Bio-Data Science Task 1 Report

Summary: In this week's task, I carried out two tasks to strengthen my skills inR-programming for data science.

First, I performed simple visualization on the data set "astrocyte_data". I created barplots, histograms, boxplots, added legends and labels. Next, I installed all the packages that would enable me perform ggplots on the data set "aflatoxin_data". I continued to melt data, stack barplots, create pie-chart, histogram and scatter plot while applying colors. I then created heatmaps which I manipulated severally to create what suits my taste.

In conclusion, this task has strengthened my skills in R-programmingfor statistical analysis/visualization.

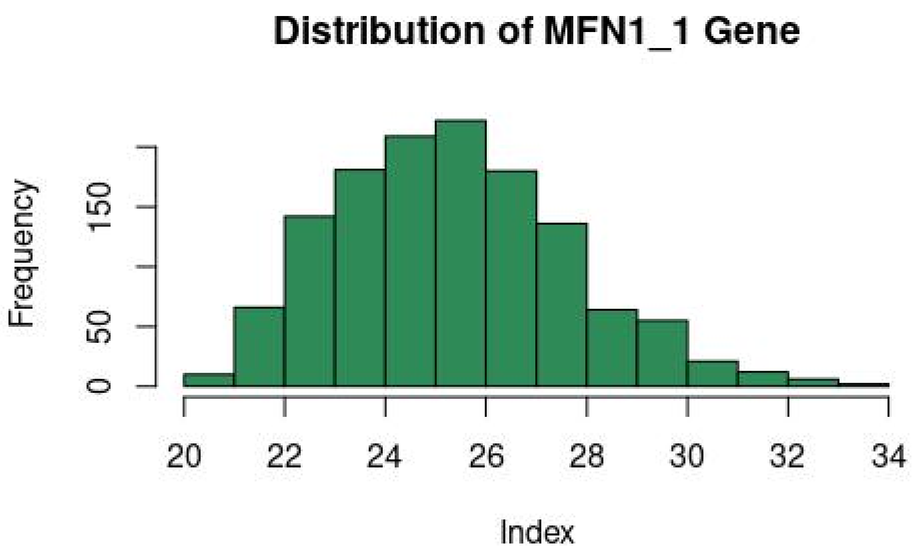
Explanations of codes and visualizations are provided below:

1. Mitochondria-endoplasmic reticulum contacts in reactive astrocytes promote vascular remodelling. Goebel et al. I want to perform the comparison of FACS-enriched astrocytes from uninjured and injured wild-type mice at different time points.I have already downloaded my data as .csv, next I will import my file in R

```
astrocyte_data <- read.csv(file.choose())
astrocyte_data
```

2. I want to perform Data Visualization in R. Let me construct a histogram for the distribution of MFN1_1 Gene

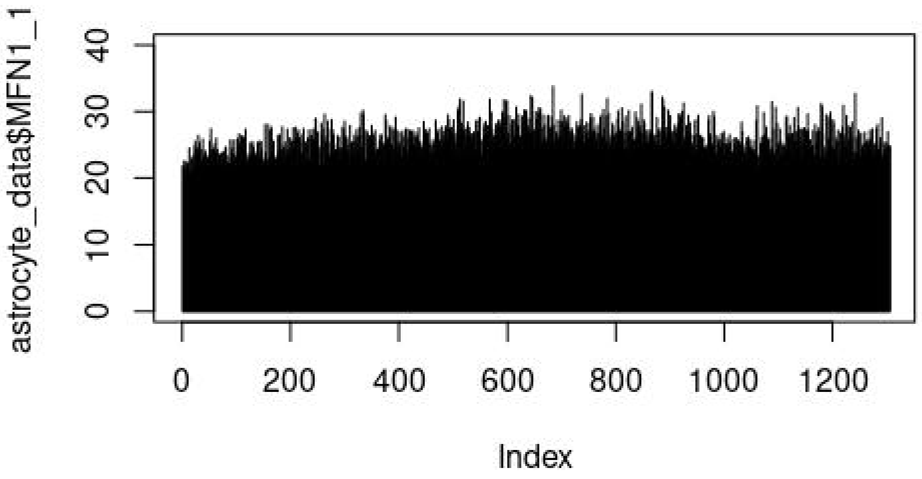
```
hist(astrocyte_data$MFN1_1, col = "Sea green", main = "Distribution of MFN1_1 Gene",
xlab = "Index")
```



Explanation: from this histogram, we see how MFN1_1 genes in the sample are spread over a range of frequency. Notice that multiple genes can not be used here.

3. I want to plot lineplots for the distribution of MFN1_1 Gene

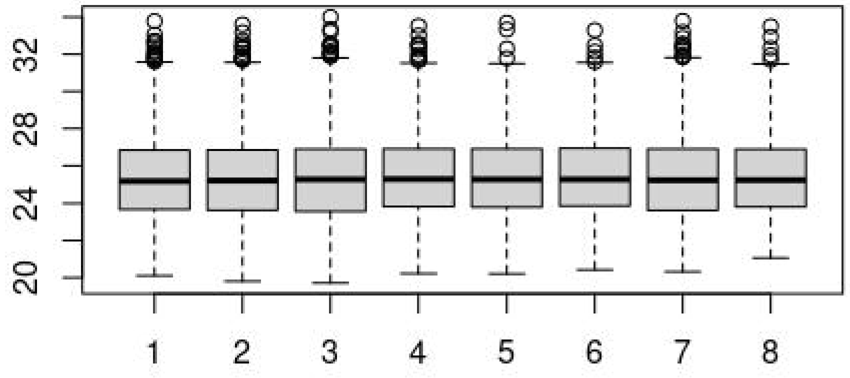
```
plot(astrocyte_data$MFN1_1, type = "h", xlim = c(0,1300), ylim = c(0, 40))
```



Explanation: though I have represented same information in the histogram above, I used line plot to make my information more individualized.

4. I want to create black and white boxplots for my genes

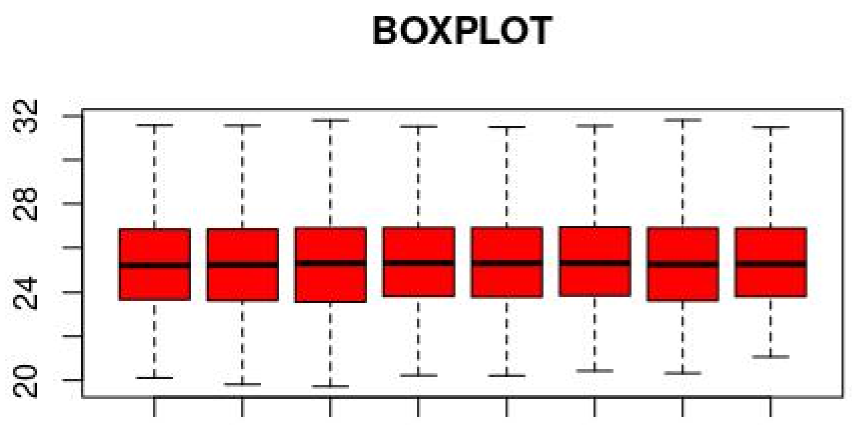
```
boxplot(astrocyte_data$MFN1_1, astrocyte_data$MFN1_2, astrocyte_data$MFN1_4,
astrocyte_data$WT_1, astrocyte_data$WT_3, astrocyte_data$WT_4, astrocyte_data$MFN1,
astrocyte_data$WT)
```



Explanation: Previously, we visualized for only one gene. I used boxplot to visualize the uninjured and injured wild-type mice at different time points

5. Let me add red color and title to my boxplot

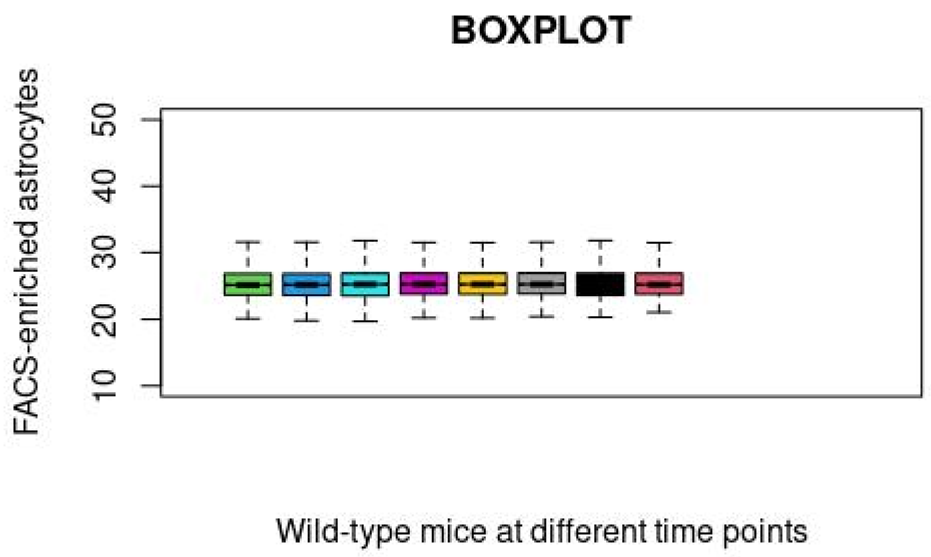
```
boxplot(astrocyte_data$MFN1_1, astrocyte_data$MFN1_2, astrocyte_data$MFN1_4,
astrocyte_data$WT_1, astrocyte_data$WT_3, astrocyte_data$WT_4, astrocyte_data$MFN1,
astrocyte_data$WT, col = 'red', outline = F, main = 'BOXPLOT')
```



Explanation: I added red color to the box plot and removed outline to make it more visually attractive.

6. I want to color by gene types

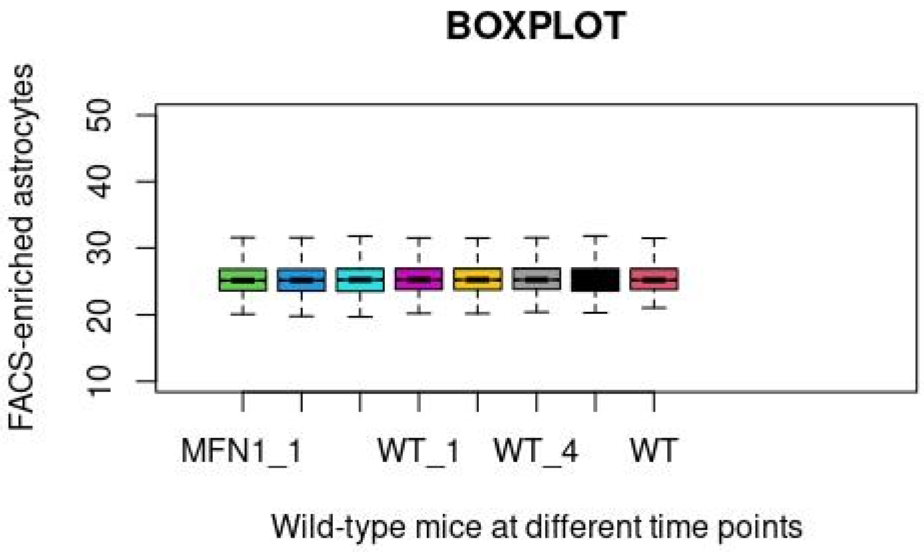
```
boxplot(astrocyte_data$MFN1_1, astrocyte_data$MFN1_2, astrocyte_data$MFN1_4, astrocyte_data$WT_1, astrocyte_data$WT_3, astrocyte_data$WT_4, astrocyte_data$MFN1, astrocyte_data$WT, col = 3:10, notch = T, outline = F, main = 'BOXPLOT', xaxt = 'n', xlab = 'Wild-type mice at different time points', ylab = 'FACS-enriched astrocytes', ylim = c(10,50), xlim = c(0,12))
```



Explanation: In order to distinguish my uninjured and injured wild-type mice at different time points on first sight, I added a different color to each of them.

7. I want to add axis

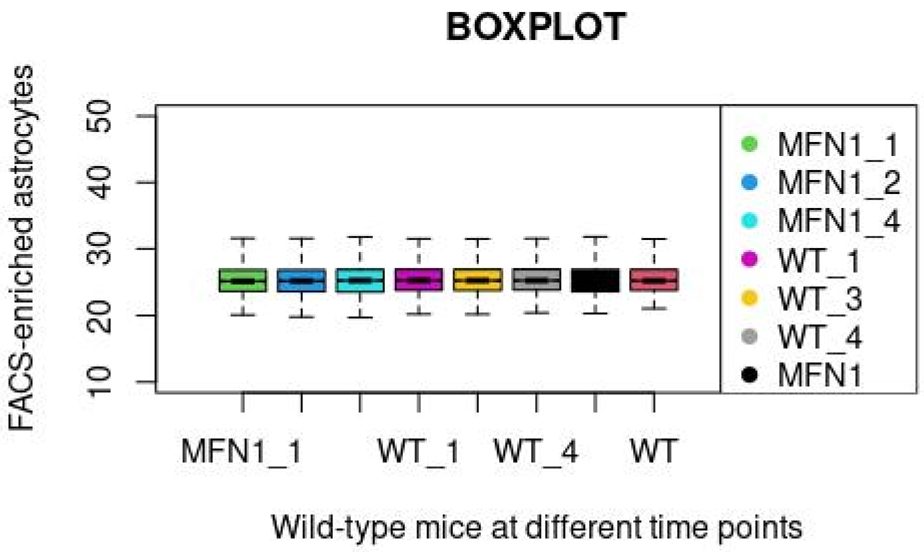
```
axis(side = 1, at = c(1,2,3,4,5,6,7,8), labels = c('MFN1_1', 'MFN1_2', 'MFN1_4', 'WT_1', 'WT_3', 'WT_4', 'MFN1', 'WT'))
```



Explanation: I added axis to enable viewers to understand that I am performing the comparison of FACS-enriched astrocytes from uninjured and injured wild-type mice at different time points.

8. Finally I will add legend

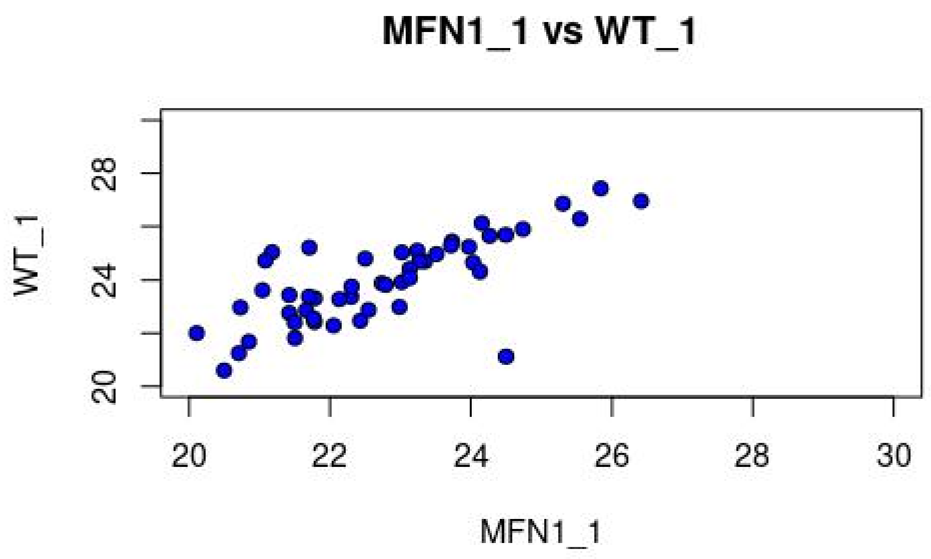
```
legend('topright', legend = c('MFN1_1', 'MFN1_2', 'MFN1_4', 'WT_1', 'WT_3', 'WT_4', 'MFN1', 'WT'), col = 3:10, pch = 19)
```



Explanation: to understand the mice type that is represented by each color, I have added legends.

9. I want to create a scatterplot of MFN1_1 vs WT_1. I will color it black with a background of blue.

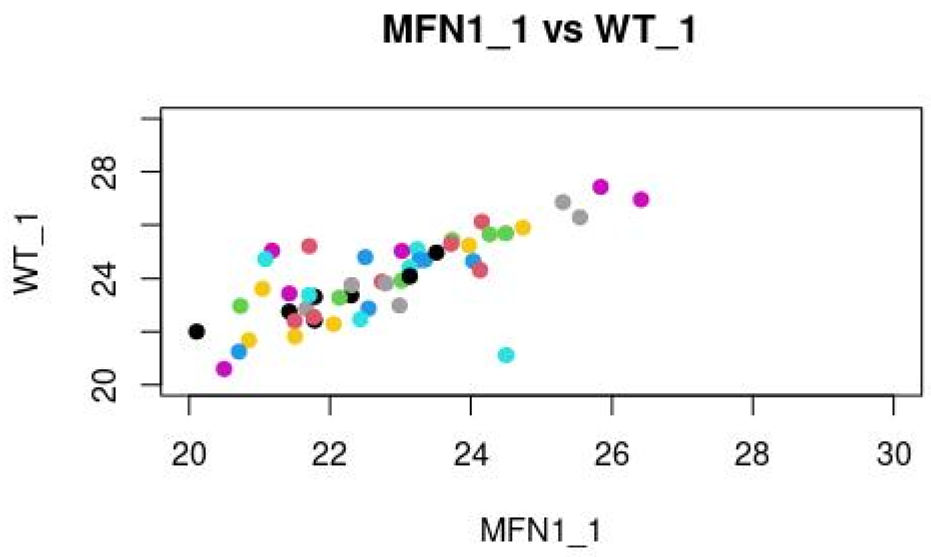
```
plot(x = astrocyte_data$MFN1_1[1:50], y = astrocyte_data$WT_1[1:50], col = 'black', pch = 21, bg = 'blue', main = 'MFN1_1 vs WT_1', xlab = 'MFN1_1', ylab = 'WT_1', xlim = c(20,30), ylim = c(20,30))
```



Explanation: I have now compared 2 mice types using scatter plot.

10. I want to color by gene type.

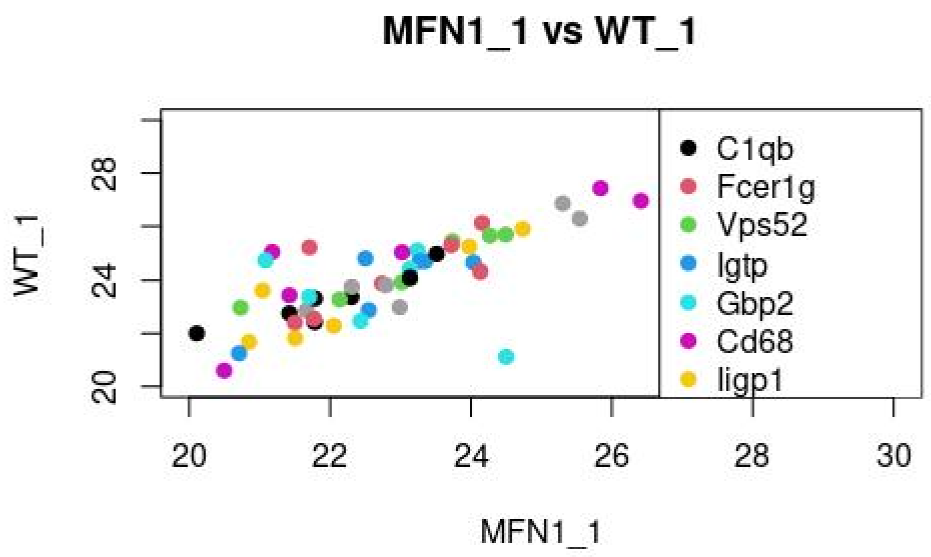
```
plot(astrocyte_data$MFN1_1[1:50], astrocyte_data$WT_1[1:50], col = 1:50, pch = 19, main
= 'MFN1_1 vs WT_1', xlab = 'MFN1_1', ylab = 'WT_1', xlim = c(20,30), ylim = c(20,30))
```



Explanation: coloring by genes has enabled me to visualize the various genes that exist in the comparison.

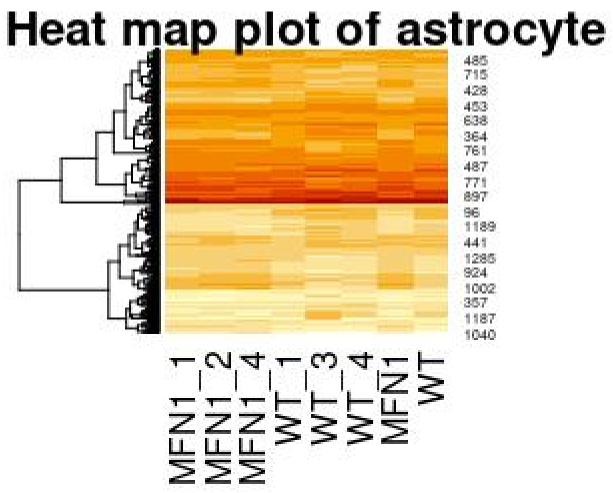
11. Let me add legend to my plot

```
legend('topright', legend = c(astrocyte_data$T..Gene.names[1:50]), pch = 19, col = 1:50)
```



Explanation: I performed this step to understand the specific gene each color represents

12. I plot the heatmap next.
heatmap(as.matrix(astrocyte_data[2:9]), Colv = NA, scale = 'col', margins = c(10,10), main = 'Heat map plot of astrocyte')



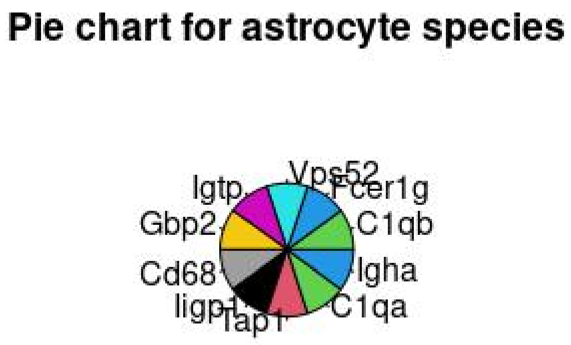
Explanation: I performed the heat map step in order to be able to visualize everything in my sample with just one visualization.

13. Let me now plot Pie Charts

```
table(astrocyte_data$T..Gene.names[1:10])
item <- unique(astrocyte_data$T..Gene.names[1:10])
itemCount <- as.vector(table(astrocyte_data$T..Gene.names[1:10]))
```

14. I will start by plotting a simple pie chart for astrocyte species

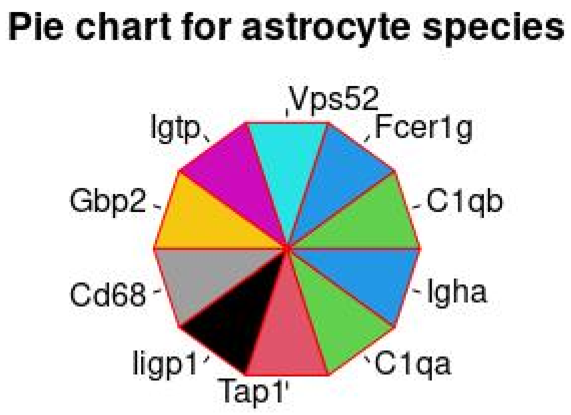
```
pie (x = itemCount, labels = item, radius = 0.5, col = 11:20, main = 'Pie chart for astrocyte species')
```



Explanation: this step shows how each astrocyte specie is distributed in the sample.

15. Next, let me add edges and color my borders

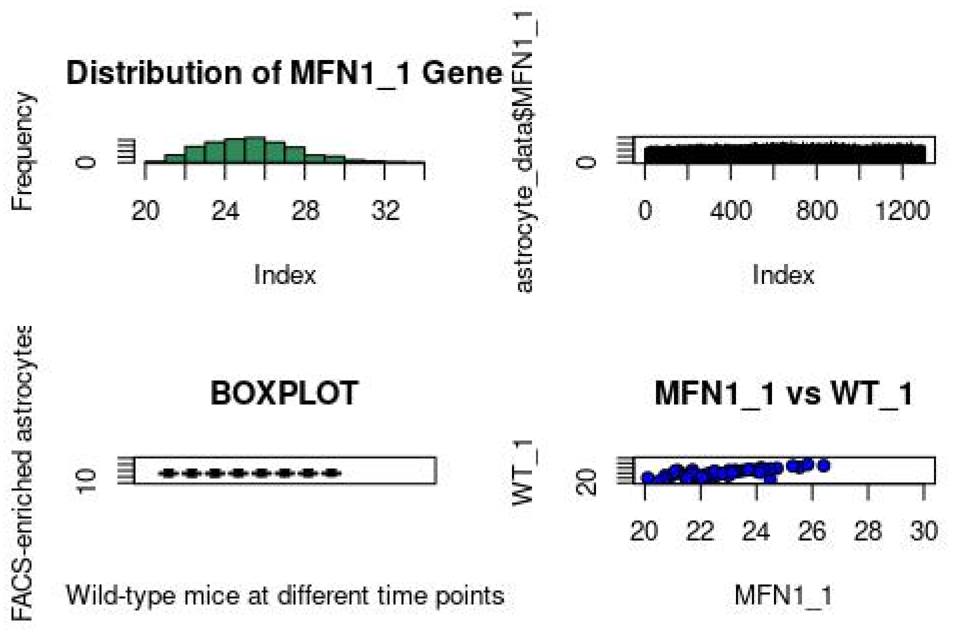
```
pie (x = itemCount, labels = item, radius = 1.0, edge = T, border = 'red', col = 11:20, main = 'Pie chart for astrocyte species')
```



Explanation: adding edges and border makes my plot clearer and more appreciable.

16. Let's practice how to arrange some of these plots as we usually see in publications

```
par(mfrow = c(2,2))
```



Explanation: this step is necessary because every scientist should know how to prepare plots that can be sent for publication.

LET US NOW GO FULLY INTO GGPLOTS

17. We are going to work with a new dataset

About data set: Growth and Toxigenicity of A. flavus on Resistant and Susceptible Peanut Genotypes. This study seeks to determine the reaction of peanut genotypes to Aflatoxigenic and non-aflatoxigenic A. flavus inoculation and also determine the mechanisms of their resistance. It was established that non-aflatoxigenic A. flavus grows faster than aflatoxigenic A. flavus. There was no significant difference in the incidence and severity of the A. flavus resistant genotypes (L027B and ICGV-03401) however, there were significant differences between resistant genotypes and the susceptible check Manipinta. This study also confirmed that non-aflatoxigenic A. flavus inoculation did not lead to aflatoxin production. Non-aflatoxigenic A. flavus identified could serve as a good biocontrol against aflatoxin contamination under field conditions. Additionally, peanut genotypes with resistance to post-harvest aflatoxin accumulation will resist the growth of A. flavus and subsequent aflatoxin accumulation.

18. Install ggplot2

```
install.packages("ggplot2")
library(ggplot2)
```

19. Import .csv file in R

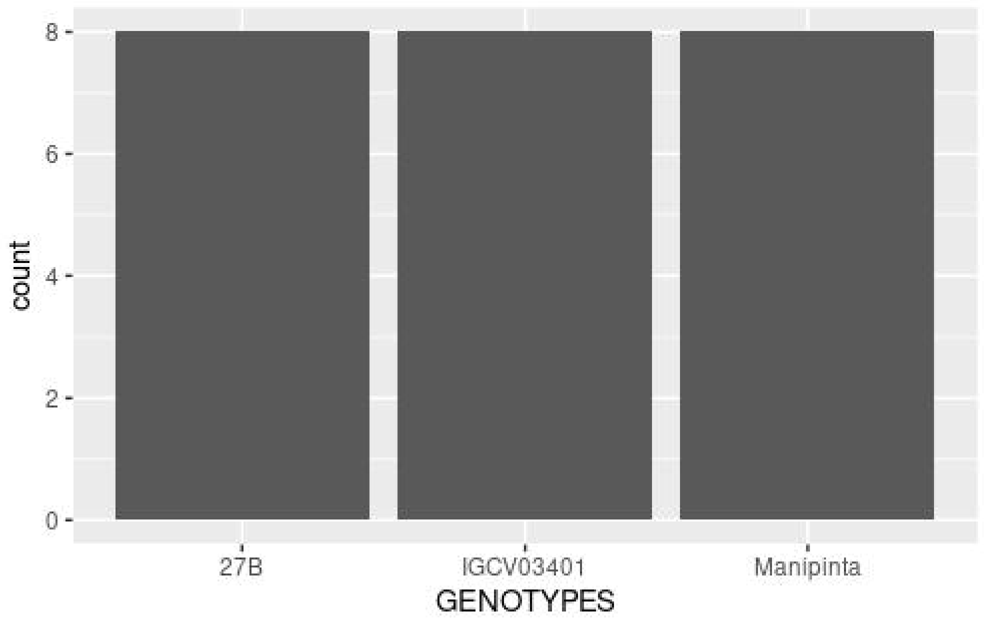
```
aflatoxin_data <- read.csv(file.choose())
aflatoxin_data
```

20. Start with defining the base data for ggplot

```
pl <- ggplot(data = aflatoxin_data)
```

21. To the base, I will add what I want to plot and color

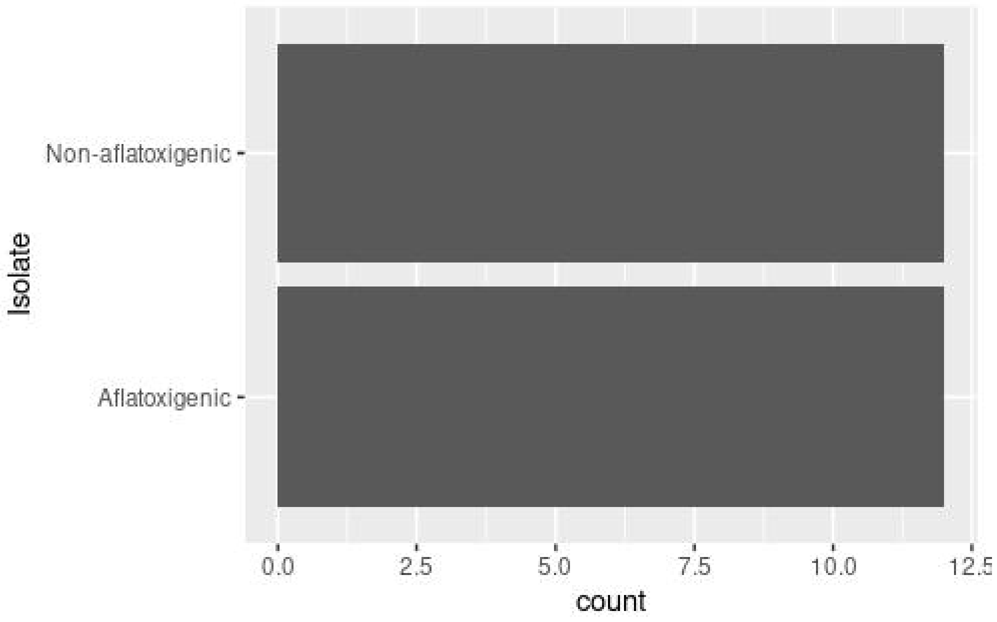
```
pl + geom_bar(aes(x=GENOTYPES))
```



Explanation: I created the bar plot to show the distribution of the genotypes of the given aflatoxin species.

22. I will plot frequency of the data from a single column

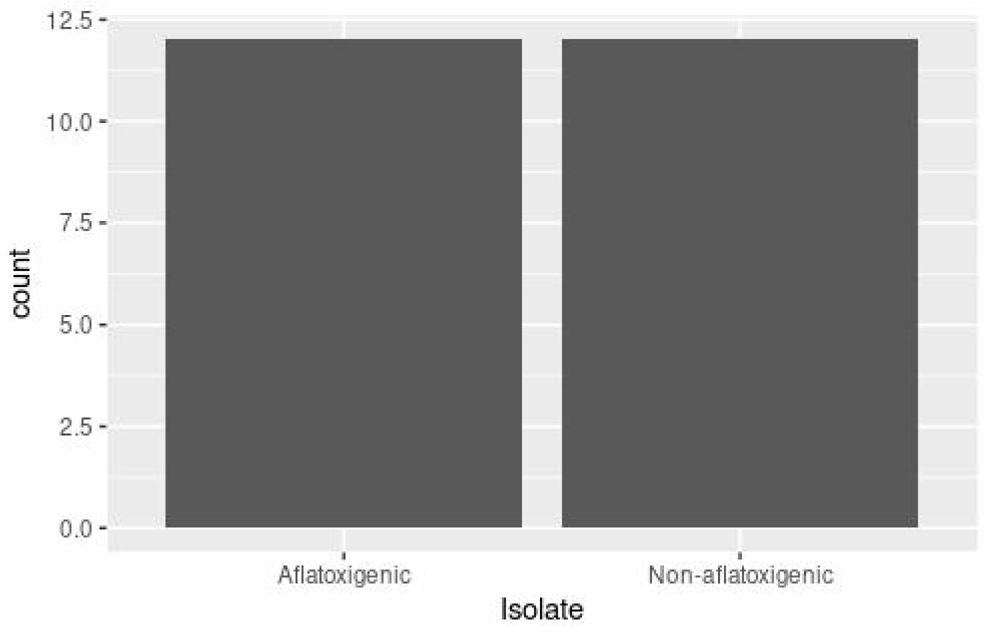
```
pl + geom_bar(aes(y=Isolate))
```



Explanation: It is important to distinguish between non-aflatoxigenic and aflatoxigenic species. We can see that they are equal in count.

23. Let me flip the coordinates

```
pl + geom_bar(aes(y=Isolate))+coord_flip()
```



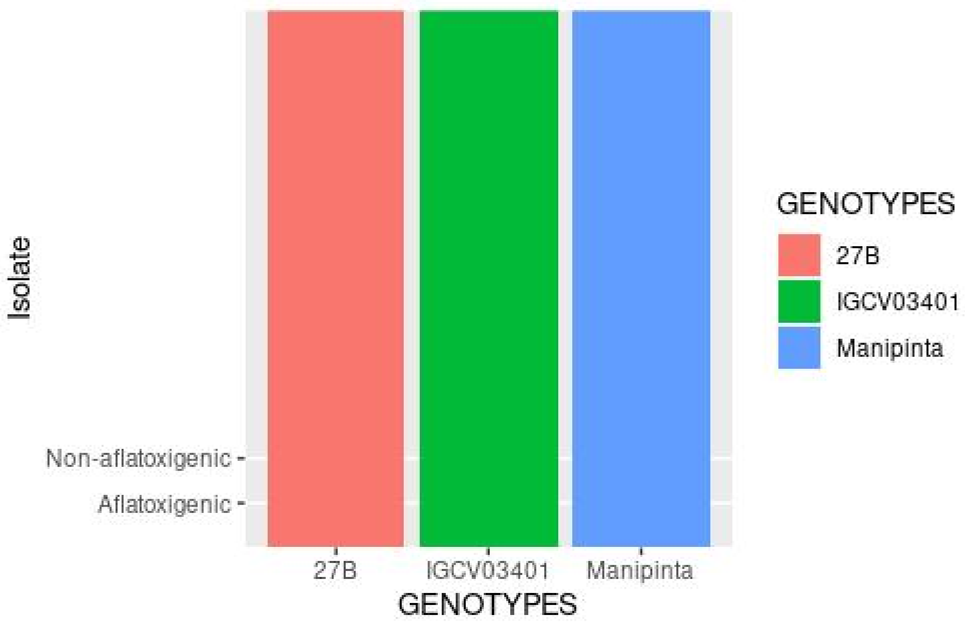
Explanation: I flipped my barplot so as to make it easier to view while comparing the count.

24. See the frequency of each variable within the species. Install.package(reshape2). Melt the data to have something understandable by ggplot better.

```
install.packages("reshape2")
install.packages("Rcpp")
library(reshape2)
melted_aflatoxin_data <- melt(aflatoxin_data)
```

25. I want to produce a ttacked/continuous bar plot for genotypes.

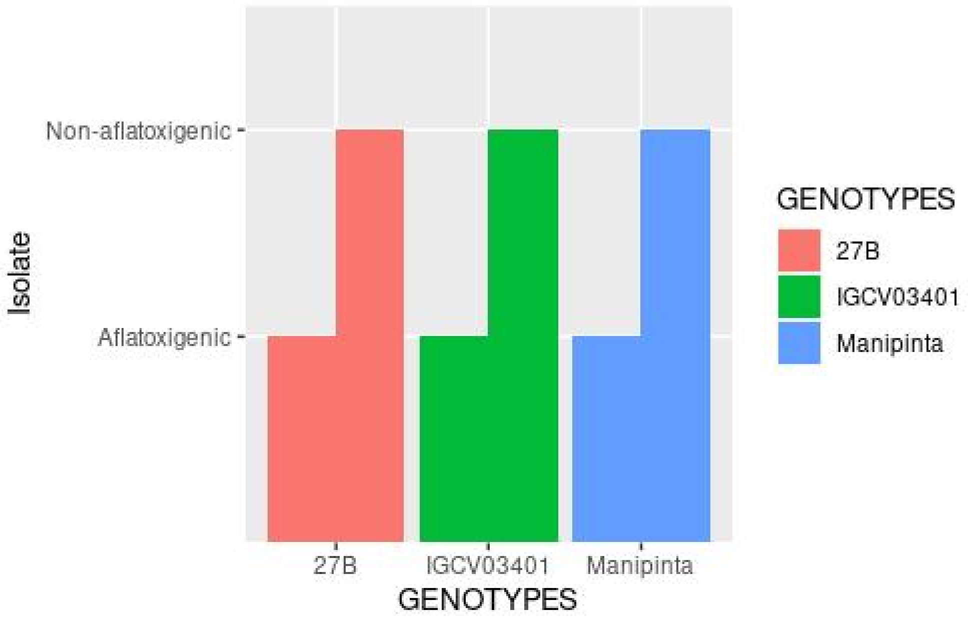
```
ggplot(data = aflatoxin_data) + geom_bar(aes(x=GENOTYPES, y=Isolate, fill = GENOTYPES), stat = "identity")
```



Explanation: This plot is necessary to to show how my 3v give genotypes compare to each other in terms of distribution.

26. Let me produce a multiple bar plot

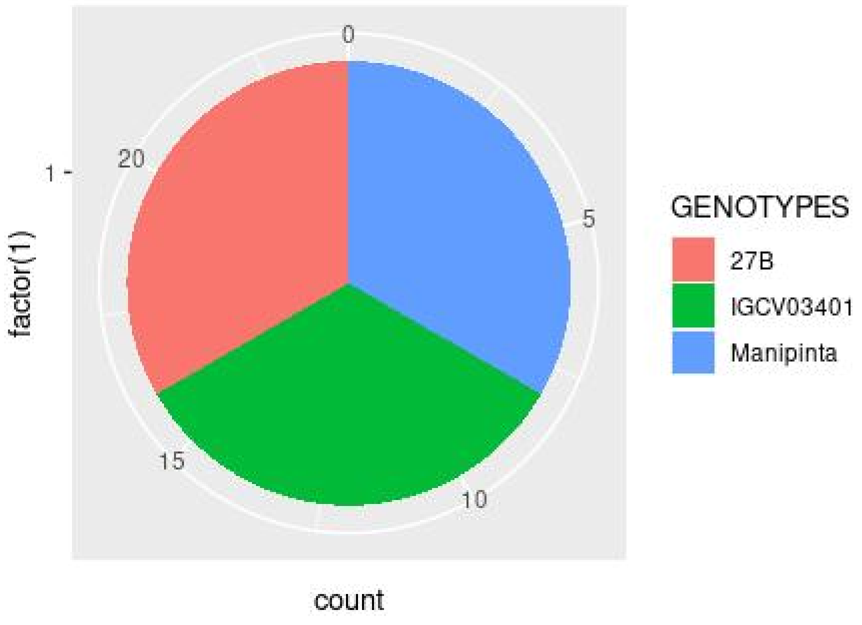
```
ggplot(data = aflatoxin_data) + geom_bar(aes(x=GENOTYPES, y=Isolate, fill =
GENOTYPES), stat = "identity", position = 'dodge')
```



Explanation: this time we are using both genotypes and the and toxicity as variables to compare our isolates at the same time.

27. Let me plot a simple pie chart and label by genotypes

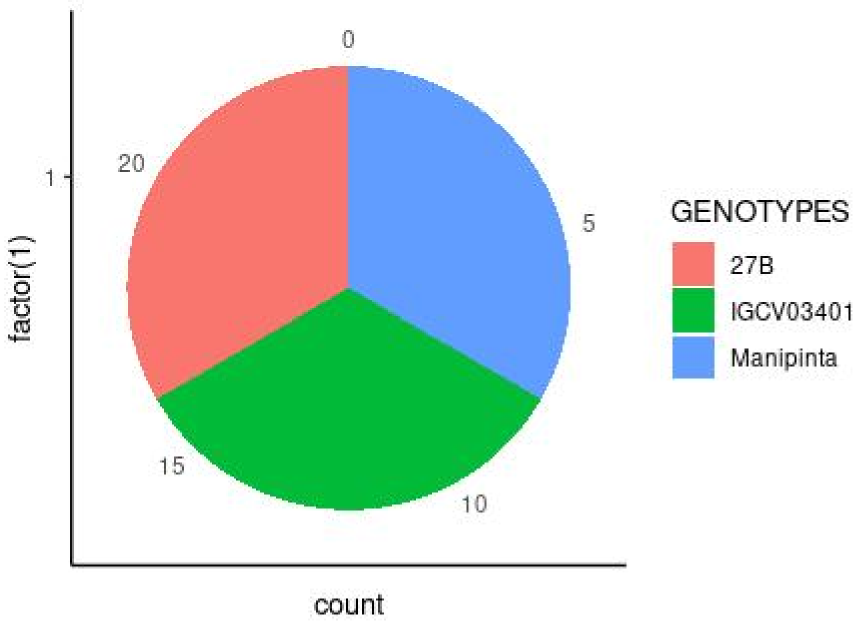
```
ggplot(data = aflatoxin_data) + geom_bar(aes(x=factor(1), fill = GENOTYPES), width = 1) +
coord_polar(theta = 'y')
```



Explanation: I compared my counts of genotypes by using pie chart. This showed that all genotypes used are of same quantity.

28. I will add a theme of my choice

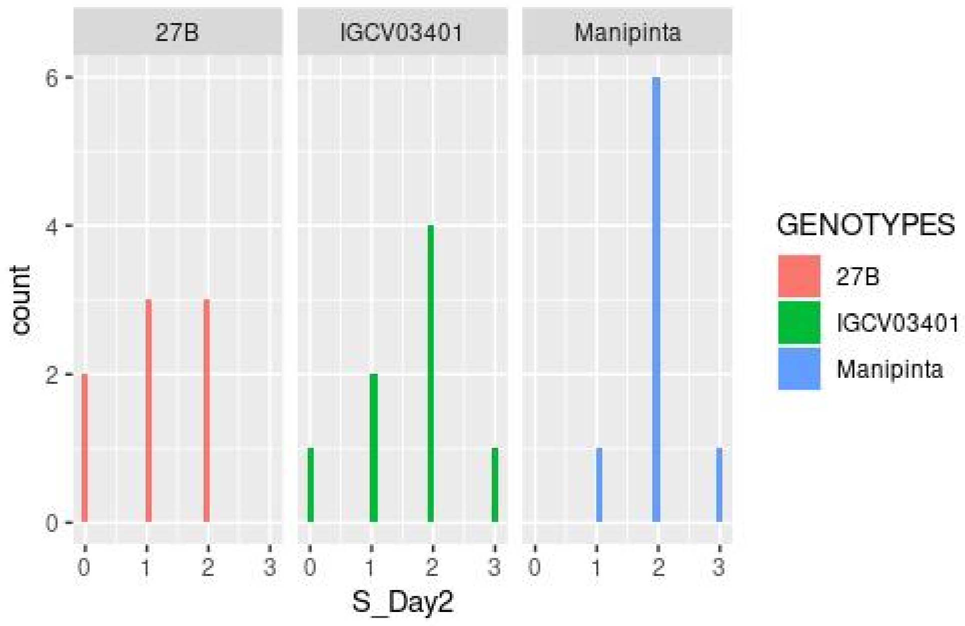
```
ggplot(data = aflatoxin_data) + geom_bar(aes(x=factor(1), fill = GENOTYPES), width = 1) +
coord_polar(theta = 'y') + theme_classic()
```



Explanation: I used theme to make my visualization more appealing.

29. Let me plot a simple histogram for aflatoxin_data

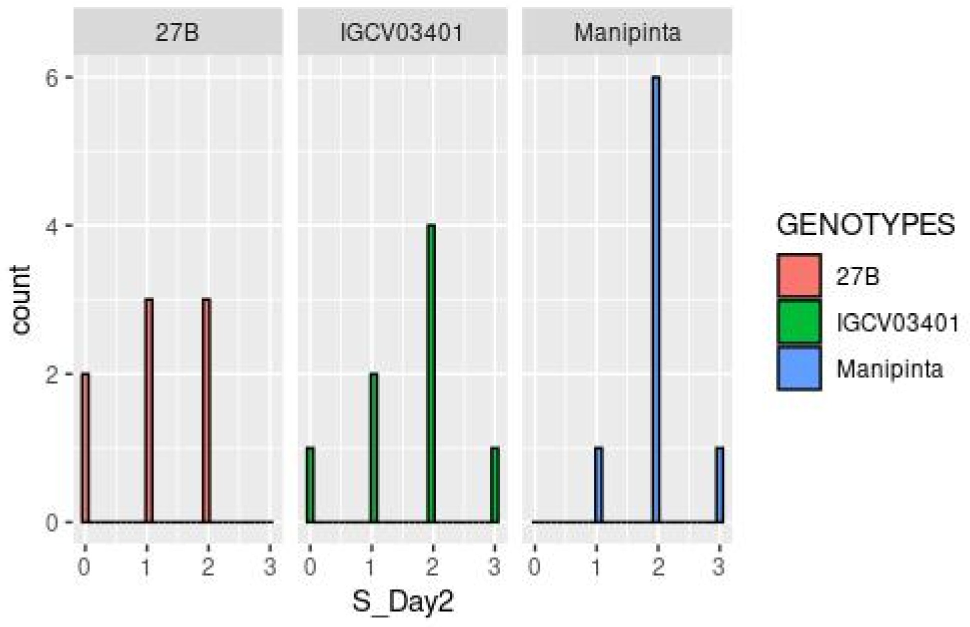
```
pl <- ggplot(data = aflatoxin_data)
pl + geom_histogram(aes(x=S_Day2, fill = GENOTYPES)) + facet_grid(. ~GENOTYPES)
```



Explanation: I used my histogram to compare the various genotypes that existed in Day2 by counts.

30. I will add a color parameter

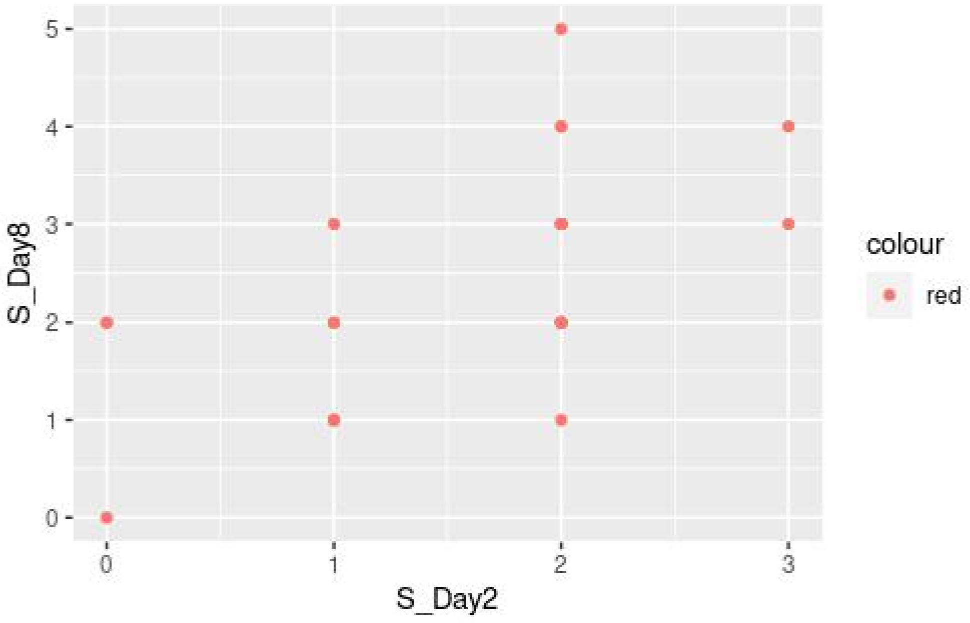
```
pl + geom_histogram(aes(x=S_Day2, fill = GENOTYPES), color = "black") + facet_grid(.
~GENOTYPES)
```



Explanation: I added color parameter to make my plots more conspicious.

31. Let me construct a Scatter plot of S_Day8 against S_Day2 with red color

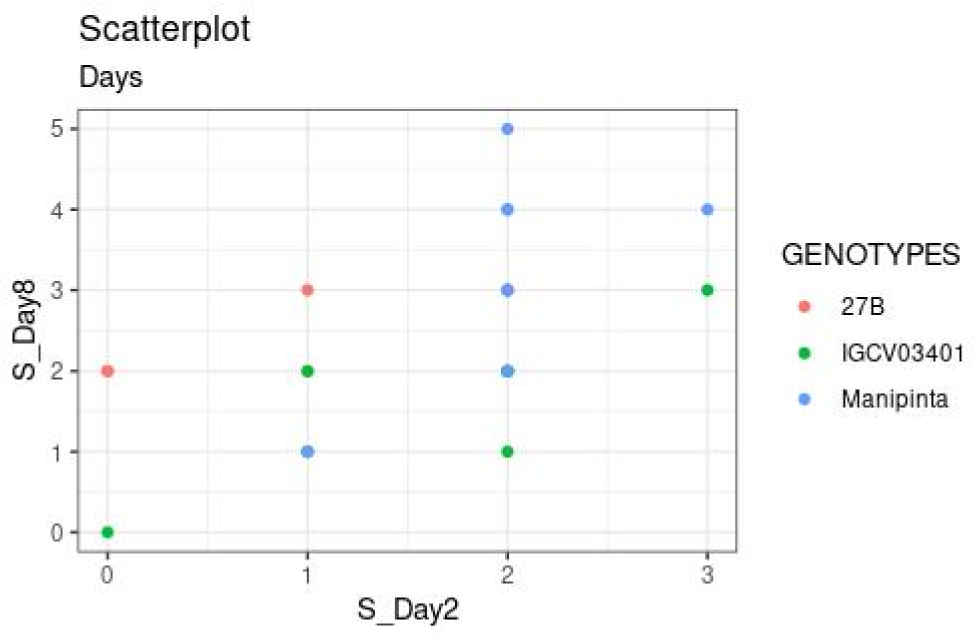
```
pl + geom_point(aes(x = S_Day2, y = S_Day8, color = 'red'))
```



Explanation: I used this scatter plot to compare genotype counts in Day2 and Day8.

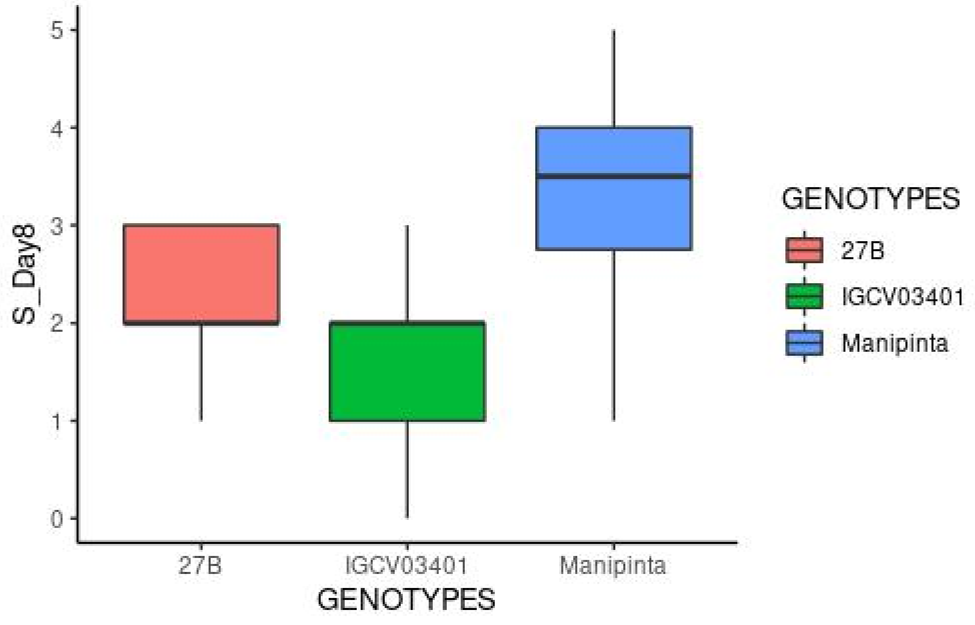
32. I will Color by Genotype (quite automated)

```
pl + geom_point(aes(x = S_Day2, y = S_Day8, color = GENOTYPES)) + theme_bw() +
ggtitle(label = 'Scatterplot', subtitle = 'Days')
```



Explanation: by adding colors to scatter plot base on genotypes, I can easily distinguish btween the varios genotypes that exist in the scatterplot.

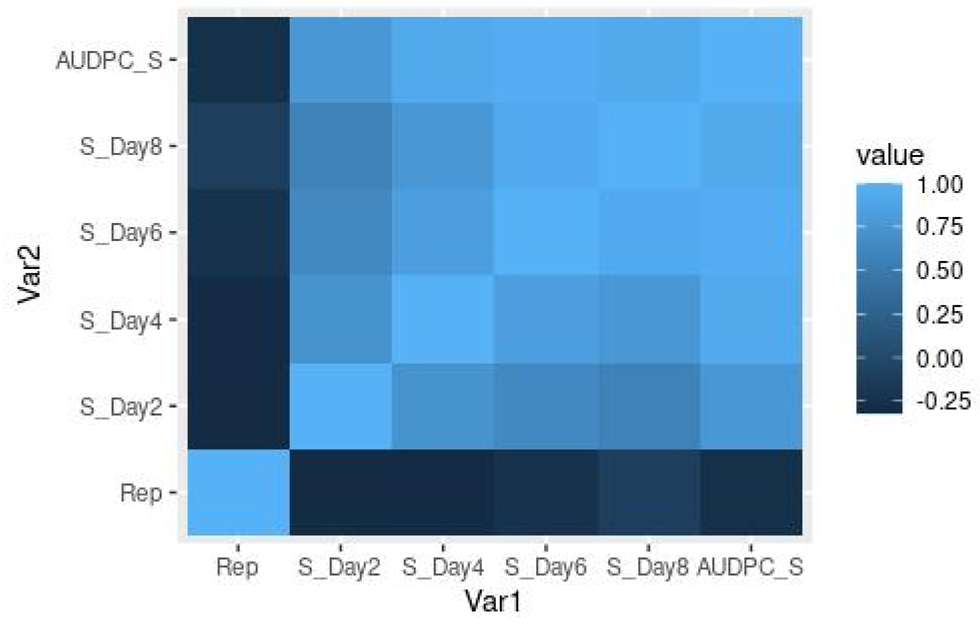
33. I will construct a boxplot plus a theme
pl + geom_boxplot(notch = F, aes(x= GENOTYPES, y = S_Day8, fill = GENOTYPES)) + theme_classic()



Explanation: I plotted boxplot in order to easily visualize the distribution of genotypes in the sample on Day8 (which is the final day).

34. Heat map plotting. I will use melting methods to create heat map with ggtiles.
meltCorData <- melt(cor(aflatoxin_data[3:8]))

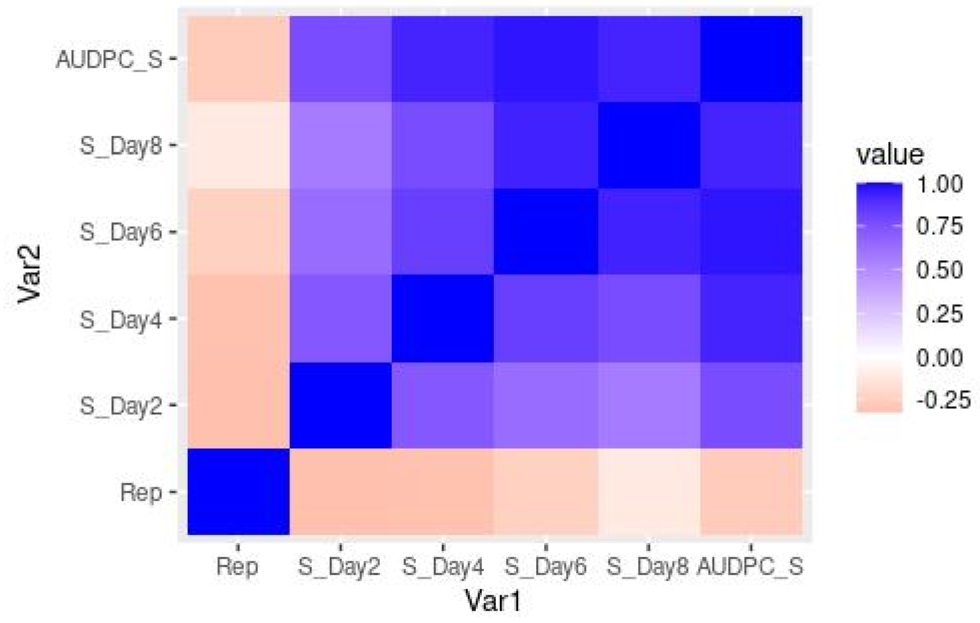
35. I start by setting my new ggplot
hm <- ggplot(data = meltCorData)
hm + geom_tile(aes(x = Var1, y = Var2, fill = value))



Explanation: we have several variables to compare. In order to visualize all of them at once, I used heat map.

36. Let me start changing colors in heatmaps

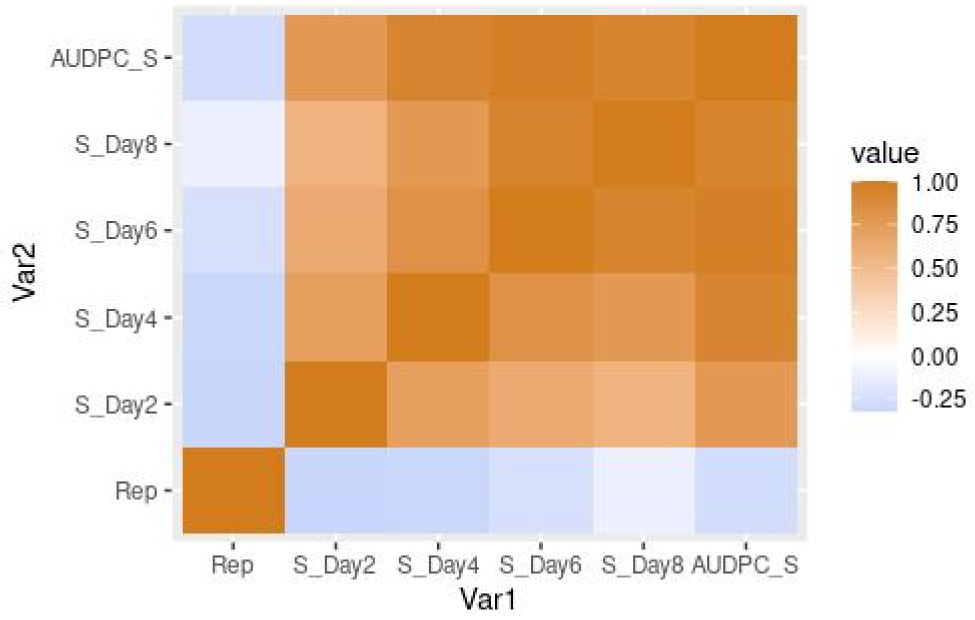
```
hm + geom_tile(aes(x = Var1, y = Var2, fill = value)) + scale_fill_gradient2(low = 'red', high = 'blue')
```



Explanation: I performed this step in order to master the ability to produce heatmaps of different colors.

37. I will also use hex code

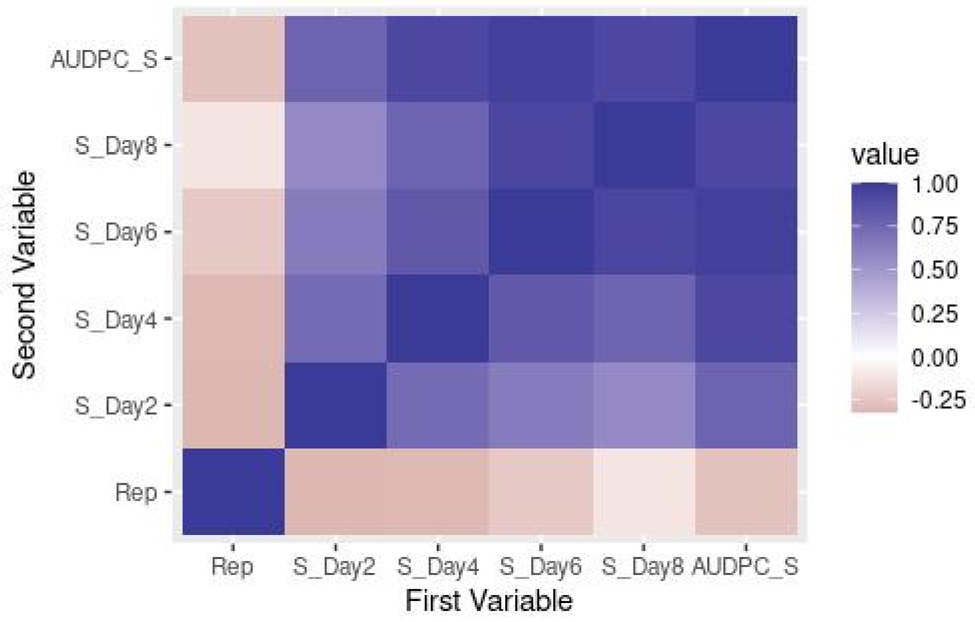
```
hm + geom_tile(aes(x = Var1, y = Var2, fill = value)) + scale_fill_gradient2(low = '#1687ee', high = '#d27C1c')
```



Explanation: we can see more color representation with 'hex code'. A lot can be done with R.

38. Let me change my x and y labels

hm + geom_tile(aes(x = Var1, y = Var2, fill = value)) + scale_fill_gradient2() + xlab('First Variable') + ylab('Second Variable')



Explanation: This step is necessary to show us what our x and y column represent (i.e first and second variable respectively).