Homework 1 - answers

Almog Angel

01/12/2024

In the following tasks you will use statistics and sequence alignment to analyze real COVID-19 world data.

- Load packages

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.4.4     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(Biostrings)
```

```
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
##
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
##
## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union
##
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
##
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min
##
## Loading required package: S4Vectors
## Loading required package: stats4
##
## Attaching package: 'S4Vectors'
##
## The following objects are masked from 'package:lubridate':
##
##     second, second<-
##
## The following objects are masked from 'package:dplyr':
##
##     first, rename
##
## The following object is masked from 'package:tidyr':
##
##     expand
##
## The following object is masked from 'package:utils':
##
##     findMatches
##
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
##
## Loading required package: IRanges
##
## Attaching package: 'IRanges'
##
## The following object is masked from 'package:lubridate':
##
##     %within%
##
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
##
## The following object is masked from 'package:purrr':
##
##     reduce
##
## The following object is masked from 'package:grDevices':
##
##     windows
##
## Loading required package: XVector
##
## Attaching package: 'XVector'
##
## The following object is masked from 'package:purrr':
##
##     compact
##
## Loading required package: GenomeInfoDb
##
## Attaching package: 'Biostrings'
##
## The following object is masked from 'package:base':
```

```
##
##     strsplit
```

```
library(msa)
library(msaR)
library(seqinr)
```

```
##
## Attaching package: 'seqinr'
##
## The following object is masked from 'package:Biostrings':
##
##     translate
##
## The following object is masked from 'package:dplyr':
##
##     count
```

```
library(ape)
```

```
##
## Attaching package: 'ape'
##
## The following objects are masked from 'package:seqinr':
##
##     as.alignment, consensus
##
## The following object is masked from 'package:Biostrings':
##
##     complement
##
## The following object is masked from 'package:dplyr':
##
##     where
```

- Set your working directory

```
# Enter your code here:
setwd("C:/technion/Bio/hw1/")
```

Task 1 - Data cleaning

- Read “COVID19_line_list_data.csv” into a variable called “covid_data”

```
# Enter your code here:
covid_data <- read_csv("COVID19_line_list_data.csv", show_col_types = FALSE)
```

```
## New names:
## • `` -> `...4`
## • `` -> `...22`
## • `` -> `...23`
## • `` -> `...24`
## • `` -> `...25`
## • `` -> `...26`
## • `` -> `...27`
```

- Take a look at the data

```
View(covid_data)
```

- Find a way to remove empty columns:

```
# Enter your code here:
covid_data <- covid_data[,apply(covid_data, 2, function(x) { sum(!is.na(x)) > 0 })]
```

- Find a way to deal with missing values (NAs) in “age” and “gender” columns

```
# Enter your code here:
covid_data <- covid_data %>%
  subset(gender != is.na(gender) ) %>%
  subset(age != is.na(age) )
```

- Assuming ‘death’ is a binary variable (0 for no, 1 for yes) that represent categorical data
- Convert ‘death’ to a factor class that contains either “yes” or “no”

```
# Enter your code here:
covid_data <- covid_data %>%
  mutate(death = ifelse(death == 0, "no", "yes"))
```

Task 2 - Descriptive statistics analysis

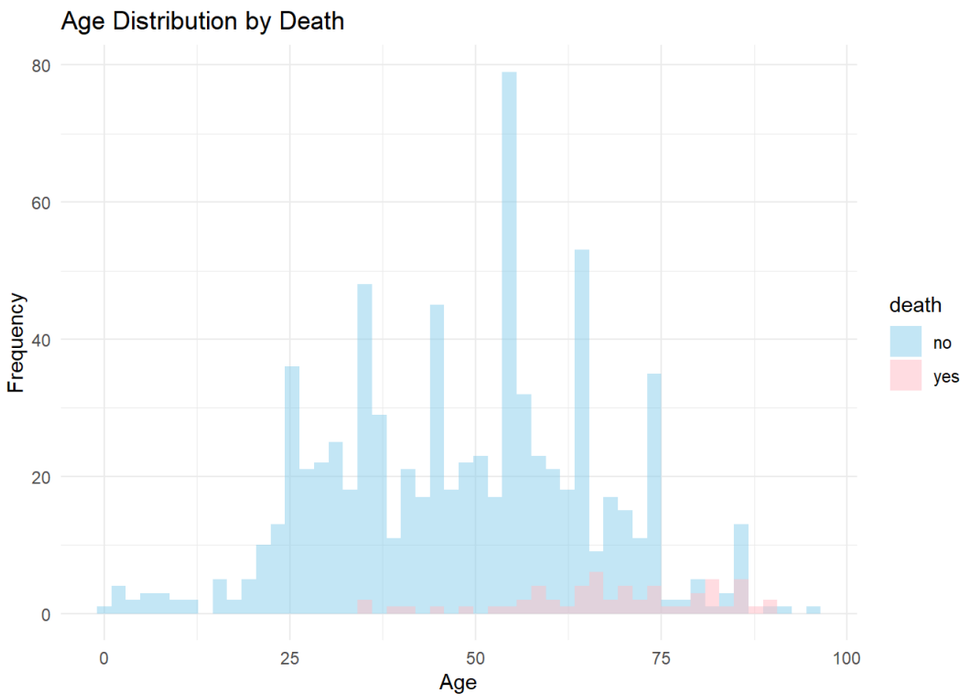
- Print a descriptive summary of “age”

```
# Enter your code here:
summary(covid_data$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.50   35.00   51.00   49.76   64.00   96.00
```

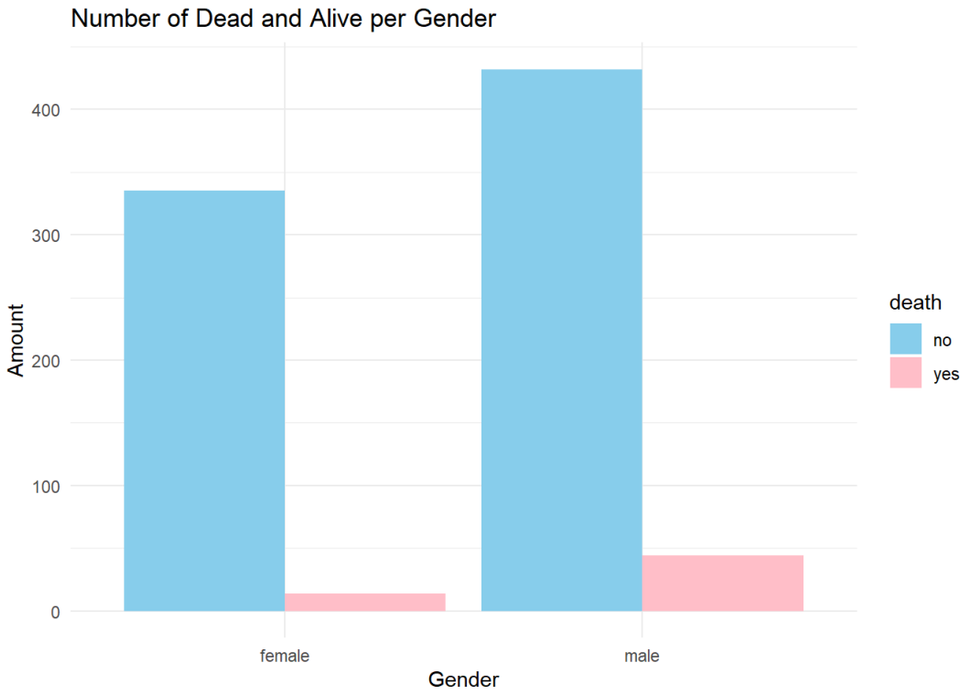
- Distribution:
- 1. Generate histogram for “age” distribution and color the bars by “death”
- Use: position = ‘identity’, bins = 50, alpha = 0.5

```
# Enter your code here:
ggplot(covid_data, aes(x = age, fill = death)) +
  geom_histogram(position = 'identity', bins = 50, alpha = 0.5) +
  labs(title = "Age Distribution by Death",
       x = "Age",
       y = "Frequency") +
  scale_fill_manual(values = c("yes" = "pink", "no" = "skyblue")) + # assigns colors to the "Alive" and "Dead" categories
  theme_minimal() # adjusts the plot's appearance to a minimal theme
```



- 2. Generate bar plot that visualize the number of dead and alive per gender

```
# Enter your code here:
ggplot(covid_data, aes(x = gender, fill = death)) +
  geom_bar(position = "dodge") +
  labs(title = "Number of Dead and Alive per Gender",
       x = "Gender",
       y = "Amount") +
  scale_fill_manual(values = c("no" = "skyblue", "yes" = "pink")) +
  theme_minimal()
```



Task 3 - Statistical Testing

- Run Chi-squared test to investigate the association between gender and death

```
# Enter your code here:
chisq.test(table(covid_data$gender, covid_data$death))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(covid_data$gender, covid_data$death)
## X-squared = 7.6526, df = 1, p-value = 0.005669
```

- Why do we use Chi-squared and not t-test?

```
Type your answer here:
We are using Chi-squred test because Chi-squared test checks relation between categorical varibales, where the outcome var
ibal is also categorical.
t-test tests categorical varibals the outcome varibals are quantitative (a t-test is a hypothesis test whether there is a
significant difference between the means of two numerical  groups).
In our case, we check gender VS death. In gender we have "male" or "female" (categorical). In death the parameters are  "y
es" or "no", also categorical. So we can not run t-test.
In the other hand, Chi test  used to assess the association or independence between two categorical variables, which is ex
actly what we are want to do.
```

- Define the Null Hypothesis of the Chi-squared Test in this example

```
Type your answer here:

There is no association between gender and death.
```

- What is the p-value? Is it significant (alpha = 0.05)? Write your conclusions from the statistical test results.

```
Type your answer here:
The P value is 0.005669 < 0.05.
This result indicates that we reject the null hypothesis. Therefore, we conclude that there is a difference between males
and females in terms of the likelihood of death during the disease.
```

According to the media, older people are more likely to die from COVID-19. - Test this claim statistically - Frame your null and alternative hypothesis here:

```
My null hypothesis:
There is no association between age and death.

My alternative hypothesis:
The chance of death among older people is higher.
```

- Check if the assumption of normality is valid for your data (use alpha of 0.05)

```
# Enter your code here:
shapiro.test(covid_data$age[covid_data$death == 'yes'])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  covid_data$age[covid_data$death == "yes"]
## W = 0.94869, p-value = 0.01585
```

```
shapiro.test(covid_data$age[covid_data$death == 'no'])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  covid_data$age[covid_data$death == "no"]
## W = 0.98965, p-value = 3.129e-05
```

- Run a statistical test:
- If you assume normality -> use t-test
- else -> use Wilcoxon rank-sum test

```
# Enter your code here:
wilcox.test(covid_data$age ~ covid_data$death)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  covid_data$age by covid_data$death
## W = 7923.5, p-value = 2.648e-16
## alternative hypothesis: true location shift is not equal to 0
```

- Report the p-value. Is it significant? Write your conclusions from the statistical test results.

```
Type your answer here:
The p-value is 2.648e-16 ( << 0.05 )
The p-value is much smaller 0.05, we reject the null hypothesis. Therefore, we conclude that there is a significant associ
ation between age and death.
```

- Read “owid-covid-data.csv” from “https://covid.ourworldindata.org/data/owid-covid-data.csv (https://covid.ourworldindata.org/data/owid-covid-data.csv)” into a variable called “covid_world”

```
covid_world <- read.csv('https://covid.ourworldindata.org/data/owid-covid-data.csv')
```

- Take a look at the data

```
View(covid_world)
```

Your friend suggest that countries with high Human Development Index (HDI) have a greater risk to die from COVID-19

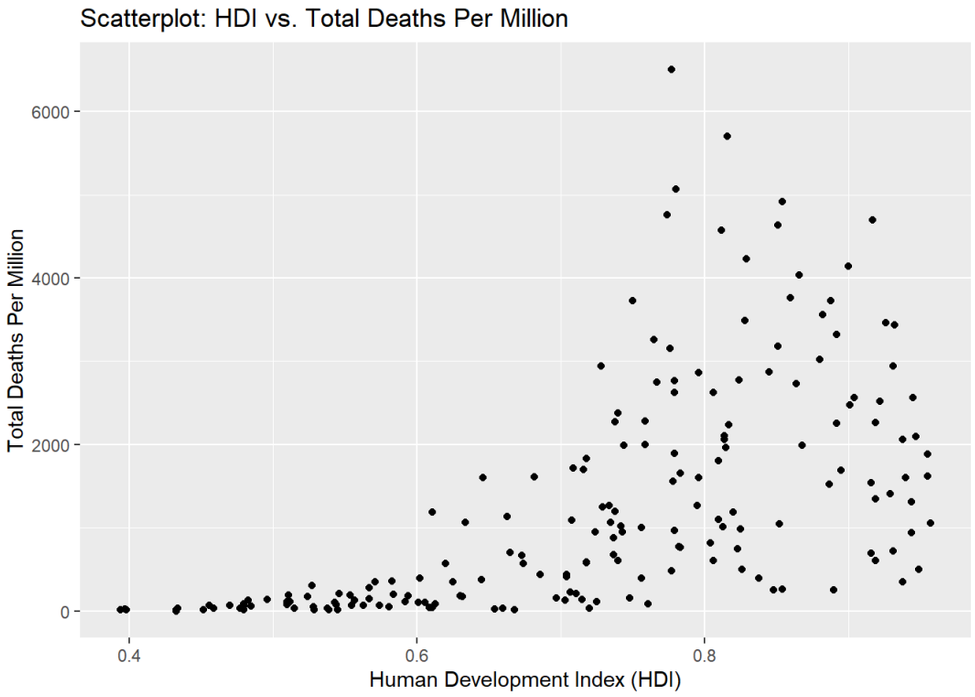
- Make a second table and call it “covid_hdi_vs_deaths”:
    1. Select rows with “date” equal to “2024-01-21”
    2. Select the following columns: “location”, “human_development_index”, “population”, “total_deaths”
    3. Remove rows with NAs
    4. Calculate the total deaths per million people to a column named “total_deaths_per_million” (round the number)

```
# Enter your code here:
covid_hdi_vs_deaths <- subset(covid_world, date == "2024-01-21") %>% select(location, human_development_index, population,
total_deaths) %>% filter(complete.cases(.)) %>% mutate(total_deaths_per_million = round(total_deaths / (population / 1e
6)))
```

- Use a scatterplot to visualize the relations between HDI and the total deaths per million

```
# Enter your code here:
plot_hdi_deaths <- covid_hdi_vs_deaths %>%
  ggplot(aes(x = human_development_index, y = total_deaths_per_million)) +
  geom_point() +
  labs(x = "Human Development Index (HDI)", y = "Total Deaths Per Million") +
  ggtitle("Scatterplot: HDI vs. Total Deaths Per Million")

print(plot_hdi_deaths)
```



- Calculate the Pearson and Spearman correlation between HDI and the total deaths per million

```
# Enter your code here:
pearson_correlation <- cor(covid_hdi_vs_deaths$human_development_index, covid_hdi_vs_deaths$total_deaths_per_million, meth
od = "pearson")

spearman_correlation <- cor(covid_hdi_vs_deaths$human_development_index, covid_hdi_vs_deaths$total_deaths_per_million, met
hod = "spearman")
```

- Report the Spearman correlation coefficient, the strength of the correlation (weak/moderate/strong/very strong) and the direction (negative/positive)

```
Type your answer here:
Spearman Correlation: 0.7523251
This is a strong correlation, the direction is positive. meaning as the HDI increases such is the probability for total de
aths per million people to also increase.
```

- Do you agree with the claim that increased in HDI causes high COVID-19 mortality? (write in detail)

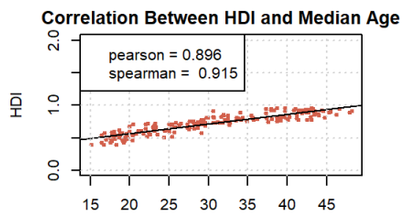
```
Type your answer here:
The claim that increased HDI directly causes higher COVID-19 mortality overlooks crucial factors such as age demographics.
Higher HDI countries often have older populations, and advanced age is a significant risk factor for severe illness and mo
rtality from COVID-19,as we have concluded earlier by the strong coralletion between age and death. Therefore, attributing
higher mortality solely to HDI without considering age distribution would be misleading.
```

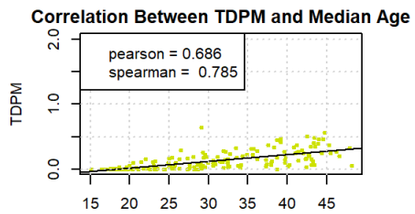
- Find another feature in “covid_world” that explain the correlation between HDI and COVID-19 mortality
- Hint: you already have the answer in the first part of the task

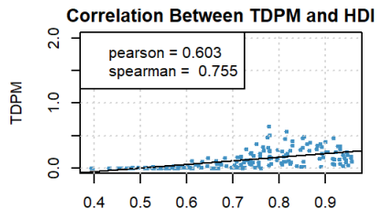
```r
# Enter your code here:
# Function to calculate Pearson and Spearman correlation coefficients
calculate_correlation <- function(x, y) {
  pearson <- cor(x, y, method = "pearson")
  spearman <- cor(x, y, method = "spearman")
  return(c(paste("pearson =", round(pearson, 3)), paste("spearman = ",round(spearman, 3))))
}

covid_hdi_vs_age_deaths <- subset(covid_world, date == "2024-01-21") %>% select(location, human_development_index, populat
ion, total_deaths, median_age) %>% filter(complete.cases(.))  %>% mutate(total_deaths_per_million = (total_deaths / popula
tion)*100)

par(mfrow = c(3, 1), mar = c(10, 6, 2, 2))
par(pin=c(2, 1))
plot(human_development_index ~ median_age,
     data = covid_hdi_vs_age_deaths,
     xlab = "Median Age",
     ylab = "HDI",
     main = "Correlation Between HDI and Median Age",
     pch = 20,
     cex = 0.75,
     ylim = c(0,2),
     col = "#D6604D")
abline(lm(human_development_index ~ median_age, data = covid_hdi_vs_age_deaths), col = "black")
grid()
legend("topleft",
       legend = calculate_correlation(covid_hdi_vs_age_deaths$human_development_index, covid_hdi_vs_age_deaths$median_ag
e))
plot(total_deaths_per_million ~ median_age,
     data = covid_hdi_vs_age_deaths,
     xlab = "Median Age",
     ylab = "TDPM",
     main = "Correlation Between TDPM and Median Age",
     pch = 20,
     cex = 0.75,
     ylim = c(0,2),
     col = "#D1E500")
abline(lm(total_deaths_per_million ~ median_age, data = covid_hdi_vs_age_deaths), col = "black")
grid()
legend("topleft",
       legend = calculate_correlation(covid_hdi_vs_age_deaths$total_deaths_per_million, covid_hdi_vs_age_deaths$median_ag
e))
plot(total_deaths_per_million ~ human_development_index,
     data = covid_hdi_vs_age_deaths,
     xlab = "HDI",
     ylab = "TDPM",
     main = "Correlation Between TDPM and HDI",
     pch = 20,
     cex = 0.75,
     ylim = c(0,2),
     col = "#4393C3")
abline(lm(total_deaths_per_million ~ human_development_index, data = covid_hdi_vs_age_deaths), col = "black")
grid()
legend("topleft",
       legend = calculate_correlation(covid_hdi_vs_age_deaths$total_deaths_per_million, covid_hdi_vs_age_deaths$human_deve
lopment_index))
```







```
Write your explanation here:
The correlation plots indicate a strong positive relationship between Human Development Index (HDI) and Median Age (Plot 1), suggesting that higher HDI countries tend to have older populations. Plots 2 and 3 reveal similar patterns, showing comparable correlations between Total Deaths Per Million and Median Age, as well as between Total Deaths Per Million and HDI. This suggests that the observed correlation between HDI and COVID-19 mortality may be primarily driven by the demographic factor of age. Older populations, prevalent in higher HDI countries, are at greater risk of severe illness and mortality from COVID-19.
```

Task 4 - Sequence Alignment

- Take a look at the “Biostrings” package vignettes

```
browseVignettes("Biostrings")
```

```
## starting httpd help server ... done
```

- Read the file “covid_spike_variants.fasta” This file contain the amino acids sequence of the COVID-19 spike protein from different variants
- Read the file using the correct function from the package “Biostrings” and assign to a variable called “variants”

```
# Enter your code here:
variants <- readAAStringSet("covid_spike_variants.fasta")
print(variants)
```

```
## AAStringSet object of length 6:
##     width seq                                               names
## [1]  1273 MFVFLVLLPLVSSQCVNLTTRTQ...GSCCKFDEDDSEPVLKGVKLHYT 19A_China_Dec19
## [2]  1270 MFVFLVLLPLVSSQCVNLTTRTQ...GSCCKFDEDDSEPVLKGVKLHYT Alpha_UK_Sep20
## [3]  1270 MFVFLVLLPLVSSQCVNLTTRTQ...GSCCKFDEDDSEPVLKGVKLHYT Beta_SouthAfrica_...
## [4]  1273 MFVFLVLLPLVSSQCVNFTNRTQ...GSCCKFDEDDSEPVLKGVKLHYT Gamma_Brazil_Jul20
## [5]  1271 MFVFLVLLPLVSSQCVNLRTRTQ...GSCCKFDEDDSEPVLKGVKLHYT Delta_India_Dec20
## [6]  1273 MFVFLVLLPLVSIQCVNLTTRTQ...GSCCKFDEDDSEPVLKGVKLHYT Epsilon_USA_Sep20
```

- How many amino acids are in the Alpha variant?

```
Type your answer here:
1270
```

- Read the documentation for the Multiple Sequence Alignment (msa) function from the package “msa”

```
?msa()
```

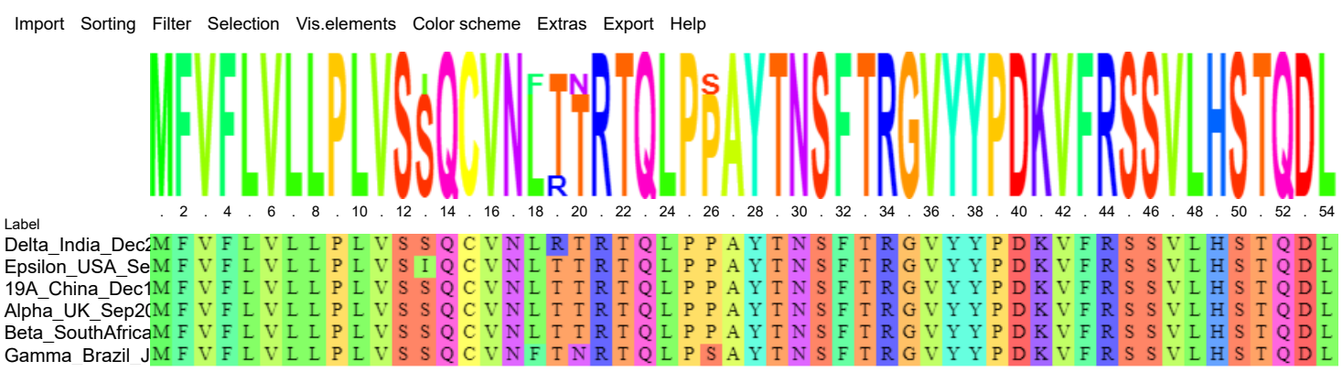
- Run MSA for the sequences in “variants” and assign the results to “variants.msa”

```
# Enter your code here:
variants.msa <- msa(variants)
```

```
## use default substitution matrix
```

- Take a look at the results using the package “msaR” (notice: you can scroll right and left to see all the sequence)

```
msaR(AAMultipleAlignment(variants.msa), colorscheme = "taylor")
```



- 1. Which amino-acids appear in position #13? (type “AMINO_ACID_CODE” in the console)
- 2. Write an example of a SNP that can cause the change in amino acid as we see in the Epsilon variant

```
Type your answer here:
(1) Isoleucine and Serine appears in position #13
(2) The codon "AGU" codes for the amino acid Serine
    The codon "AUU" codes for the amino acid Isoleucine
    SNP that occurs at the second position of the codon, changes it from Serine to Isoleucine.
```

Phylogenetic tree for the COVID19 variants: - Use the package “seqinr” to generate a distance matrix from the MSA results and save it in a variable called “distMat”

```
# Enter your code here:
variants.msa.seqinr <- msaConvert(variants.msa, type="seqinr::alignment")
distMat <- seqinr::dist.alignment(variants.msa.seqinr)
distMat
```

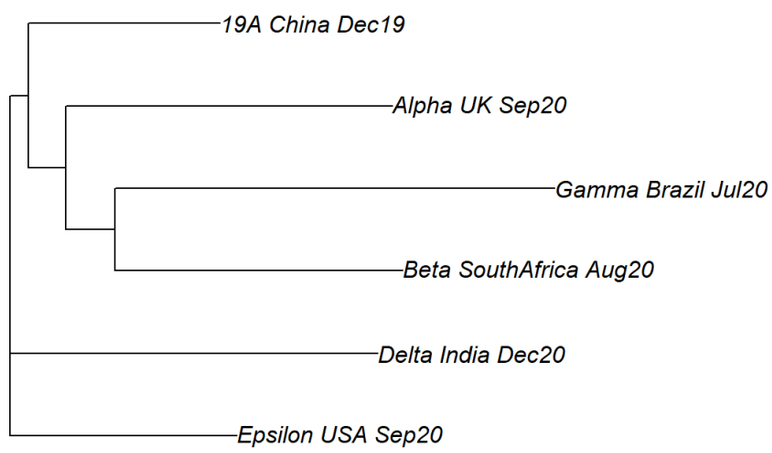
```
##                       Delta_India_Dec20 Epsilon_USA_Sep20 19A_China_Dec19
## Epsilon_USA_Sep20             0.07933635
## 19A_China_Dec19               0.07933635        0.05605519
## Alpha_UK_Sep20                0.09728167        0.08418203      0.07424157
## Beta_SouthAfrica_Aug20        0.10125397        0.08418203      0.07424157
## Gamma_Brazil_Jul20            0.11900453        0.10486965      0.09709043
##                       Alpha_UK_Sep20 Beta_SouthAfrica_Aug20
## Epsilon_USA_Sep20
## 19A_China_Dec19
## Alpha_UK_Sep20
## Beta_SouthAfrica_Aug20     0.08884064
## Gamma_Brazil_Jul20         0.10867853             0.09720504
```

- Use the package “ape” to apply “neighbor-joining” (nj) clustering algorithm to construct a phylogenetic tree
- Save the results in a variable called “ptree”

```
# Enter your code here:
ptree <- nj(distMat)
```

- Look at the phylogenetic tree and answer the following questions:

```
ape::plot.phylo(ptree)
```



- 1. Which variant is the closest to Gamma?
- 2. Which pair seems to emerged from the Alpha variant?
    a. Epsilon and Delta
    b. Beta and Gamma
    c. Epsilon and Delta

```
Type your answer here:
(1) The variant closest to Gamma is "Beta South Africa"
(2) (b) Beta and Gamma
```

- Save this R Markdown as HTML before submitting

1. Go to “Knit” > “Knit to HTML”
2. Change the file name to your IDs (i.e, “123456789_123456789.html”)