Introduction to R

Lesson 2

5th September 2024

Now we open R studio

1

Molecular_Approaches_Clinical_Microbiology_2024 / course_data /

2

Name

- ..
- Quanstudio_files
- RStudio
- bioinformatics
- Error_face.md
- README.md

3

Errors Faced and changes made:

1. Changing file permission `sudo chmod 777 filename` another alternative is `sudo chmod a+rwx filename`
2. Rstudio not opening and crashing `sudo sysctl -w kernel.apparmor_restrict_unprivileged_userns=0`
3. Rstudio package installation line `install.packages(c("data.table", "janitor", "dplyr", "stringr", "stringi","tidyverse","ggplot2","tidyr", "ggpubr","plotly","arsenal","cowplot", "openxlsx"))`

Paste it on your Terminal and press Enter. Now you can open R studio

By the end of today

- View and Understand data
- Selecting Columns
- Filtering Rows
- Mutating/Adding Columns
- Grouping and Summarizing Data
- Plotting in R

Recap

✗ Load the packages

✓
```
library(data.table)
```

✓
```
library(data.table)
```

✓
```
require(data.table)
```

✓
```
male <- c(20, 30, 40)
```

✓
```
female <- c(40, 30, 20)
```

✓
```
month <- c("Jan", "Feb")
```

✗
```
data.frame(male, female, month)
```

✗
```
data_Africa <- read_excel[path = "../inputs/African_meningococci.xlsx"]
```

✗
```
data_Africa <- read_excel(path = ../inputs/African_meningococci.xlsx)
```

View and Understanding your data

```r
#==============================================
# 1. to know the structure of your data
str(data_01_Africa)

# OR

glimpse(data_01_Africa)

#==================================
# 2. view the column names
names(data_01_Africa)

# OR

colnames(data_01_Africa)

#=================================================
# 3. tidy up the column names.
data_02_Africa <- clean_names(data_01_Africa)

# confirm that the names of the columns have changed
names(data_01_Africa)
names(data_02_Africa)
```

```r
#==================================================================
# 4. To view the first top rows - by default will view 6
data_03_Africa <- head(data_01_Africa)


#==================================================================
# 5. To view the bottom rows - by default will view the last 6
data_04_Africa <- tail(data_01_Africa)

#==================================================================
# 6. Selecting columns
data_01_subset <- select(data_01_Africa, id, isolate, country)

# OR (remove certain columns)

data_02_subset <- select(data_01_Africa, -id, -isolate)

#=========================================================
# 7. Filtering rows

Nigeria <- filter(data_01_Africa, country == "Nigeria")


#=========================================
# 8. Number of samples in each country
table(data_01_Africa$country)
```

Questions we can answer?

1. The number of samples in each year? [hint: table]

2. The number of samples in each serogroup?

3. The number of samples collected in each year only in Togo?
[hint: filter then table]

Answers

1. The number of samples in each year? [hint: table]

2011	2012	2013	2014	2015	2016
115	272	60	26	158	85

2. The number of samples in each serogroup?

A	C	NG	W	X	Y
90	124	8	431	61	2

3. The number of samples collected in each year only in Togo?
[hint: filter then table]

2014	2015	2016
16	12	42

Too many variables

- Pipe operator: **%>%**
- Can I clean col names, select, filter all in one variable and run once.

```
#==========================================================================#
# remove all the data frames loaded except the data_01_Africa
rm(data_02_Africa, data_03_Africa, data_04_Africa, Nigeria, data_01_subset, data_02_subset)
```

```
# 9. clean names
data_02_Africa <- data_01_Africa %>%
  clean_names()
```

```
# 9. clean names | selecting columns
data_02_Africa <- data_01_Africa %>%
  clean_names()  %>%
  select(id, isolate, year, country)
```

```
# 9. clean names | selecting columns | Filter only "Burkina Faso"
data_02_Africa <- data_01_Africa %>%
  clean_names()  %>%
  select(id, isolate, year, country) %>%
  filter(country == "Burkina Faso")
```

The number of samples each year per country

country	year	count
Benin	2012	41
Burkina Faso	2012	167
Burkina Faso	2011	41
Burkina Faso	2013	20
Burkina Faso	2015	11
Burkina Faso	2016	5
Burkina Faso	2014	4
Cameroon	2012	4
Central African Republic	2016	23
Central African Republic	2015	7

Grouping and summarizing data

- **group_by** and **summarise** function

```
#==========================================================
# 10. Number of samples identified in each year per Country
data_03_Africa <- data_01_Africa %>%
  clean_names() %>%
  group_by(country, year) %>%
  summarise(n())
```

```
# 10. Number of samples identified in each year per Country
data_03_Africa <- data_01_Africa %>%
  clean_names() %>%
  group_by(country, year) %>%
  summarise(count = n())
```

More questions we can answer

1. The clonal complexes identified in each genogroup? [hint: group_by and summarise]

2. The number of clonal complexes identified by country? [hint: group_by and summarise]

Include the proportions of each

country	clonal_complex_mlst	count	prop
Benin	ST-11 complex	38	92.6829268
Benin	ST-181 complex	3	7.3170732
Burkina Faso	ST-10217 complex	1	0.4032258
Burkina Faso	ST-11 complex	194	78.2258065
Burkina Faso	ST-175 complex	2	0.8064516
Burkina Faso	ST-181 complex	48	19.3548387

```
# 11. Number of samples identified in each year per Country
data_04_Africa <- data_01_Africa %>%
  clean_names() %>%
  group_by(country, clonal_complex_mlst) %>%
  summarise(count = n()) %>%
  mutate(prop = count/sum(count) *100)
```

Task

1. Which Country has the highest proportion of Serogroup W?

2. Create one table showing the number of each serogroup by year and country****

The number of each serogroup by year and country

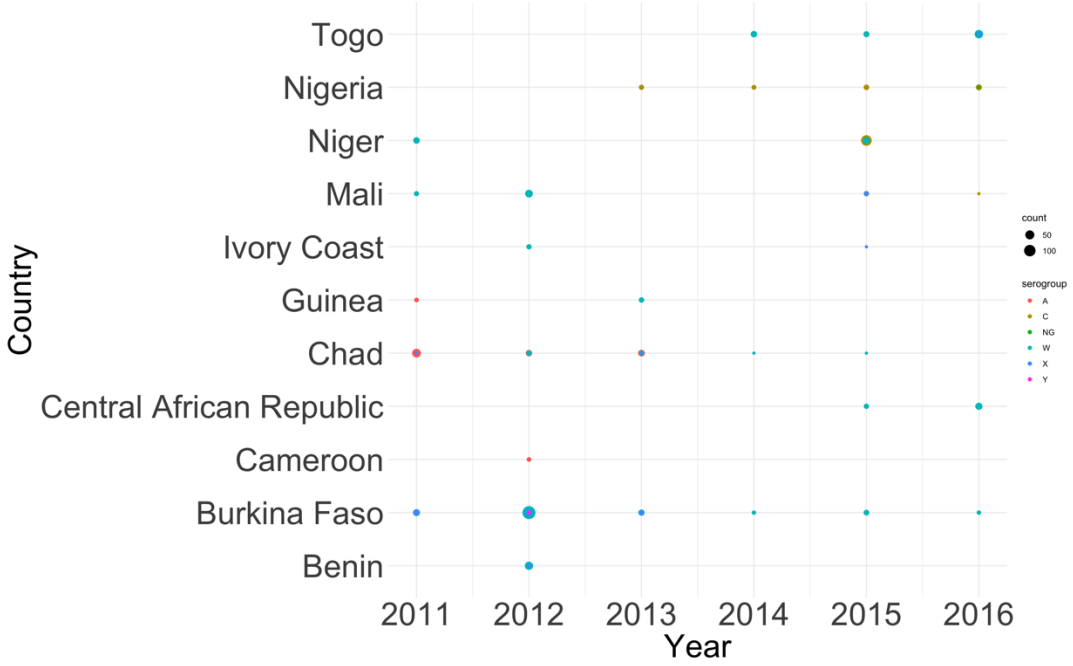
country	year	serogroup	count	prop
Benin	2012	W	38	92.682927
Benin	2012	X	3	7.317073
Burkina Faso	2011	A	1	2.439024
Burkina Faso	2011	W	21	51.219512
Burkina Faso	2011	X	19	46.341463
Burkina Faso	2012	NG	3	1.796407
Burkina Faso	2012	W	140	83.832335
Burkina Faso	2012	X	22	13.173653
Burkina Faso	2012	Y	2	1.197605
Burkina Faso	2013	W	14	70.000000

```r
# 12. The number of each serogroup by year and country
data_05_Africa <- data_01_Africa %>%
  clean_names() %>%
  group_by(country, year, serogroup) %>%
  summarise(count = n()) %>%
  mutate(prop = count/sum(count) *100)
```

Function: ggplot

country	year	serogroup	count	prop
Benin	2012	W	38	92.682927
Be NG	2012	X	3	7.317073
Burkina Faso	2011	A	1	2.439024
Burkina Faso	2011	W	21	51.219512
Burkina Faso	2011	X	19	46.341463
Burkina Faso	2012	NG	3	1.796407
Burkina Faso	2012	W	140	83.832335
Burkina Faso	2012	X	22	13.173653
Burkina Faso	2012	Y	2	1.197605
Burkina Faso	2013	W	14	70.000000
Burkina Faso	2013	X	6	30.000000
Burkina Faso	2014	NG	1	25.000000
Burkina Faso	2014	W	3	75.000000
Burkina Faso	2015	NG	1	9.090909
Burkina Faso	2015	W	10	90.909091
Burkina Faso	2016	NG	2	40.000000
Burkina Faso	2016	W	3	60.000000
Cameroon	2012	A	4	100.000000
Central African Republic	2015	W	7	100.000000
Central African Republic	2016	W	23	100.000000
Chad	2011	A	45	95.744681
Chad	2011	W	1	2.127660
Chad	2011	X	1	2.127660
Chad	2012	A	16	69.565217
Chad	2012	W	7	30.434783
Chad	2013	A	19	79.166667
Chad	2013	W	4	16.666667
Chad	2013	X	1	4.166667
Chad	2014	W	1	100.000000
Chad	2015	W	1	100.000000
Guinea	2011	A	4	100.000000
Guinea	2013	A	1	11.111111
Guinea	2013	W	8	88.888889
Ivory Coast	2012	W	7	100.000000
Ivory Coast	2015	X	1	100.000000
Mali	2011	W	6	100.000000
Mali	2012	W	30	100.000000
Mali	2015	C	8	53.333333

OR



Plotting the number of Serogroups in each Country by Year

```r
# 13. Visualize with points on a graph
# We use the function ggplot, which is under ggplot2 package
library(ggplot2)
plot_Africa <- data_05_Africa %>%
  ggplot(aes(x = year, y = country)) +
  geom_point(aes(color = serogroup, size = count), alpha = 6)
```

```r
# 13. Visualize with points on a graph
# We use the function ggplot, which is under ggplot2 package
library(ggplot2)
plot_Africa <- data_05_Africa %>%
  ggplot(aes(x = year, y = country)) +
  geom_point(aes(color = serogroup, size = count), alpha = 6) +
  xlab("Year") +
  ylab("Country") +
  theme_minimal() +
  theme(axis.title = element_text(size = 35),
        axis.text = element_text(size = 35),
        legend.text = element_text(size = 14),       # Increase legend text size
        legend.title = element_text(size = 16),      # Increase legend title size
        legend.key.size = unit(1.5, "lines"))
```

Task

- Plot the Serogroup C distribution by Country for each Year