# Assignment1

Yuanhang Yang

10/03/2023

# Task 1 - Literature

1a. What is the medically relevant insight from the article?

The research identifies step-wise downstream targets in the insulin signaling pathway, shedding light on metabolic diseases linked to insulin resistance, such as obesity and diabetes. 1b. Which genomics technology/ technologies were used? Phosphoproteome and metabolome

2a. List and explain at least three questions/ hypotheses you can think of that extend the analysis presented in the paper. (1) Modelling the phosphorylation site S775 of PFKL by phospohorylase; (2) Study the enzyme activity kinetics; (3) Bench mark different data quality and fold change cutoff as mentioned in discussion.

# Task 4

1. Use the R internal CO2 dataset ("data(CO2)").

```
data(CO2)
```

2. Describe briefly the content of the CO2 dataset using the help function.

```
help(data(CO2))
```

3. What is the average and median CO2 uptake of the plants from Quebec and Mississippi?

```
median_Quebec = median(CO2$uptake[which(CO2$Type=="Quebec")])

median_Mississippi = median(CO2$uptake[which(CO2$Type=="Mississippi")])
print(paste0("Quebec: " ,median_Quebec))
```

```
## [1] "Quebec: 37.15"
```

```
print(paste0("Mississippi: ",median_Mississippi))
```

```
## [1] "Mississippi: 19.3"
```

# Task 5

1. Write a function that calculates the ratio of the mean and the median of a given vector.

```
MeanToMedian = function(x){
    #calculate the mean and median
    meanX = mean(x)
    medianX = median(x)

    #calculate the mean to median ratio if median is not zero; otherwise return a warning that median is zero and return the mean value
    if (medianX!=0){
        ratioX = meanX/medianX
        return(ratioX)} else{
        print("Warning: Can't divide because median is 0.")
        print(paste0("Mean is: ",meanX,"."))
           }
}
```

2. Write a function that ignores the lowest and the highest value from a given vector and calculate the mean.

```
TrimMean = function(x){
    x = sort(x)
    x = x[-1]
    x = x[-length(x)]
    meanx = mean(x)
    return(meanx)
    }
```

3. Read about piping from here:https://r4ds.had.co.nz/pipes.html#pipes (https://r4ds.had.co.nz/pipes.html#pipes) (you don't have to learn everything, a basic understanding of the usage is enough). Write a short (max. 300 characters, no spaces) explanation of why, how, and when not to use pipes. Answer: (1). If the process involves more than one objects, one should not use pipes; (2). I think pipe is not explicit for debugging so when one is explorative for his code I'd suggest to not use pipe;

4. Familiarize yourself with the apply-family of functions (apply, lapply, sapply etc.) http://uc-r.github.io/apply_family (http://uc-r.github.io/apply_family) Write a short explanation (max. 300 characters, no spaces) of why they could be useful in your work.

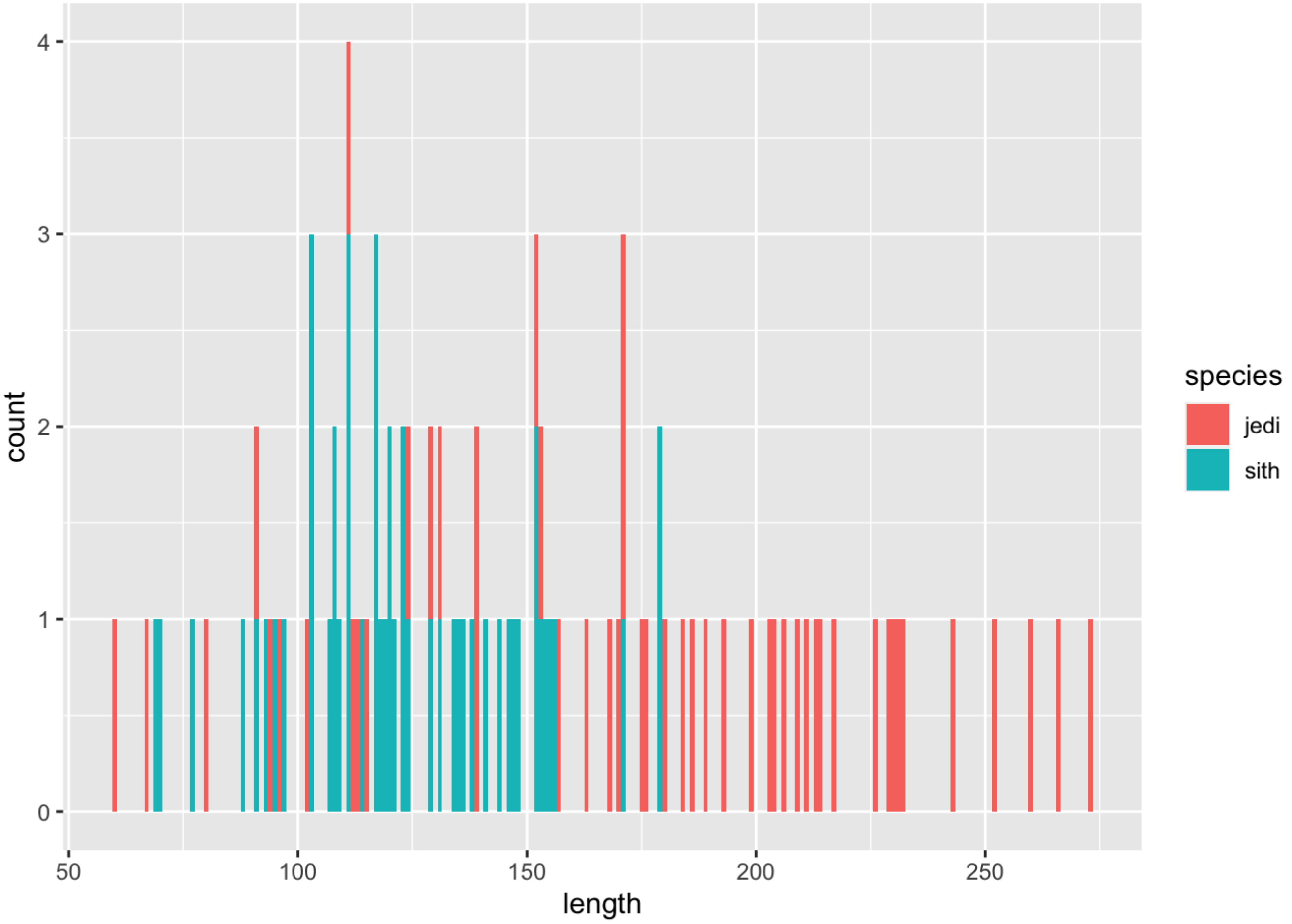
Answer: (1) Using the apply family help avoids loops and therefore keeps the code tidy; (3) They are marginally faster than a regular for loop; (2) Some apply functions have mcapply version so multi-processing can be utilized.

# Task 6

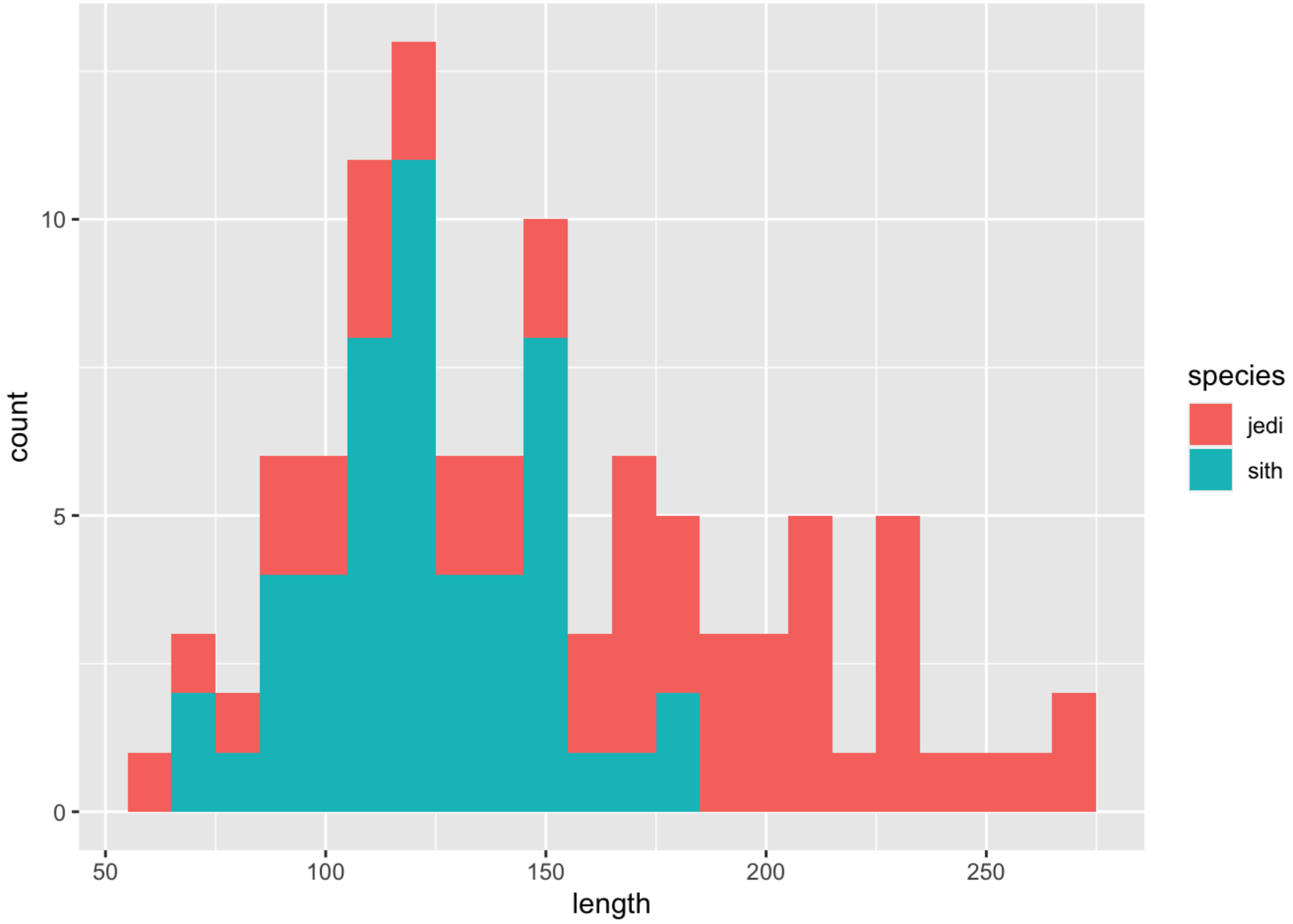
1. Compare the distributions of the body heights of the two species from the 'magic_guys.csv' dataset graphically

a. using the basic 'hist' function as well as 'ggplot' and 'geom_histogram' functions from the ggplot2 package. Optimize the plots for example by trying several different 'breaks'. Note that ggplot2-based functions give you many more options for changing the visualization parameters, try some of them.

```
magic_guys =  read.csv("/Users/yyh/Downloads/magic_guys.csv")
library(ggplot2)
ggplot(magic_guys)+ geom_histogram(aes(x=length,fill=species),binwidth=1)
```

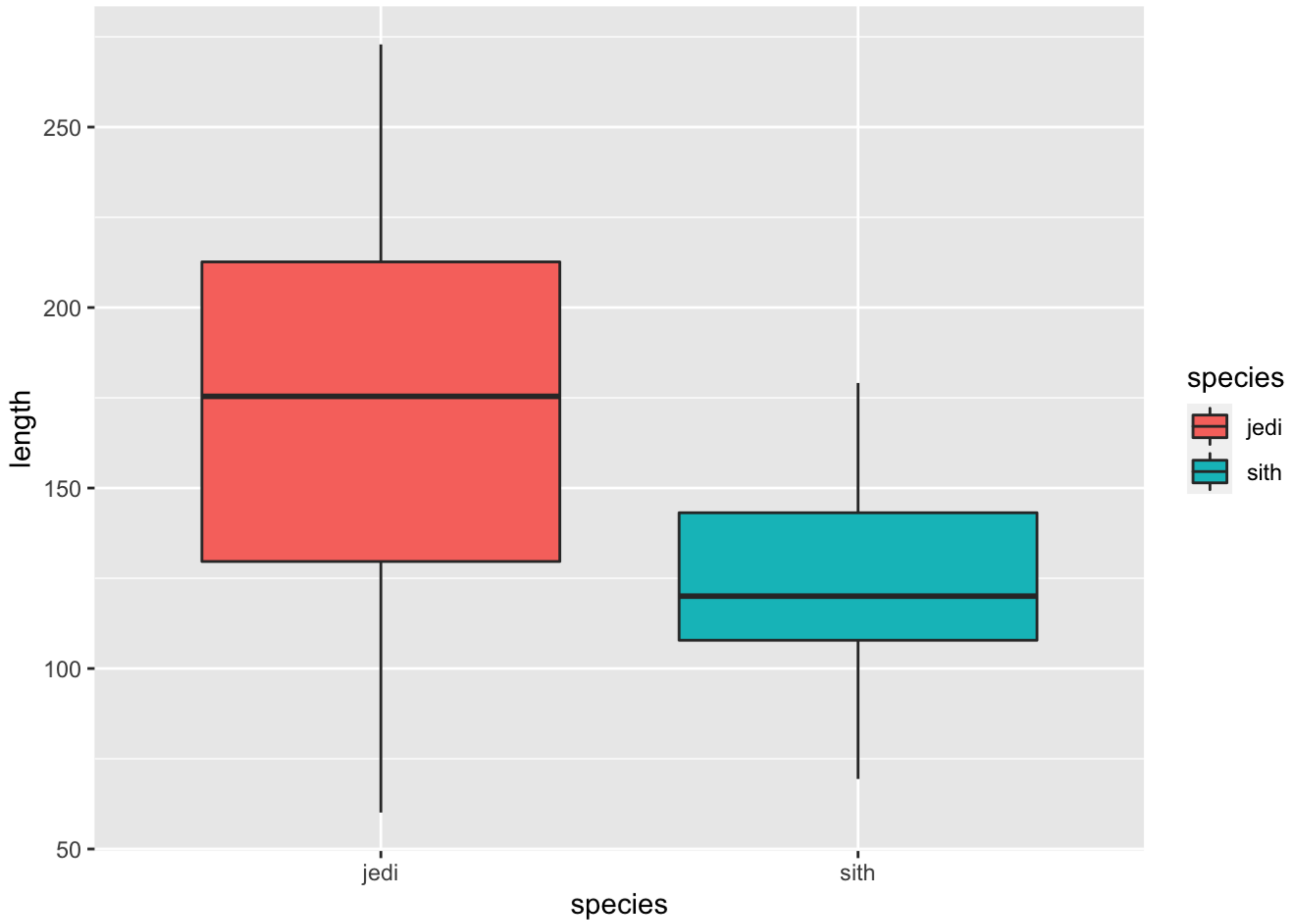


```
ggplot(magic_guys)+ geom_histogram(aes(x=length,fill=species),binwidth=10)
```



b. Do the same comparison as in a. but with boxplots. If you want to use the ggplot2-package, use the functions ‘ggplot’ and ‘geom_boxplot’.

```
ggplot(magic_guys)+ geom_boxplot(aes(x=species,y=length,fill=species))
```



c. Save the plots with the ‘png’, ‘pdf’, and ‘svg’ formats. In which situation would you use which file format?

```
pdf("/Users/yyh/Downloads/Length_by_species.pdf",width = 6,height = 6)
ggplot(magic_guys)+ geom_boxplot(aes(x=species,y=length,fill=species))
dev.off()
```

```
## quartz_off_screen
##                 2
```

```
png("/Users/yyh/Downloads/Length_by_species.png",width = 500,height = 500)
ggplot(magic_guys)+ geom_boxplot(aes(x=species,y=length,fill=species))
dev.off()
```

```
## quartz_off_screen
##                 2
```

```
svg("/Users/yyh/Downloads/Length_by_species.svg",width = 6,height = 6)
ggplot(magic_guys)+ geom_boxplot(aes(x=species,y=length,fill=species))
dev.off()
```

```
## quartz_off_screen
##                 2
```

When there are too muany elements I save as png; otherwise I save as pdf since it is Vector graphics.

2. Load the gene expression data matrix from the ‘microarray_data.tab’ dataset provided in the shared folder, it is a big tabular separated matrix.

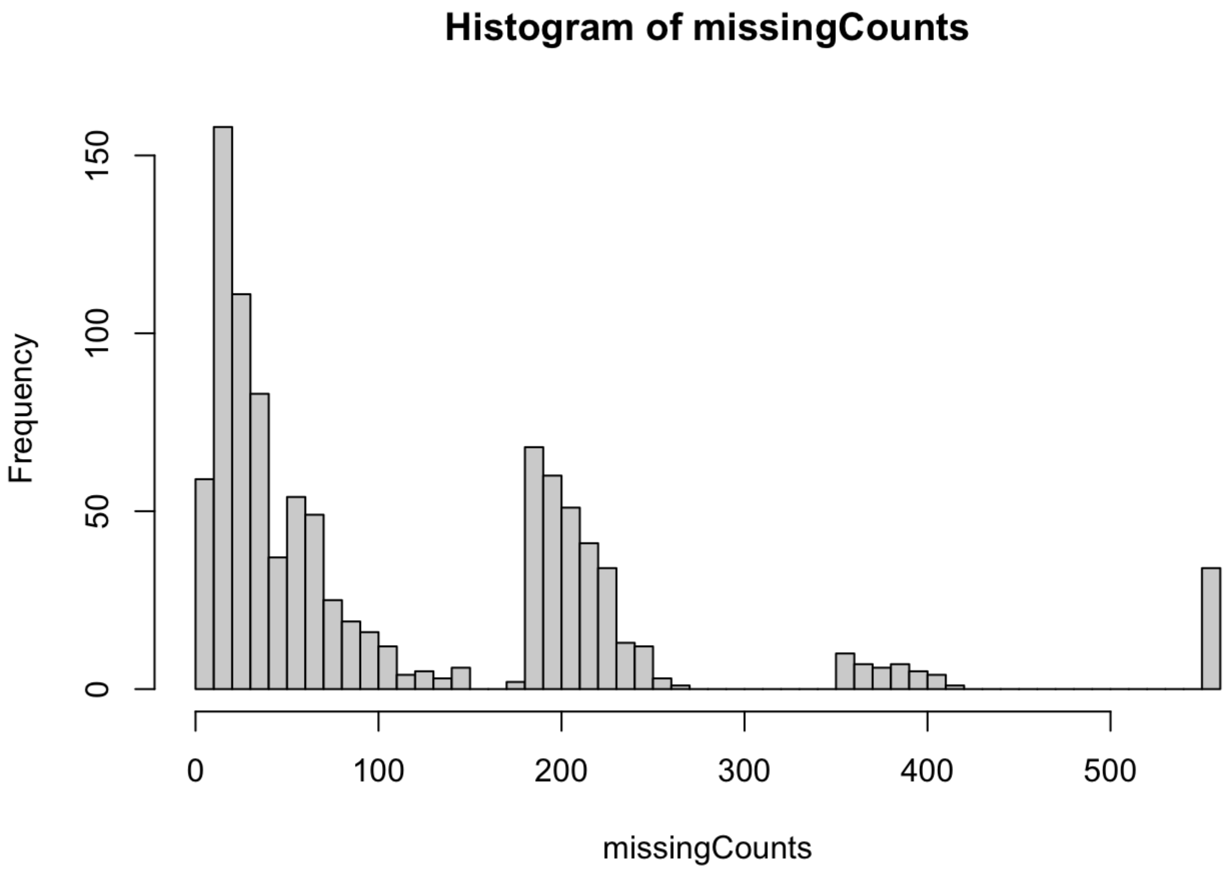
a. How big is the matrix in terms of rows and columns?

```
microarray_data =  read.table("/Users/yyh/Downloads/microarray_data.tab",sep="\t",header=T)
dim(microarray_data )
```

```
## [1]  553 1000
```

b. Count the missing values per gene and visualize this result.

```
missingCounts = colSums(is.na(microarray_data))
hist(missingCounts,n=50)
```



c. Find the genes for which there are more than X% (X=10%, 20%, 50%) missing values.

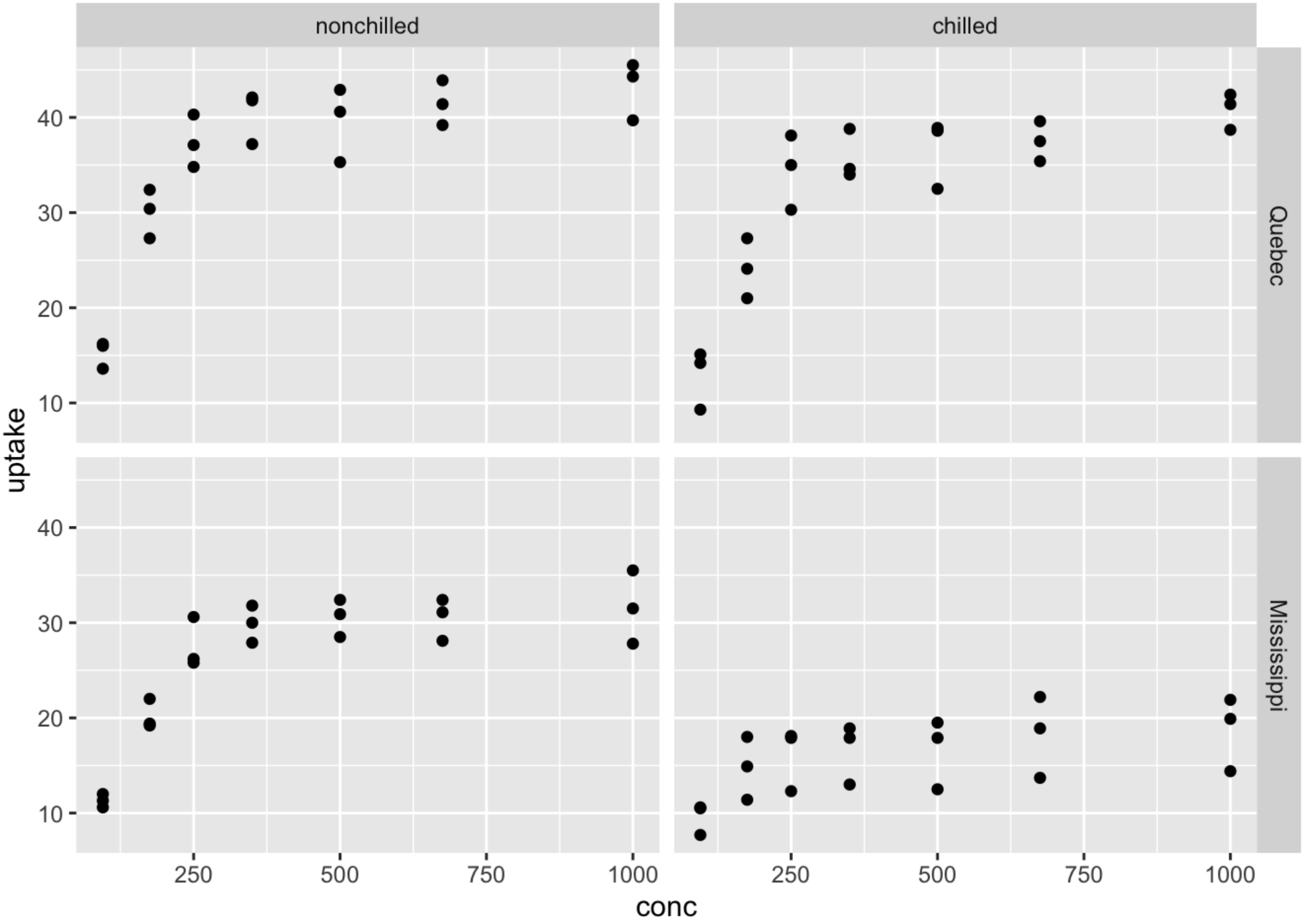
```
missingByPercent = missingCounts/dim(microarray_data)[1]
Cut_0.1 = names(which(missingByPercent > 0.1))
Cut_0.2 = names(which(missingByPercent > 0.2))
Cut_0.5 = names(which(missingByPercent > 0.5))
```

d. Replace the missing values by the average expression value for the particular gene. (Note: Imputing data has to be used with caution!)

```
ArrayMeans = colMeans(microarray_data,na.rm = T )
ReplaceNA = function(m){
 m[is.na(m)] = mean(m,na.rm=T)
 return(m)
}
microarray_data_new = apply(microarray_data, MARGIN = 2, FUN = ReplaceNA)
```

3. Visualize the data in the CO2 dataset in a way that gives you a deeper understanding of the data. What do you see?

```
ggplot(CO2)+geom_point(aes(conc,uptake))+
    facet_grid(vars(Type),vars(Treatment))
```



# Task 7

a. Extract summary statistics (mean, median and maximum) for the following variables from the ‘chromosome’ data: variations, protein coding genes, and miRNAs. Utilize the tidyverse functions to make this as simply as possible.

```
library(tidybiology)
data("chromosome")
chromosome = as.data.frame(chromosome)
M3 = function(feature){
    x =chromosome[,feature]
    print(paste0("mean of ",feature," is: ",mean(x)))
     print(paste0("median of ",feature," is: ",median(x)))
      print(paste0("maximum of ",feature," is: ",max(x)))
}
M3("variations")
```

```
## [1] "mean of variations is: 6484571.5"
## [1] "median of variations is: 6172346"
## [1] "maximum of variations is: 12945965"
```

```
M3("protein_codinggenes")
```

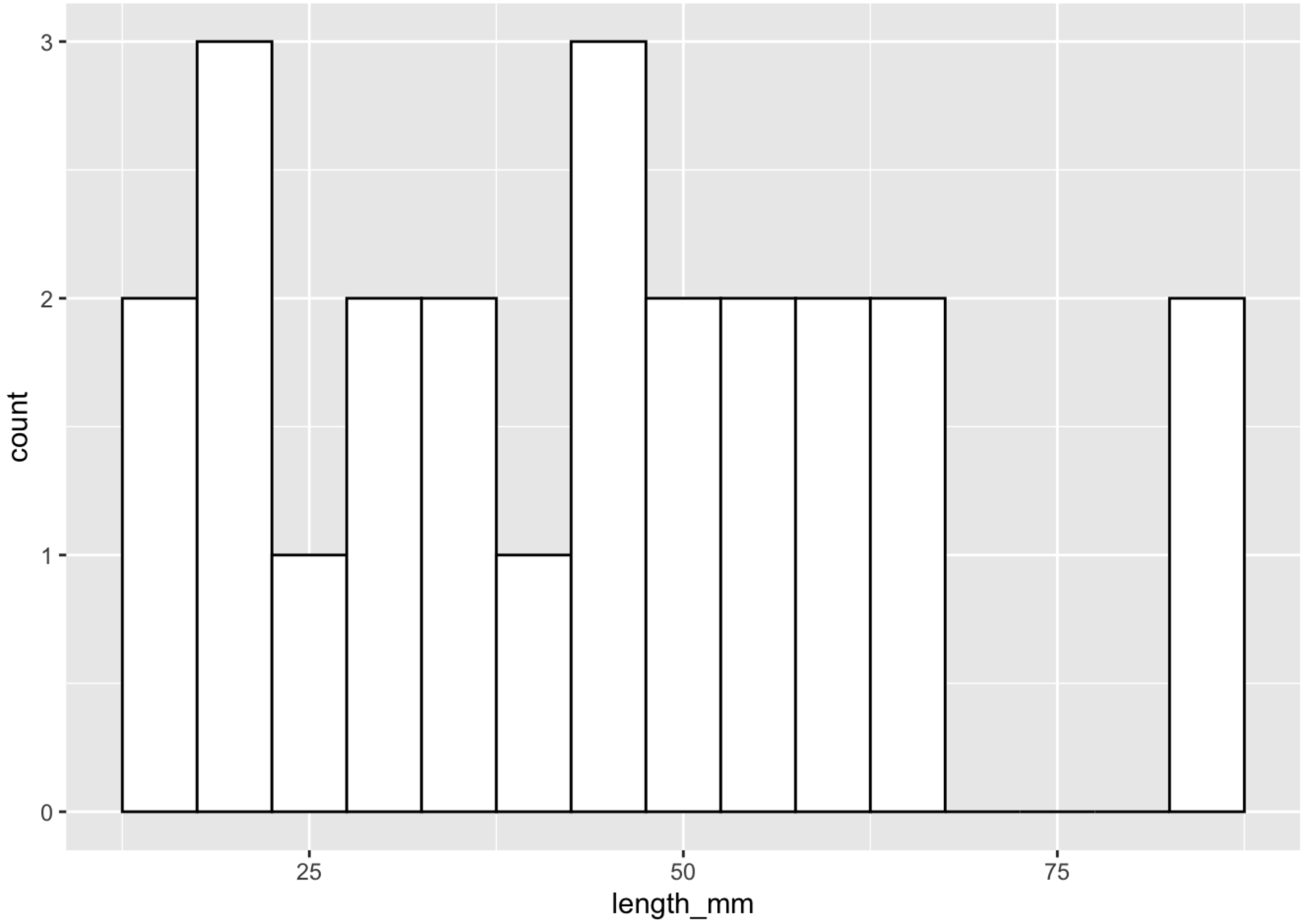
```
## [1] "mean of protein_codinggenes is: 849.958333333333"
## [1] "median of protein_codinggenes is: 836"
## [1] "maximum of protein_codinggenes is: 2058"
```

```
M3("mi_rna")
```

```
## [1] "mean of mi_rna is: 73.1666666666667"
## [1] "median of mi_rna is: 75"
## [1] "maximum of mi_rna is: 134"
```

b. How does the chromosome size distribute? Plot a graph that helps to visualize this by using ggplot2 package functions.

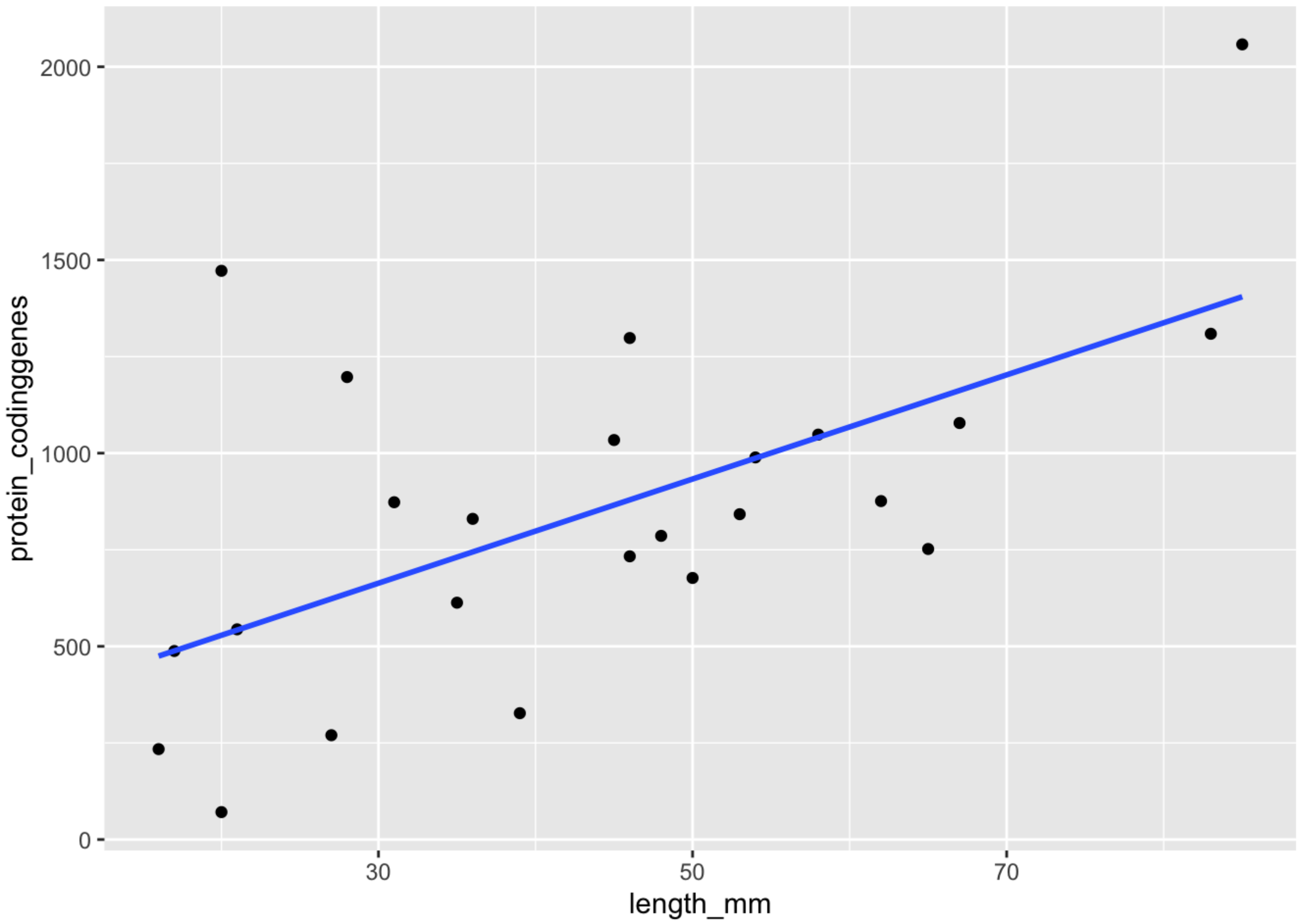
```
library(ggplot2)
p=ggplot(chromosome)+ geom_histogram(aes(x=length_mm),binwidth=5, colour="black", fil
l="white")
p
```



c. Does the number of protein coding genes or miRNAs correlate with the length of the chromosome? Make two separate plots to visualize these relationships.

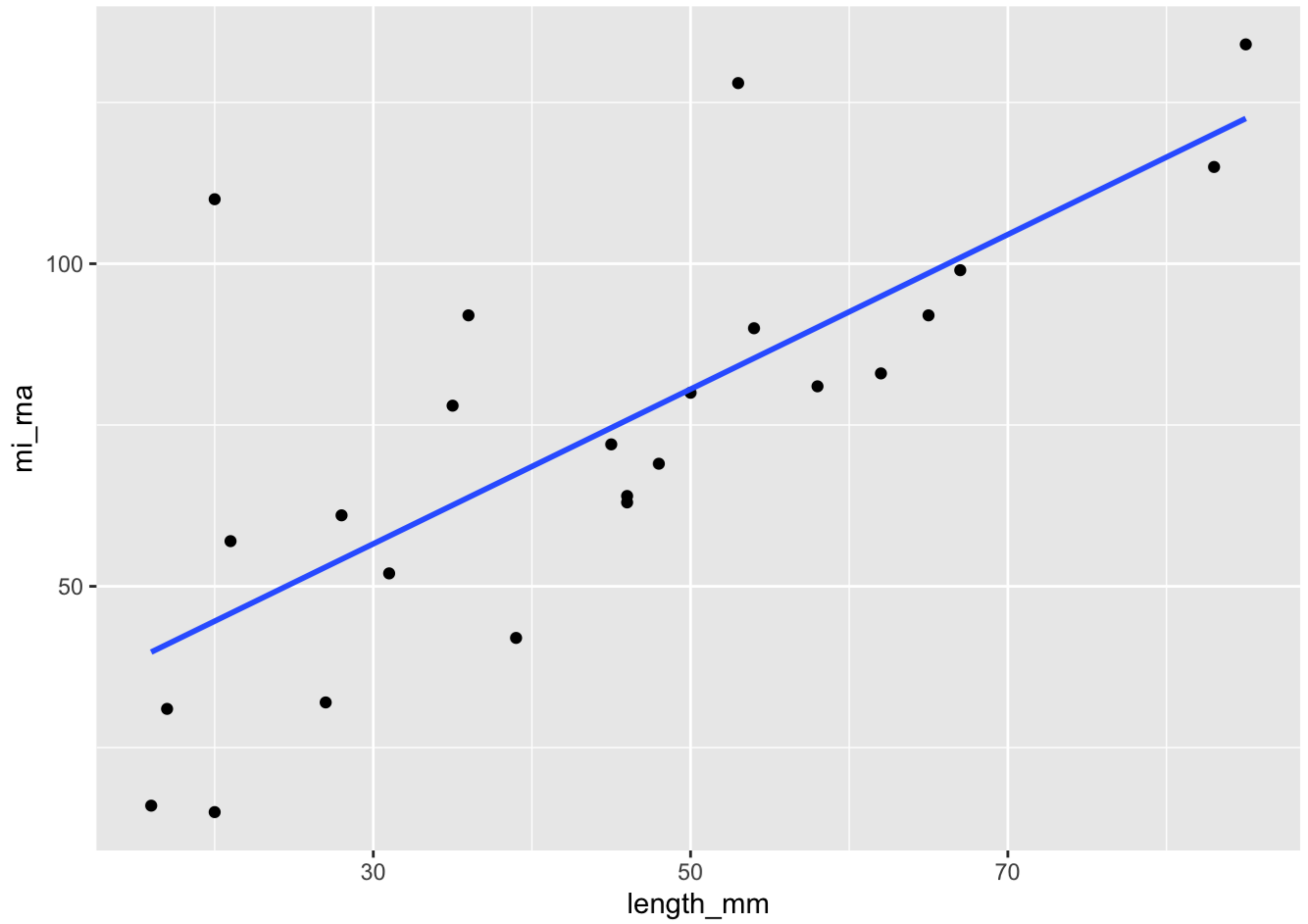
```
p1=ggplot(chromosome,aes(x=length_mm,y=protein_codinggenes))+ geom_point()+geom_smoot
h(method = "lm", se = FALSE)
p2=ggplot(chromosome,aes(x=length_mm,y=mi_rna))+ geom_point()+geom_smooth(method = "l
m", se = FALSE)
p1
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
p2
```

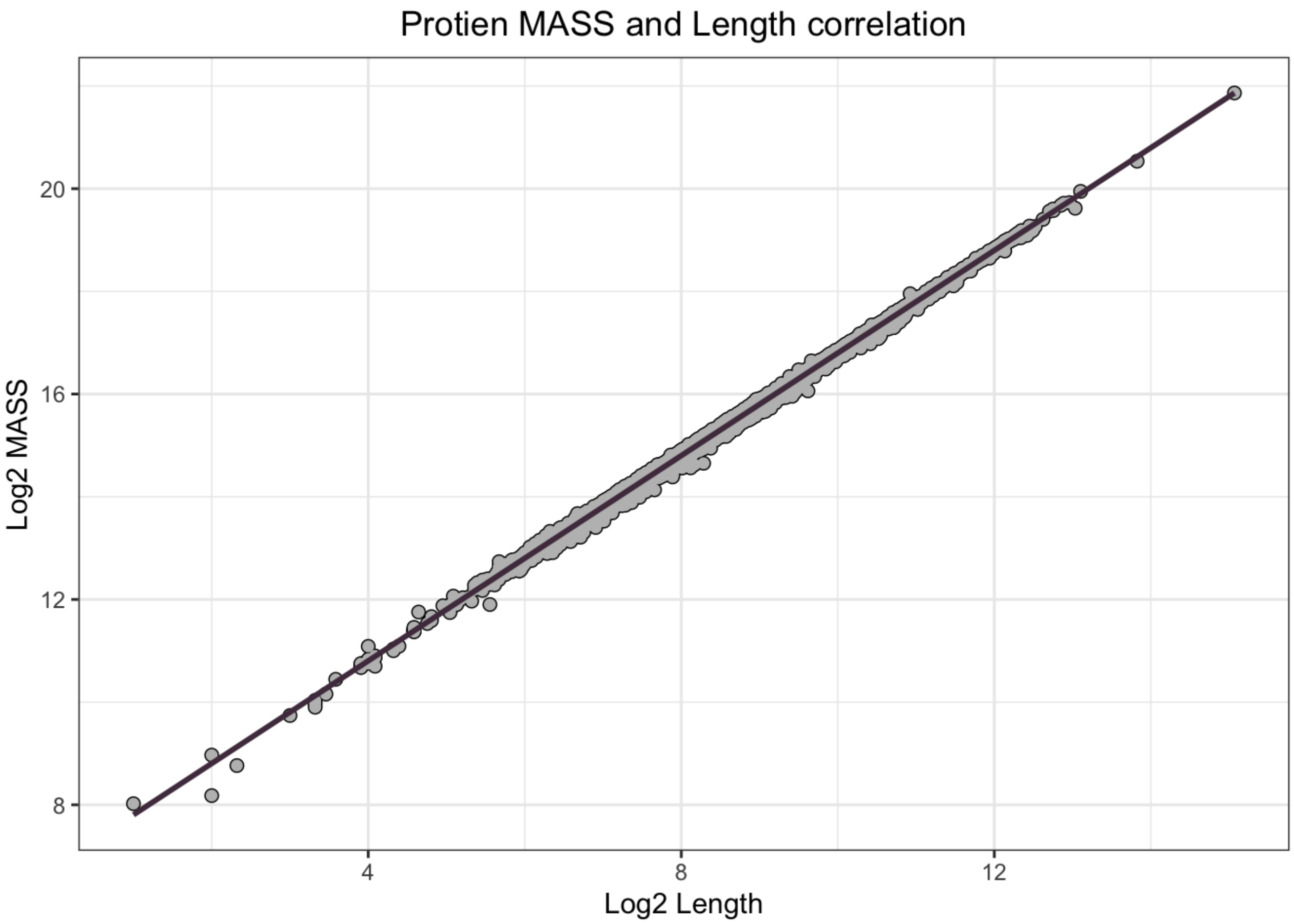
```
## `geom_smooth()` using formula 'y ~ x'
```



d. Calculate the same summary statistics for the 'proteins' data variables length and mass. Create a meaningful visualization of the relationship between these two variables by utilizing the ggplot2 package functions. Play with the colors, theme- and other visualization parameters to create a plot that pleases you.

```
data("proteins")
ggplot(proteins)+ geom_point(aes(x=log2(length),y=log2(mass)),color='black', size=2)+
geom_point(aes(x=log2(length),y=log2(mass)),color='grey',size=1.5)+geom_smooth(aes(x=
log2(length),y=log2(mass)),method = "lm", se = FALSE,col="#50394c",)+ggtitle("Protien
MASS and Length correlation")+xlab(label="Log2 Length")+ylab(label="Log2 MASS")+theme
_bw()+theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```


Protien MASS and Length correlation
Log2 MASS
20
16
12
8
4
8
12
Log2 Length