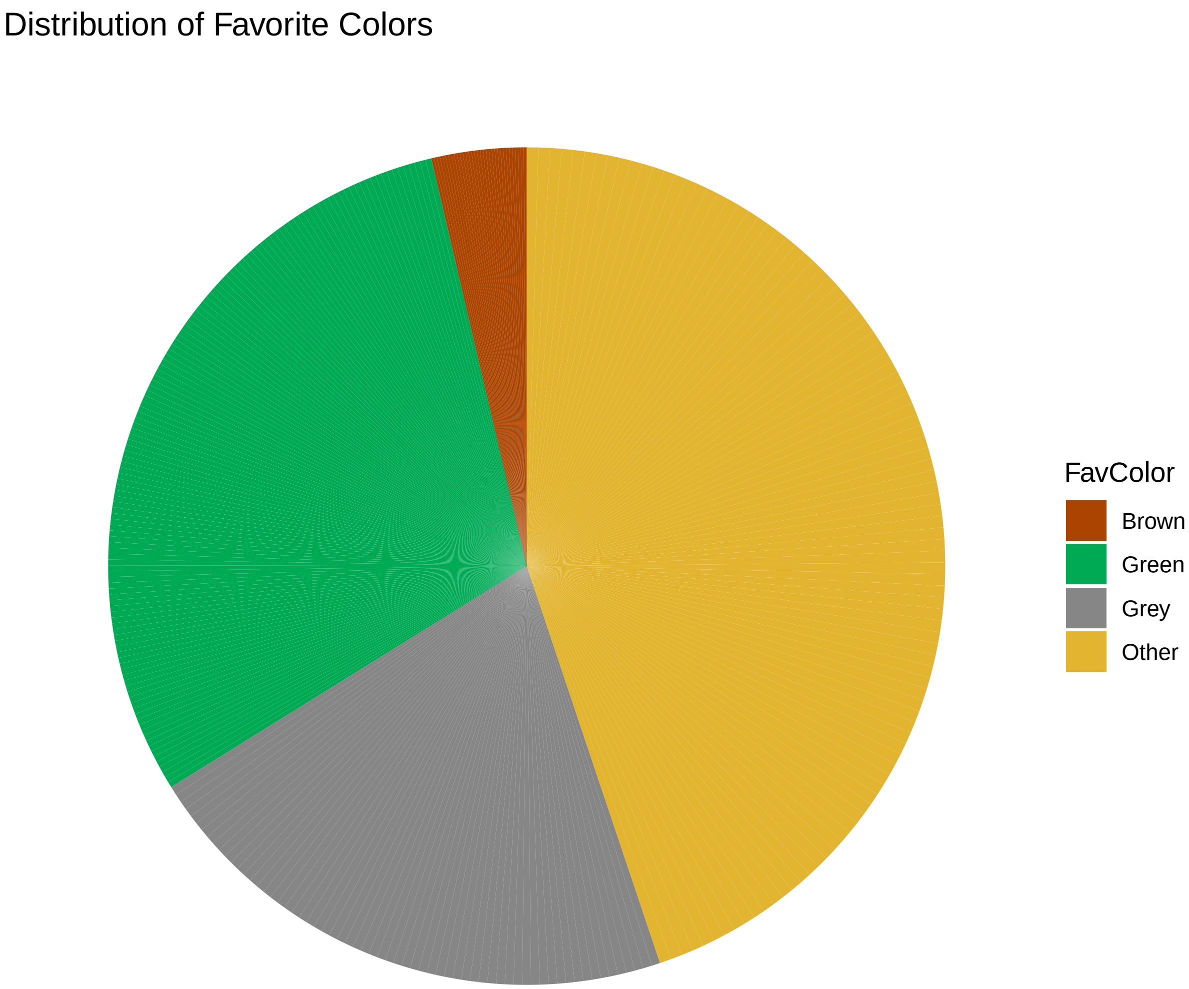


```
## integer(0)
```

Question 04

Create a bar graph of the percentages summarizing the Favorite Color variable, using colors, and add appropriate plot labels (look back at the previous lab).

```
mycolors=c("#AA4400","#00AA55","#868686","#E2B430")
g01 <- ggplot(data = FakeData)+
  geom_bar(mapping=aes(x='',
                       y=FavColor,
                       fill=FavColor,),
           width = 0.5,
           stat='identity')+
  labs(title="Distribution of Favorite Colors")+
  scale_fill_manual(values = mycolors)+
  coord_polar('y',start=0)+
  theme_void()
g01
```


Distribution of Favorite Colors
FavColor
Brown
Green
Grey
Other

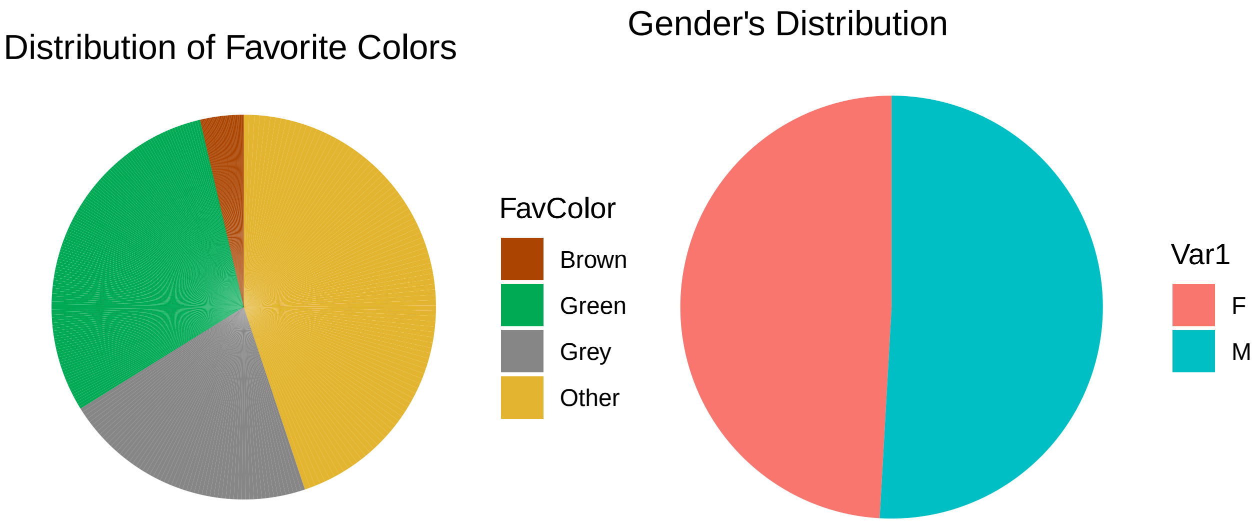
Question 05

For the remainder of this lab you are going to summarize the data from the class survey using the appropriate numerical and graphical summary. In your final report include an appropriate numerical and graphical summary for each variable in the `ClassSurveysFA2020.csv` dataset.

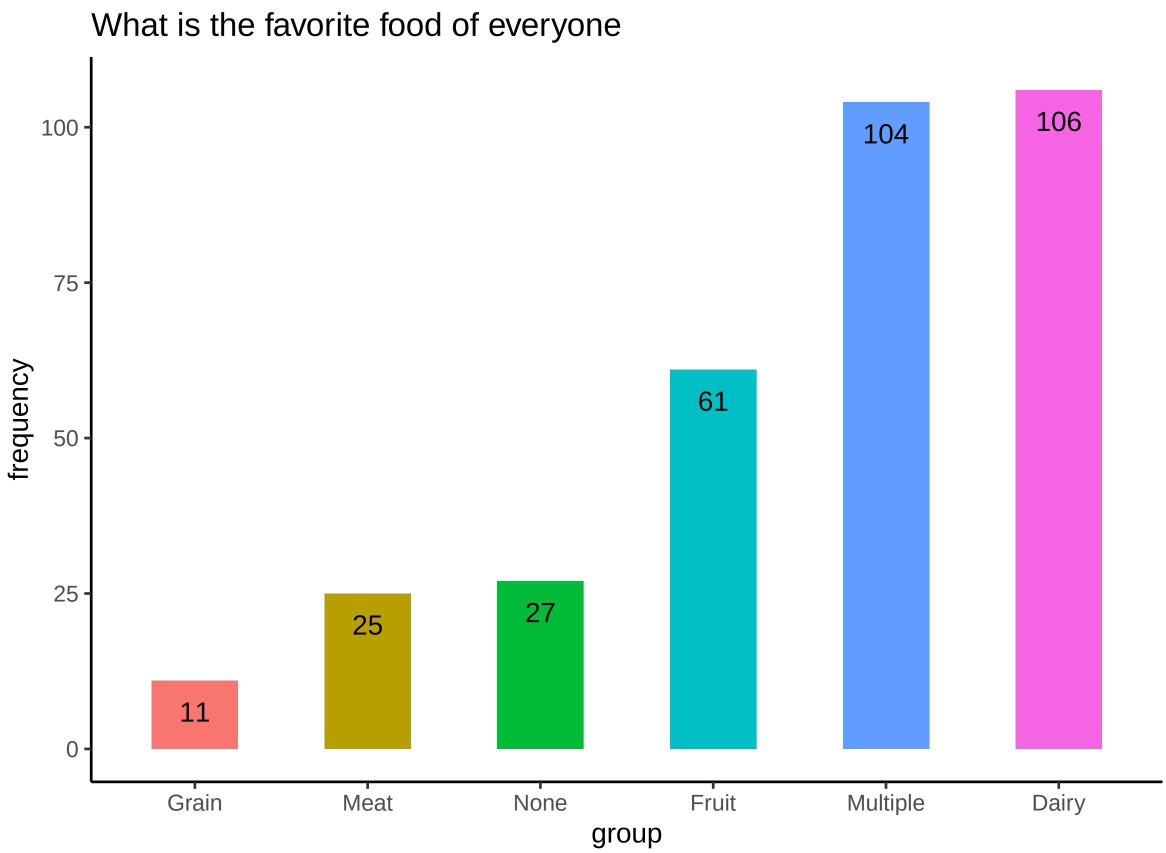
Use of Multiplot Formula from online for simplicity

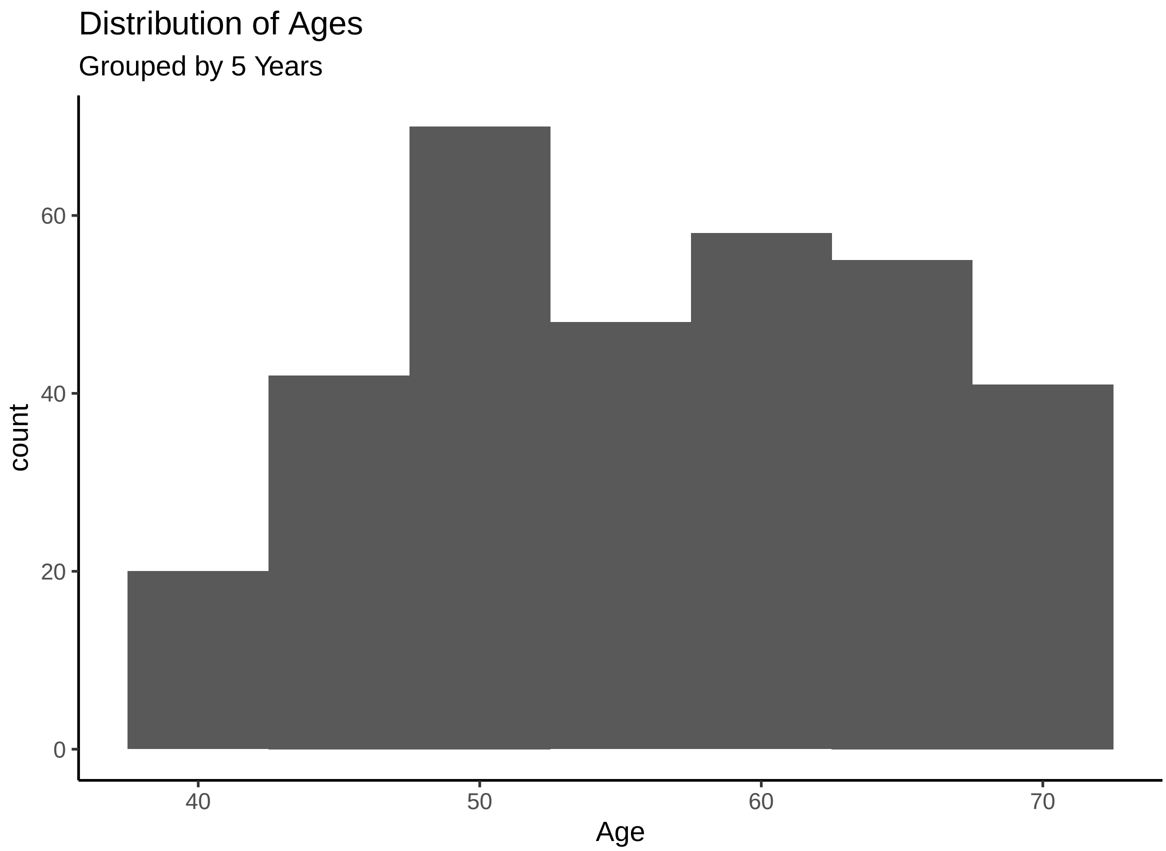
```
## This isn't my code, I just learned how to apply it online.
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

  if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

```
#This shows everyone's favorite color
mycolors=c("#AA4400","#00AA55","#868686","#E2B430")

g01 <- ggplot(data = FakeData)+
  geom_bar(mapping=aes(x='',
                       y=FavColor,
                       fill=FavColor,),
           width = 0.5,
           stat='identity')+
  labs(title="Distribution of Favorite Colors")+
  scale_fill_manual(values = mycolors)+
  coord_polar('y',start=0)+
  theme_void()


#This shows everyone's gender
g02<- ggplot(data=as.data.frame(table(FakeData$Gender)))+
  geom_bar(mapping=aes(x='',y=Freq,fill=Var1),width=1,stat='identity')+
  coord_polar('y',start=0)+
  theme_void()+
  labs(title="Gender's Distribution")
multiplot(g01,g02,cols=2)
```



```
#This shows what people have for breakfast.
g03<- ggplot(data = df3)+
  geom_bar(mapping=aes(x=group,
                       y=frequency,
                       fill=group)
           ,width=0.5,
           stat='identity',
           show.legend=FALSE)+
  labs(title="What is the favorite food of everyone")+
  geom_text(mapping =
              aes(x=group,
                  y=frequency,
                  label=frequency),
            check_overlap=TRUE,nudge_y=-5)+theme_classic()
#This Shows the Ages of People
g04 <- ggplot(data = FakeData)+
  geom_histogram(mapping =aes(x=Age,
                              color=Age),binwidth = 5)+
  labs(title="Distribution of Ages",
       subtitle='Grouped by 5 Years')+
  theme_classic()
#The final result presented nicely

multiplot(g03,g04,cols=1)
```


What is the favorite food of everyone
frequency
0
25
50
75
100
11
25
27
61
104
106
Grain
Meat
None
Fruit
Multiple
Dairy
group


Distribution of Ages
Grouped by 5 Years
count
0
20
40
60
40
50
60
70
Age

```r
g05<-ggplot(data = FakeData)+
  geom_histogram(mapping =aes(x=Age,
                              color=Age),
                 binwidth = 5)+
  labs(title="Figure 05: Ages",
       subtitle='Binwidth = 5 Years')+theme_classic()
g06<-ggplot(data = FakeData)+
  geom_histogram(mapping =aes(x=Height,color=Height),
            binwidth=1)+
  labs(title="Figure 06: Height",
       subtitle='Binwidth = 1 Inch')+theme_classic()

g07<-ggplot(data = FakeData)+
  geom_histogram(mapping =aes(x=Handspan,color=Handspan),
            binwidth=0.1)+
  labs(title="Figure 07: Handspan",
       subtitle='Binwidth = 0.1 Inch')+theme_classic()

g08<-ggplot(data = FakeData)+
  geom_histogram(mapping =aes(x=Pinkylen,color=Pinkylen),
            binwidth=0.1)+
  labs(title="Figure 08 Pinky Length",
       subtitle='Binwidth = 0.1 Inch')+
  xlab('Pinky Length')+
  theme_classic()
multiplot(g05,g06,g07,g08,cols=2)
```

Figure 05: Ages

Binwidth = 5 Years

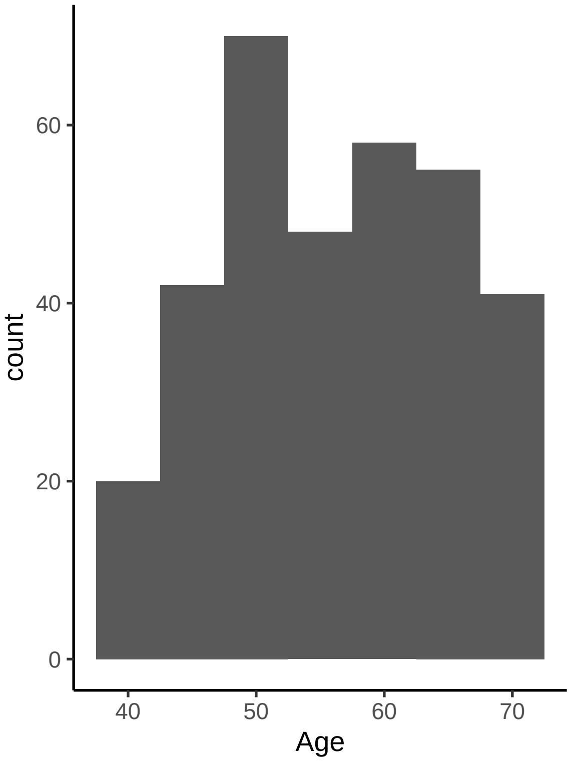


Figure 06: Height

Binwidth = 1 Inch

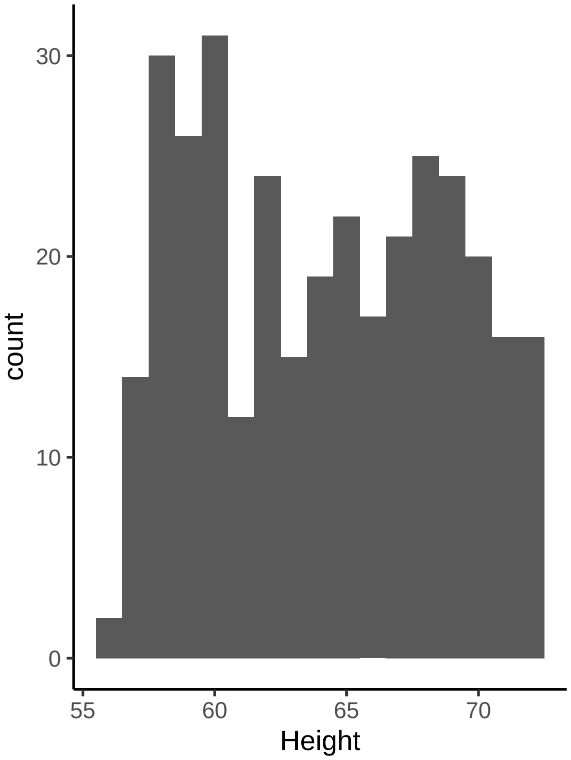


Figure 07: Handspan

Binwidth = 0.1 Inch

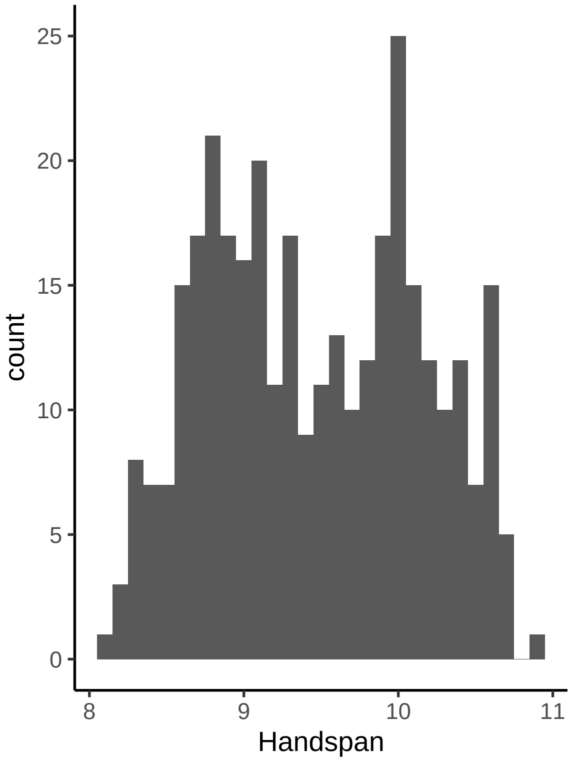


Figure 08 Pinky Length

Binwidth = 0.1 Inch

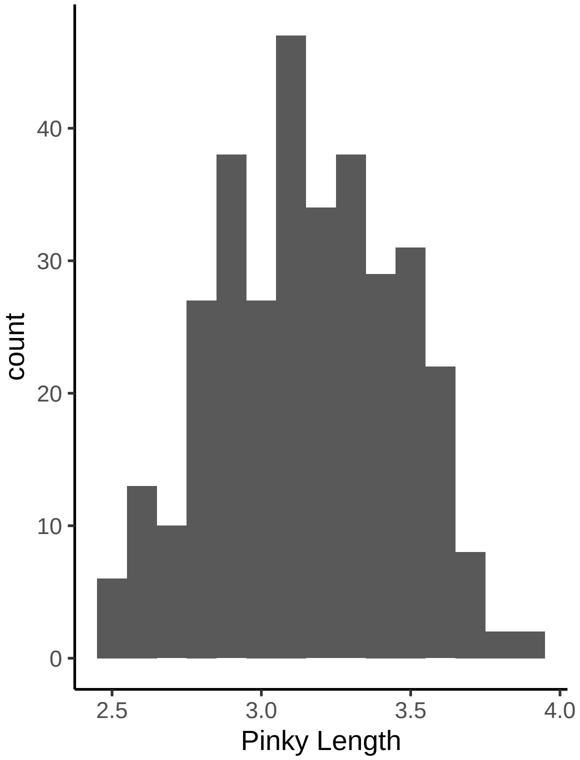


Fig-

ure 5, **Figure 6**, **Figure 7**, **Figure 8** all show the group distributions of various parameters

```
df <- ClassSurveysSP2021
Foodcount <- data.frame(names = c("Barley", "Corn", "Oates",
                                  "Rice", "Wheat", "Other"),
                        count = c(sum(df[6]),sum(df[7]),sum(df[8]),
                                  sum(df[9]),sum(df[10]),sum(df[11])))
Foodcount$count<-sort(Foodcount$count,decreasing=FALSE)
Foodcount
```

```
##    names count
## 1 Barley    13
## 2   Corn    20
## 3  Oates    34
## 4   Rice    51
## 5  Wheat    93
## 6  Other   101
```

```
g09 <- ggplot(data = Foodcount)+
  geom_bar(mapping = aes(x=names,y=count,fill=names),show.legend = FALSE)+
  geom_text(mapping =
              aes(x=names,
                  y=count,
                  label=count),
            check_overlap=TRUE,nudge_y=-5)+
    labs(title="Figure 09: What is the favorite food of everyone")+
  theme_classic()
```

```
g10 <- ggplot(data = df)+
  geom_histogram(mapping = aes(x=Caffeine),binwidth=1)+
    labs(title="The World's Most Popular Psychoactive Drug",
         subtitle="How Much Caffeine Do PLNU Students Intake")+
  theme_classic()
```

```
df.class=as.data.frame(table(df$Class))
df.class$Year=df.class$Var1
df.class$Frequency=df.class$Freq
#rewrite the data cleanly
df.class=data.frame(Year=c("Other","Freshman","Junior","Sophmore","Senior"),
                    Frequency=c(6,13,31,60,53),
                    ClassYear=c(5,1,3,2,4))
df.class <- df.class[order(df.class$ClassYear),]

g11 <- ggplot(data = df.class[order(df.class$ClassYear),])+
  geom_bar(mapping = aes(x=Year,y=Frequency,color=Year,fill=Year),
           stat='identity')+
    geom_text(mapping = aes(x=Year,y=Frequency, label=Frequency),
            check_overlap=TRUE,nudge_y=3)+
    labs(title="Figure 11: What Is Your Year",
         subtitle="Distribution of Class Year")+theme_classic()+
  theme(legend.position = 'bottom')
g11
```

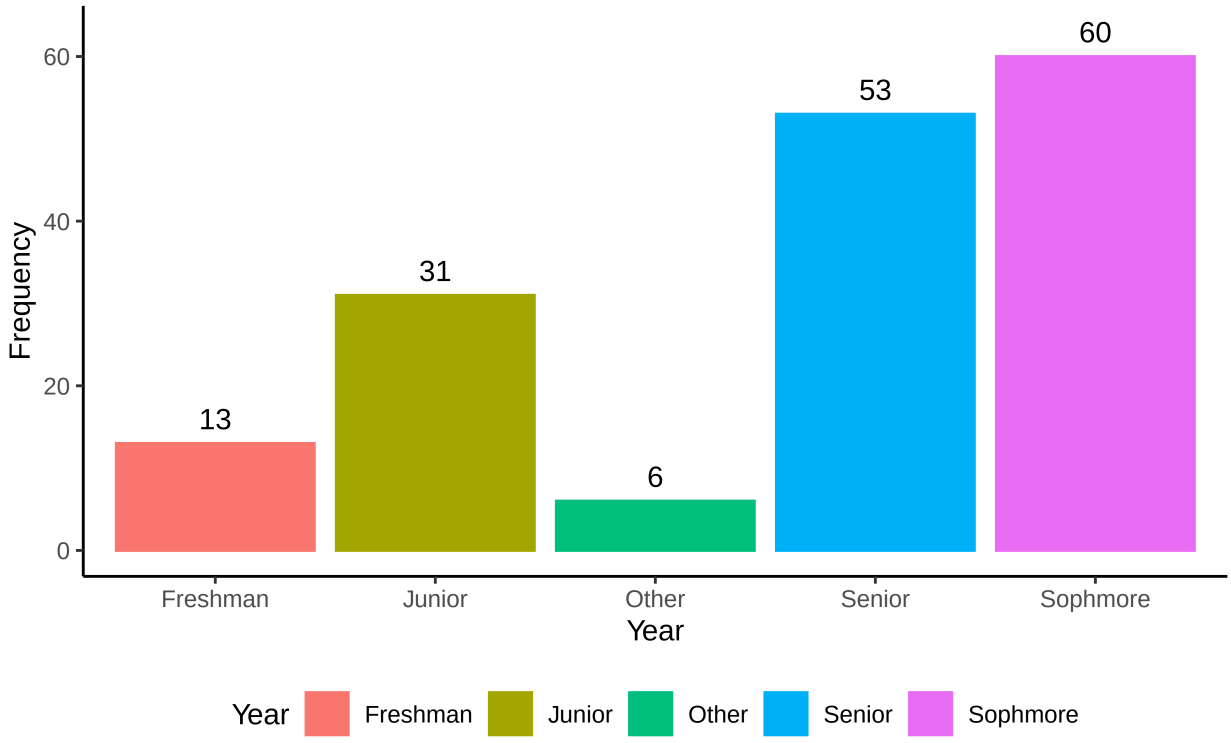
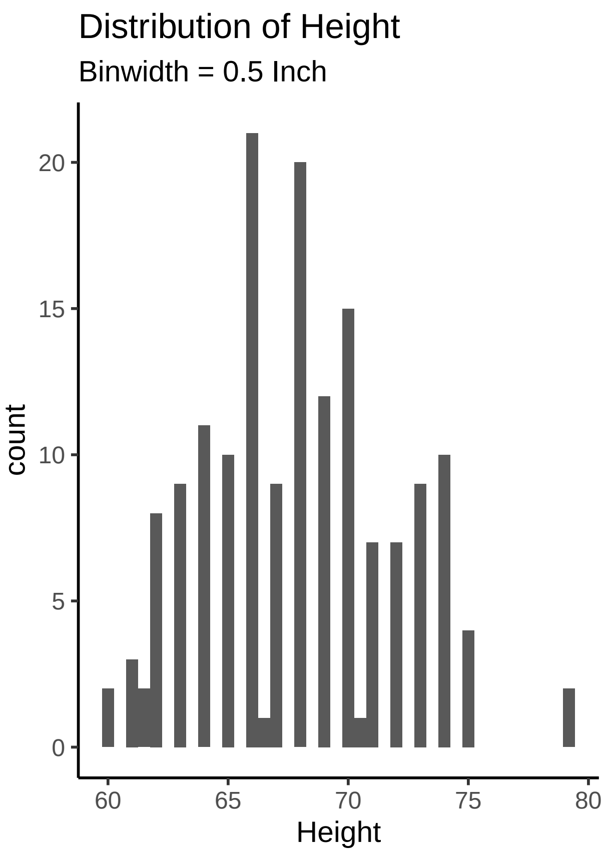
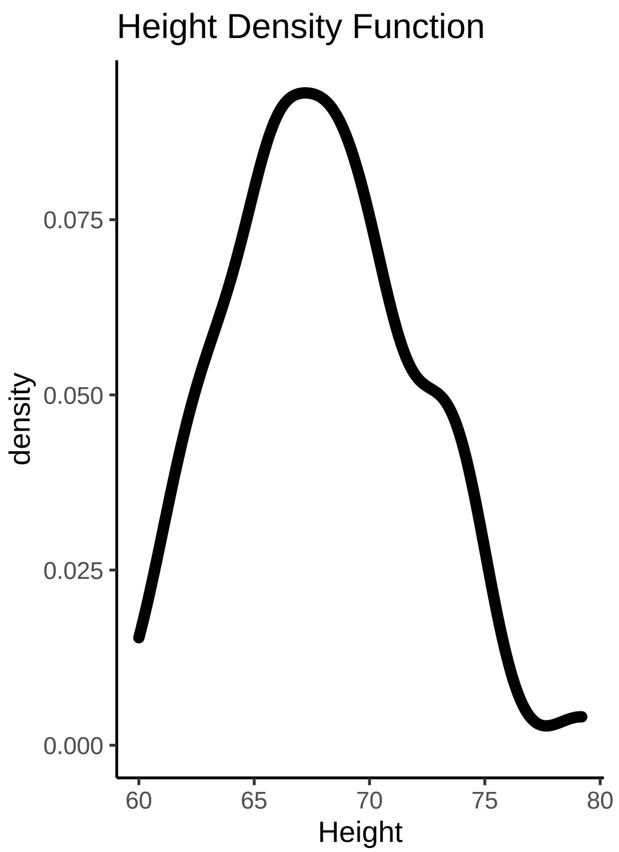


Figure 11: What Is Your Year

Distribution of Class Year

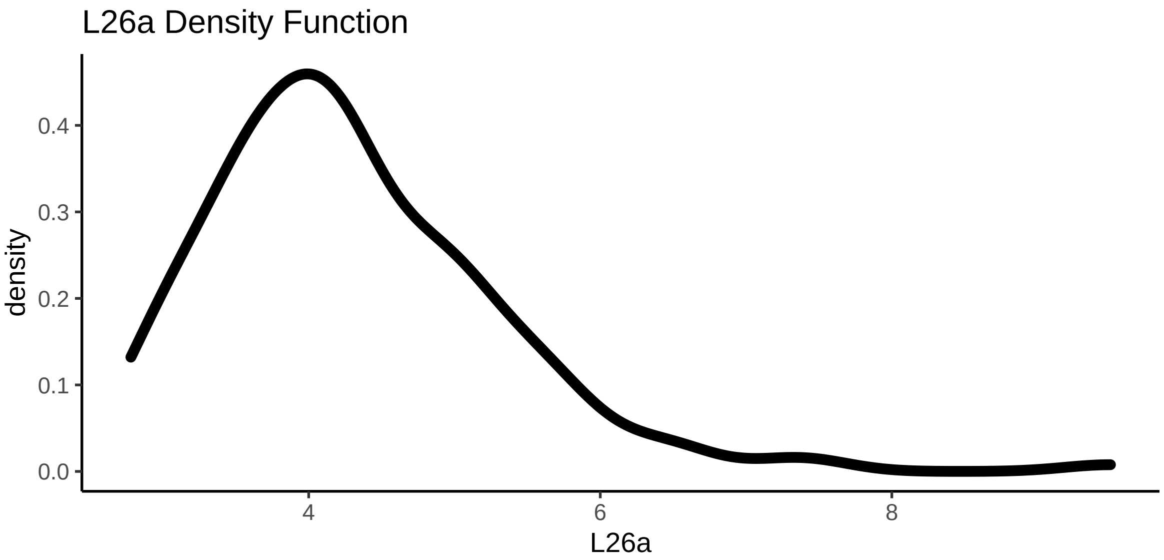
```
g12<-ggplot(data = df)+
  stat_count(mapping =aes(x=Height),width=0.5)+
  labs(title="Distribution of Height",
       subtitle='Binwidth = 0.5 Inch')+
  theme_classic()
g13 <- ggplot(data=df)+geom_density(mapping=aes(x=Height),size=2)+
  labs(title="Height Density Function")+theme_classic()
multiplot(g12,g13,cols=2)
```

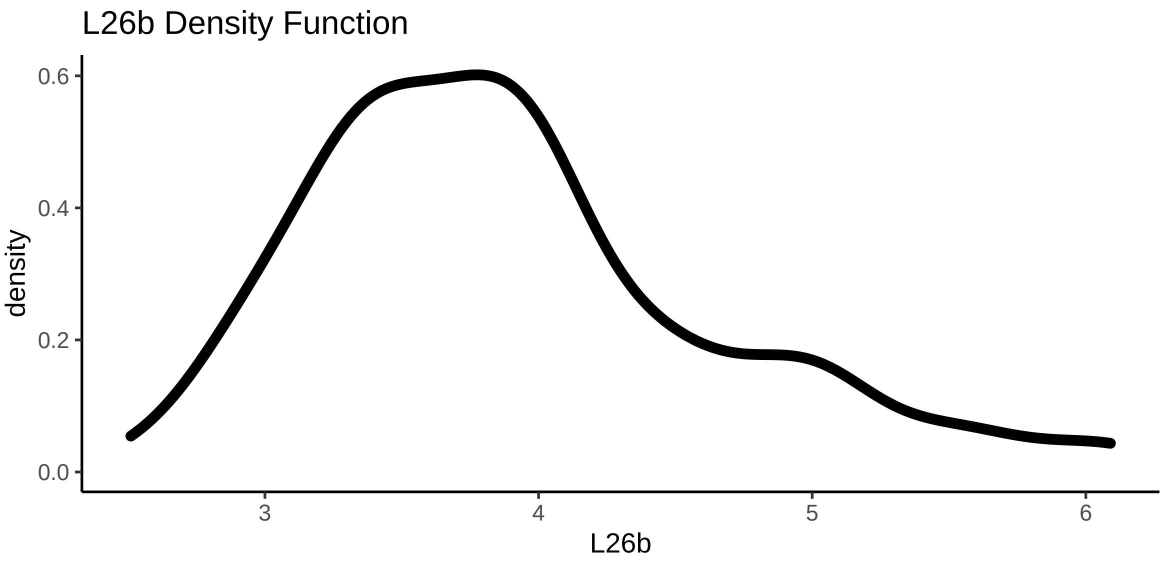


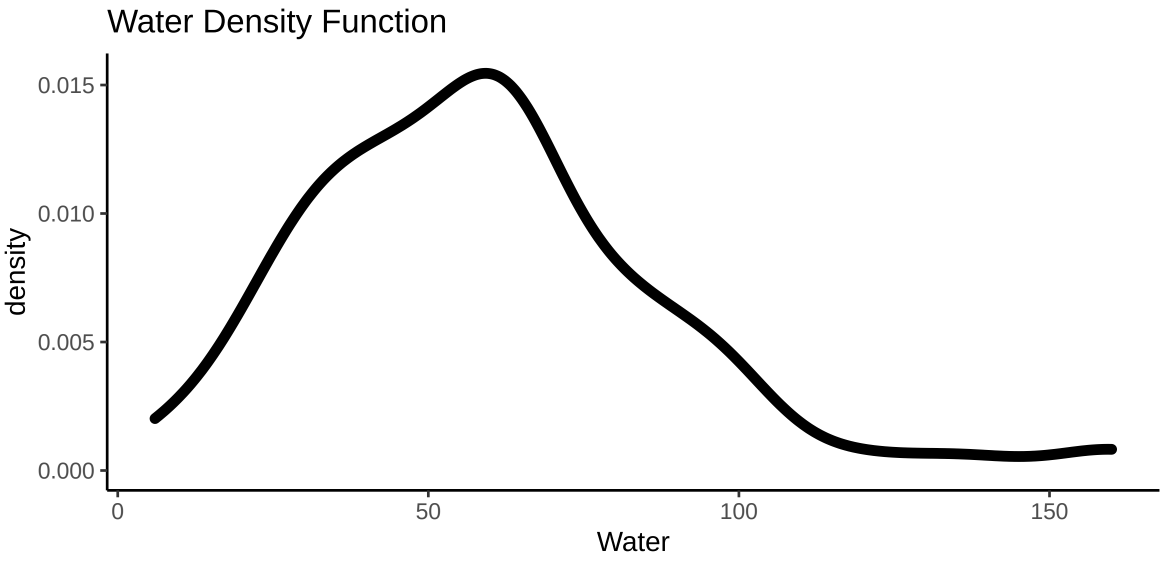


```
#This is just to clean up some of my variables
rm(list=c("g01","g02","g03","g04","g05","g06","g07","g08","g09","g10",
          "g11","g12","g13"))
```

```
g14<-ggplot(data = df)+
  geom_density(mapping=aes(x=L26a),size=2)+
  labs(title="L26a Density Function")+
  theme_classic()
g15<-ggplot(data = filter(df,L26b!=max(df$L26b)))+
  geom_density(mapping=aes(x=L26b),size=2)+
  labs(title="L26b Density Function")+
  theme_classic()
g16<-ggplot(data = df)+
  geom_density(mapping=aes(x=Water),size=2)+
  labs(title="Water Density Function")+
  theme_classic()

multiplot(g14,g15,g16,cols = 1)
```


L26a Density Function
density
0.0
0.1
0.2
0.3
0.4
4
6
8
L26a


L26b Density Function
density
0.0
0.2
0.4
0.6
3
4
5
6
L26b


Water Density Function
density
0.000
0.005
0.010
0.015
0
50
100
150
Water

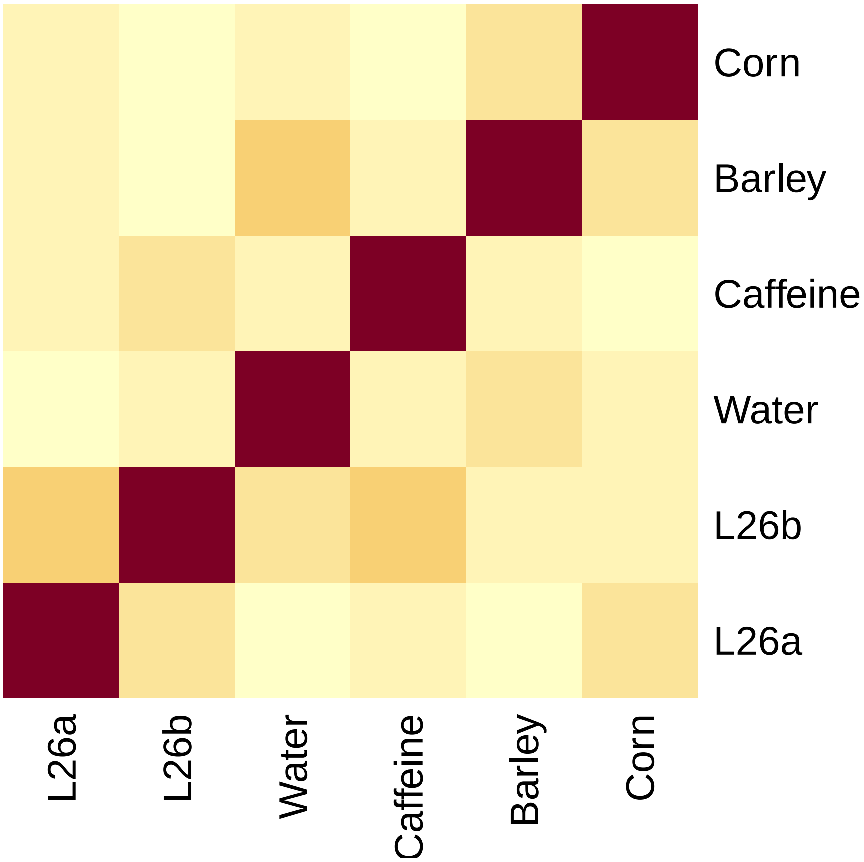
As

seen in **Figure 13 & Figure 14** the probability distribution of L26a and L26b are strongly correlated with each other. However, in **Figure 14**, it should be noted that an outlier was determined and removed.

```
cor(df[2:7])
```

```
##                 L26a        L26b       Water    Caffeine      Barley
## L26a      1.00000000  0.16415947 -0.18362596 -0.02429739 -0.03302442
## L26b      0.16415947  1.00000000  0.01367319  0.13702399 -0.02310715
## Water    -0.18362596  0.01367319  1.00000000 -0.07596182  0.12882783
## Caffeine -0.02429739  0.13702399 -0.07596182  1.00000000 -0.01381010
## Barley   -0.03302442 -0.02310715  0.12882783 -0.01381010  1.00000000
## Corn      0.01161253 -0.05309576 -0.03470164 -0.19395852  0.09438417
##                 Corn
## L26a      0.01161253
## L26b     -0.05309576
## Water    -0.03470164
## Caffeine -0.19395852
## Barley    0.09438417
## Corn      1.00000000
```

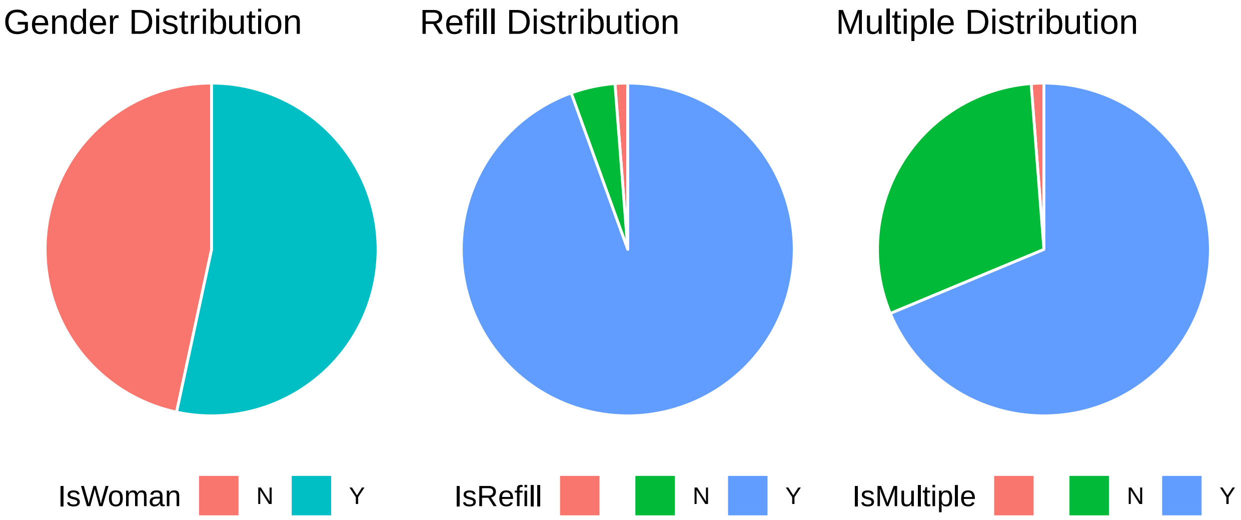
```
heatmap(cor(df[2:7]), Colv = NA, Rowv = NA, scale="column")
```



Correlation Table: In an attempt to understand the data, a correlation matrix was made using these variables. No strong relationships existed.

```
df1=as.data.frame(table(df$Woman))
df2=as.data.frame(table(df$Refill))
df3=as.data.frame(table(df$Multi))
colnames(df1)=c('IsWoman','Frequency')
colnames(df2)=c('IsRefill','Frequency')
colnames(df3)=c('IsMultiple','Frequency')
```

```
g16 <- ggplot(data=df1)+
         geom_bar(mapping =aes(x='',y=Frequency,fill=IsWoman),
                  width = 1, stat = "identity", color = "white")+
         coord_polar("y", start = 0)+
  labs(title='Gender Distribution')+
  theme_void()+theme(legend.position='bottom')

g17 <- ggplot(data=df2)+
         geom_bar(mapping =aes(x='',y=Frequency,fill=IsRefill),
                  width = 1, stat = "identity", color = "white")+
         coord_polar("y", start = 0)+
  labs(title='Refill Distribution')+
  theme_void()+
  theme(legend.position='bottom')

g18 <- ggplot(data=df3)+
         geom_bar(mapping =aes(x='',y=Frequency,fill=IsMultiple),
                  width = 1, stat = "identity", color = "white")+
         coord_polar("y", start = 0)+
  labs(title='Multiple Distribution')+
  theme_void()+
  theme(legend.position='bottom')
multiplot(g16,g17,g18,cols=3)
```


Gender Distribution
Refill Distribution
Multiple Distribution
IsWoman
N
Y
IsRefill
N
Y
IsMultiple
N
Y