Assignment 1

Christina

2023-03-05

install.packages("r package", repos = "http://cran.us.r-project.org")

## Installing package into '/Library/Frameworks/R.framework/Versions/4.2/Resources/library'  
## (as 'lib' is unspecified)

## Warning: package 'r package' is not available for this version of R  
##   
## A version of this package for your version of R might be available elsewhere,  
## see the ideas at  
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages

#task 1  
  
#a. What is the medically relevant insight from the article?  
  
#In this study, the authors developed a novel metric, the exclusively expressed index (EEI), to comprehensively quantify mutual exclusivity between two genes from sparse scRNA-seq data, in this case a glioblastoma scRNAseq dataset. Glioblastoma is a notoriously heterogeneous tumor, so methods to improve the sensitivity of the identification of cell-to-cell heterogeneity are very relevant in this context. They evaluated the effectiveness of EEI compared to four existing methods, the Pearson correlation coefficient, minet, GENIE3 and PIDC, and found that EEI was better at identifying mutually exclusive gene sets.  
  
#b. Which genomics technology/ technologies were used?  
  
#To evaluate the performance of EEI and the community detection of coexpression networks, they used six scRNA-seq datasets.  
#1. Glioblastoma scRNA-seq data from stem-like cells, before and after the addition of serum that are collected at 0 and 12 hours  
#2. Human ES progenitor scRNA-seq data by Chu et al.  
#3. Human ESC-derived neuron scRNA-seq data by Manno et al.  
#4 Mouse cortex scRNA-seq data  
#5 PBMC CELseq2 scRNA-seq dat by Mereu et al.   
#6 PBMC MARSseq scRNA-seq data by Mereu et al.  
  
#The authors evaluated the performance of EEI on the basis of the area under the precision-recall curve (AUPR) and average precision. To validate the performance for clustering of single cells, they adopted the Adjusted Rand Index (ARI) and the silhouette coefficient. ARI measures the similarity between predicted and true cluster labels and ranges from 0.0 to 1.0.  
  
#3.  
#a. List and explain at least three questions/ hypotheses you can think of that extend the analysis presented in the paper.  
  
# While the authors are comparing 5 methods to each other, they use a relative prediction accuracy score, setting the value of the EEI prediction accuracy as 1 and then comparing the 4 existing methods to EEI (Figure 2). It is still left unknown whether the EEI method is accurate at predicting real commonalities between datasets that have any biological significance.  
  
# When the authors performed a comparison of community detection, they used known marker genes for specific cell types in order to identify other candidate marker genes from known marker genes, for example they specify NSPCs as cells which co-express SOX2, PAX6, MAP2. In glioblastoma CSCs, there are several subtypes of pro-neural cells or pre-oligondentrocyte-like cells which all fall under the same umbrella of stem-like cells and the known markers for each are sometimes overlapping, so it is a slippery slope to infer candidate genes from markers which are not so well-established yet. Also the expression levels of these markers (for example SOX2high or SOX2low) could have been included to further classify the subsets of genes.  
  
# The authors used a newly generated glioblastoma scRNA-seq dataset from stem-like cells, before and after the addition of serum that are collected at 0 and 12 hours. The datasets obtained at 0h and 12h contained 2102 and 2209 single cells in total, respectively. The cells were grown in stem-cell enriching media, so a certain cell selection happened, enriching for NSPCs (since they used F12 media that the pro-neural cells prefer), so their 2 datasets will be more similar to each other than not similar, clouding the data. I also think that 12 hours might be too short of a timeframe for the cells to expand, which is reflected also in the minor difference between 2102 and 2209 cells in the 0h and 12h timepoint. It would have been more relevant to include tumor samples from different patients (which would have not only intra-tumor, but also patient-to-patient heterogeneity) and compare those to each other.

#task 4  
  
#1  
sqrt(10)

## [1] 3.162278

#2  
log2(32)

## [1] 5

#3  
sum(1:1000)

## [1] 500500

#4  
sum(seq(2,1000, by=2))

## [1] 250500

#5 the mathematical formula is n(n-1)/2 with n=100 for 100 genes  
100\*99/2

## [1] 4950

#6  
genes\_in\_triplets<-combn(100,3)  
sum(genes\_in\_triplets)

## [1] 24497550

#task 5  
  
#1  
CO2  
  
#2  
?CO2  
#Response: The CO2 data frame has 84 rows and 5 columns of data from an experiment on the cold tolerance of the grass species Echinochloa crus-galli.The CO2 uptake of six plants from Quebec and six plants from Mississippi was measured at several levels of ambient CO2 concentration. Half the plants of each type were chilled overnight before the experiment was conducted.  
  
#3  
Quebec<-subset(CO2, Type=="Quebec")  
ave(Quebec[,5])  
median(Quebec[,5])  
Mississippi<-subset(CO2, Type=="Mississippi")  
ave(Mississippi[,5])  
median(Mississippi[,5])

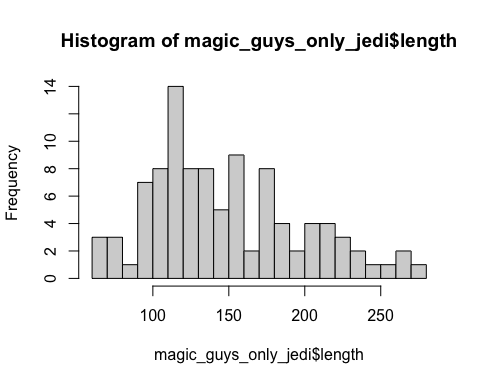
#task 6  
  
#1  
given\_vector <- c(1, 3, 3, 4, 7, 5)   
  
mean(given\_vector)/median(given\_vector)  
  
#2  
  
#ignore min and max from vector  
  
max(given\_vector)  
min(given\_vector)  
  
y <- given\_vector[! given\_vector %in% c(min(given\_vector), max(given\_vector))]  
  
y #just to check if it actually removed the values  
  
mean(y)  
  
#3 Pipes are used to chain functions one after the other, but should not be used:   
# When the data becomes too complex, with multiple data frames,  
# when you need to access intermediate results of your code,  
# when you want to use functions that require multiple arguments,  
# when you need to modify the data in-place (because pipes create new data frames at each step).  
  
#4   
  
#the apply family functions allow for the application of a function to a specified row or column of a data frame, hence could be useful in data analysis. For example, "apply" can be used to calculate the mean expression level of each gene across multiple samples in a gene expression matrix, or the correlation coefficient between two variables, such as gene expression and drug treatment, across multiple samples.

#task 7  
  
setwd("/Users/chrneo/Library/CloudStorage/OneDrive-KI.SE/Mac/Desktop/Japan course")  
  
library(tidyverse)

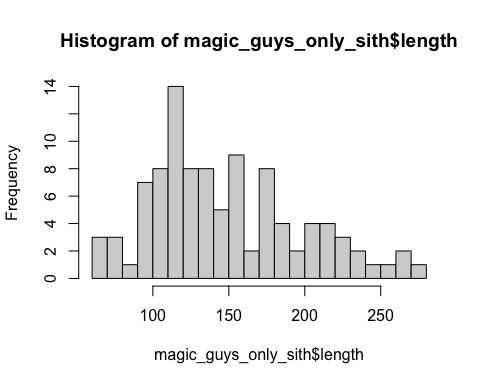
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.0 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.1.8  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library(ggplot2)  
  
read.csv(file = "/Users/chrneo/Downloads/magic\_guys.csv")  
  
magic\_guys <- read.csv(file = "/Users/chrneo/Downloads/magic\_guys.csv")  
magic\_guys

# task 7. 1   
  
#a,b,c.   
  
magic\_guys\_only\_jedi <- magic\_guys[magic\_guys$length != "sith", ]  
magic\_guys\_only\_sith <- magic\_guys[magic\_guys$length != "jedi", ]  
  
png(file = "histogram.png", width=1600, height=1167) #saves as png, best for figures  
par(mfrow=c(1,2)) #makes two graphs next to each other  
hist(magic\_guys\_only\_jedi$length, breaks=10)   
hist(magic\_guys\_only\_sith$length, breaks=10)  
dev.off() #closes the png  
  
hist(magic\_guys\_only\_jedi$length, breaks=20)

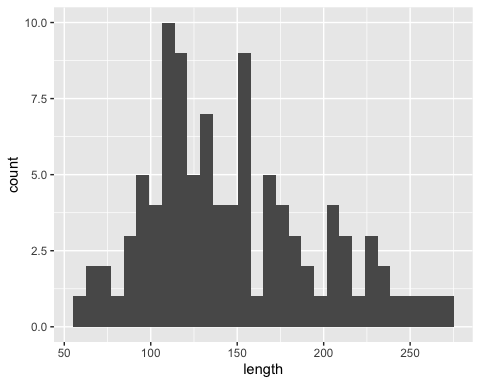


hist(magic\_guys\_only\_sith$length, breaks=20)

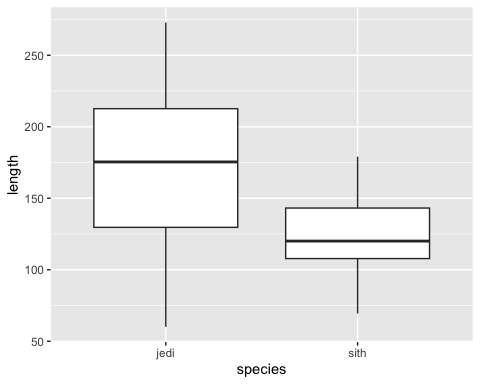


pdf(file = "histogram.pdf") #saves as pdf, for documents  
par(mfrow=c(1,2)) #makes two graphs next to each other  
hist(magic\_guys\_only\_jedi$length, breaks=10)   
hist(magic\_guys\_only\_sith$length, breaks=10)  
dev.off() #closes the pdf  
  
svg(file = "histogram.svg") #saves as svg, useful for illustrator, you an change the labels etc  
par(mfrow=c(1,2)) #makes two graphs next to each other  
hist(magic\_guys\_only\_jedi$length, breaks=10)   
hist(magic\_guys\_only\_sith$length, breaks=10)  
dev.off() #closes the svg  
  
ggplot(magic\_guys, aes(x=length)) + geom\_histogram() #histogram but with ggplot

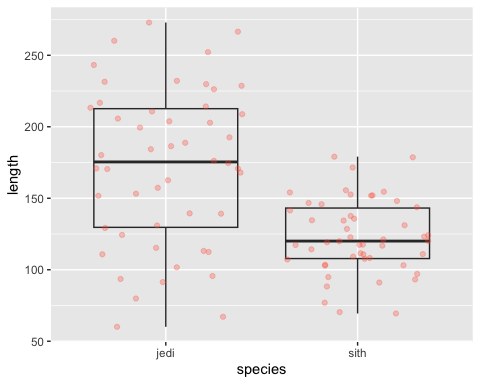
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



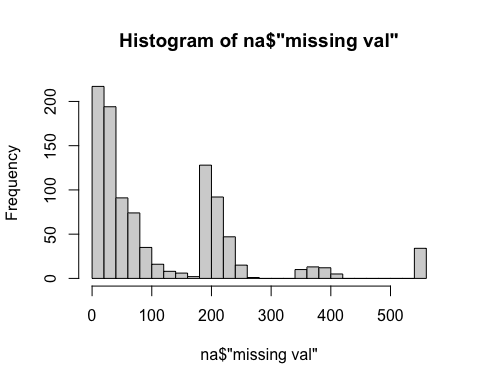
#the height of both species is normally distributed   
  
#boxplots  
png(file = "boxplot.png", width=1600, height=1167) #makes a png file  
par(mfrow=c(1,2))  
boxplot(magic\_guys\_only\_jedi$length, horizontal=TRUE, main="Jedi height")  
boxplot(magic\_guys\_only\_sith$length, horizontal=TRUE, main="Sith height")  
dev.off() #closes the png  
  
ggplot(data = magic\_guys, mapping = aes(x = species, y = length)) +  
 geom\_boxplot() #boxplot but with ggplot



ggplot(data = magic\_guys, mapping = aes(x = species, y = length)) +  
 geom\_boxplot(alpha = 0) +  
 geom\_jitter(alpha = 0.3, color = "tomato") #example of different visualisation

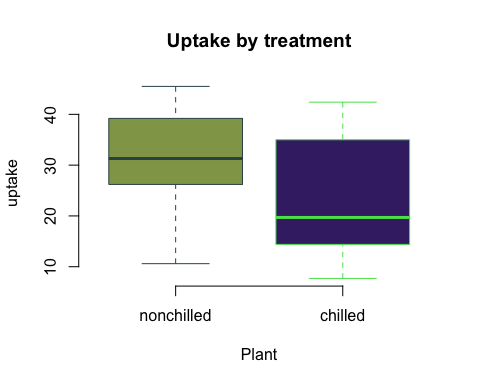


#a  
ncol(microarray) #it has 1000 columns  
nrow(microarray) #it has 553 rows  
  
#b  
  
sum(is.na (microarray)) #there is a total of 117826 missing values in the matrix  
sapply(microarray, function(x) sum(length(which(is.na(x))))) #finds n/a per column  
na <- sapply(microarray, function(x) sum(length(which(is.na(x)))))  
na<-as.matrix(na)   
na <- cbind(rownames(na), data.frame(na, row.names=NULL))  
colnames(na) <- c("gene", "missing val")  
na  
  
hist(na$"missing val", breaks=30) #visualization

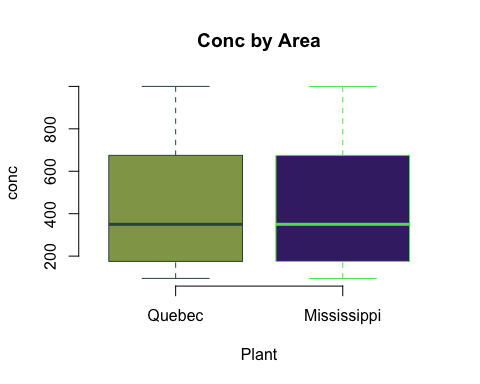


#2c  
percent\_missing\_ten <- colMeans(is.na(na))  
genes\_with\_more\_than\_10\_percent\_missing <- names(percent\_missing\_ten[percent\_missing\_ten > 0.1])  
genes\_with\_more\_than\_10\_percent\_missing  
  
percent\_missing\_twenty <- colMeans(is.na(na))  
genes\_with\_more\_than\_10\_percent\_missing <- names(percent\_missing\_twenty[percent\_missing\_twenty > 0.2])  
genes\_with\_more\_than\_10\_percent\_missing  
  
percent\_missing\_fifty <- colMeans(is.na(na))  
genes\_with\_more\_than\_10\_percent\_missing <- names(percent\_missing\_fifty[percent\_missing\_fifty > 0.5])  
genes\_with\_more\_than\_10\_percent\_missing  
  
#2d  
  
row\_means <- rowMeans(microarray, na.rm=TRUE)  
row\_means  
microarray\_data\_replaced<-apply(na, 2, function(x) {  
 x[is.na(x)] <- row\_means  
 x  
})  
microarray\_data\_replaced

#task 7.3  
  
CO2  
  
boxplot(uptake ~ Treatment, data=CO2, breaks=30, frame=F, border=c("#345355", "#54E256"), col=c("#92A357", "#422974"),xlab="Plant", main="Uptake by treatment")#We see that nonchilled plants have more uptake of CO2 and that the chilled ones have bigger variation



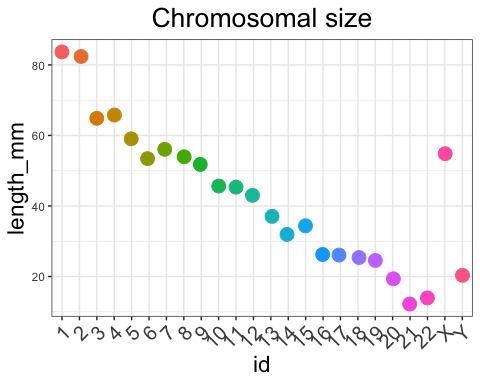
boxplot(conc ~ Type, data=CO2, breaks=30, frame=F, border=c("#345355", "#54E256"), col=c("#92A357", "#422974"),xlab="Plant", main="Conc by Area")#The Conc is not dependent by the area



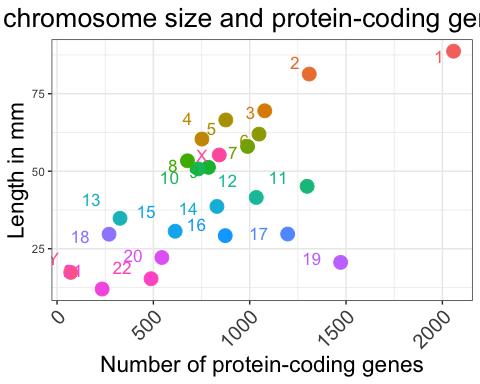
#Task 8  
  
#8a  
  
devtools::install\_github("hirscheylab/tidybiology")

## Skipping install of 'tidybiology' from a github remote, the SHA1 (d03a810a) has not changed since last install.  
## Use `force = TRUE` to force installation

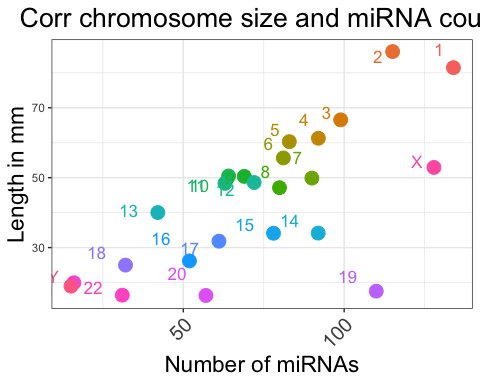
library(tidybiology)  
#a  
library(tidyverse)  
data(chromosome)  
chromosome  
Statistics<-summarise(chromosome,across(c(variations, protein\_codinggenes, mi\_rna),list(mean, median, max)))  
Statistics <- as.data.frame(matrix(unlist(Statistics), ncol = 3, byrow = TRUE))  
colnames(Statistics)<-c("Mean", "Median","Max")  
rownames(Statistics)<-c("Variations", "Protein-coding genes","miRNA")  
Statistics  
  
#8b  
library(ggplot2)  
?ggplot  
Chromosomal\_size<-ggplot(chromosome)+ aes(x =id , y = length\_mm, color = id, size=1) +   
 geom\_point(position=position\_jitter(w = 0.1,h = 5)) +  
 theme\_bw() +  
 ggtitle("Chromosomal size") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1, size=15)) +  
 theme(axis.title.x = element\_text(size=17))+  
 theme(axis.title.y = element\_text(size=17))+  
 theme(plot.title = element\_text(hjust = 0.5, size=20))+  
 theme(legend.position = "none")  
Chromosomal\_size



#8c  
Chromosomal\_num\_pcg<-ggplot(chromosome)+ aes(x =protein\_codinggenes , y = length\_mm, color = id, size=1) +   
 geom\_point(position=position\_jitter(w = 0.1,h = 5)) +  
 theme\_bw() +  
 geom\_text(aes(label = id), hjust = 2, vjust = 0)+  
 ggtitle("Corr chromosome size and protein-coding gene count") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1, size=15)) +  
 theme(axis.title.x = element\_text(size=17))+  
 theme(axis.title.y = element\_text(size=17))+  
 theme(plot.title = element\_text(hjust = 0.5, size=20))+  
 theme(legend.position = "none") +  
 labs(x = "Number of protein-coding genes", y = "Length in mm")  
Chromosomal\_num\_pcg



Chromosomal\_num\_miRNA<-ggplot(chromosome)+ aes(x =mi\_rna , y = length\_mm, color = id, size=1) +   
 geom\_point(position=position\_jitter(w = 0.1,h = 5)) +  
 theme\_bw() +  
 geom\_text(aes(label = id), hjust = 2, vjust = 0)+  
 ggtitle("Corr chromosome size and miRNA count") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1, size=15)) +  
 theme(axis.title.x = element\_text(size=17))+  
 theme(axis.title.y = element\_text(size=17))+  
 theme(plot.title = element\_text(hjust = 0.5, size=20))+  
 theme(legend.position = "none") +  
 labs(x = "Number of miRNAs", y = "Length in mm")  
Chromosomal\_num\_miRNA



#8d  
data(proteins)  
proteins  
Protein\_mass\_length<-ggplot(proteins)+ aes(x =mass , y = length, color = uniprot\_id, size=1) +   
 geom\_point(position=position\_jitter(w = 0.1,h = 5)) +  
 theme\_bw() +  
 geom\_text(aes(label = protein\_name), hjust = 1.4, vjust = 0)+  
 ggtitle("Corr chromosome size and miRNA count") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1, size=15)) +  
 theme(axis.title.x = element\_text(size=17))+  
 theme(axis.title.y = element\_text(size=17))+  
 theme(plot.title = element\_text(hjust = 0.5, size=20))+  
 theme(legend.position = "none") +  
 labs(x = "Protein mass", y = "Length")  
Protein\_mass\_length

