**Data Visualisation using RShiny: Day 1**

**1.1 - 1.4 Worksheet**

**1.1   Getting to know your data**

1.1a Explore our dataset

**1.2   Data subsetting and summarising**

1.2a Subsetting data

1.2b Summarising data

1.2c Mutating your data and using the pipe

**1.3   Building exploratory plots**

1.3a Practice plotting with ggplot2

**1.4   Building an interactive plot in RShiny**

1.4a Create a shiny app

1.4b Add an option to a radio button widget

1.4c Add a new text output

1.4d Make a simple change to the data used by a widget

1.4e Make a more complicated change to the data used by a widget

1.4f Change a widget

1.4g Explore a “Go” button

* 1. **Getting to know your data**

**1.1a   Explore our dataset**

Try out the following commands to get to know the data:

1. nrow(raw\_data)

How many entries does the data frame have?

1. names(raw\_data)

What are the names of the first 3 columns?

1. dim(raw\_data)

What are the dimensions of our data?

1. summary(raw\_data)

What is the mean age of the organisms infected?

1. head(raw\_data)

In which region did the 1st case occur?

1. tail(raw\_data)

In which region did the last case occur?

1. str(raw\_data)

Which variables are numerical (num)?

1. unique(raw\_data$species)

What types of species do we have in the data?

1. table(raw\_data$species)

Which species has the most cases?

1. ?names()

What is the first argument for the function names()?

**1.2   Data subsetting and summarising**

**1.2a   Subsetting data**

Make sure you have dplyr loaded into your R session by using:

library(dplyr)

What do each of these lines of code subset the data for? 

1. select(raw\_data, x, y, age)
2. filter(raw\_data, region %in% c("Mara", "Pwani", "Dar-es-salaam"))
3. filter(raw\_data, age >= 30)

**1.2b   Summarising data**

1. What do each of these lines of code summarise?
2. summarise(raw\_data, n\_males = length(which(sex=="M")))
3. summarise(raw\_data, total\_age = sum(age))
4. How are these two tables different?

df\_1 <- group\_by(raw\_data, sex)

summarise(df\_1,

n\_records = length(sex),

mean\_age = mean(age))

df\_2 <- group\_by(raw\_data, region, sex)

summarise(df\_2,

n\_records = length(sex),

mean\_age = mean(age))

**1.2c   Mutating your data and using the pipe**

1. What do each of these lines of code add as a new column?
2. raw\_data <- mutate(raw\_data, "new\_col\_1" = substr(date, 1, 4))
3. raw\_data <- mutate(raw\_data,

"new\_col\_2" = ifelse(density >= 500,"High", "Low"))

1. Fill in the blanks for the following lines of code in your R script:
   1. Subset for only records with wildlife (i.e. jackal, lion)

raw\_data %>%

\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_)

* 1. Subset for humans, and summarise the mean age per region

raw\_data %>%

\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_) %>%

\_\_\_(region) %>%

\_\_\_(mean\_age = \_\_\_\_\_\_\_\_)

**1.3   Building exploratory plots**

**1.3a   Practice plotting with ggplot2**

Make sure you have the following packages loaded into your R session:

library(ggplot2)

library(lubridate)

library(leaflet)

**Questions:**

Fill in the blanks for the following lines of code in your R script:

1. Create a barplot of species and colour by species.

ggplot() +

geom\_bar(data=raw\_data, aes(x=\_\_\_\_\_\_\_\_, fill = \_\_\_\_\_\_\_\_)) +

theme\_classic()

1. Add a title, x and y axis labels to the barplot of species

ggplot() +

geom\_bar(data=raw\_data, aes(x=\_\_\_\_\_\_\_\_, fill = \_\_\_\_\_\_\_\_)) +

\_\_\_\_\_\_\_\_\_ +

\_\_\_\_\_\_\_\_\_ +

\_\_\_\_\_\_\_\_\_ +

theme\_classic()

**1.4   Building an interactive plot in RShiny**

**1.4a   Create a shiny app**

**Activity: Create an app in a single R script**

1. Open up R Studio
2. On the top menu, click on: File > New File > Shiny Web App
3. Type in a name for your app, click on *Single File (app.R)*, check you are happy with the directory, then click Create!
4. Click on the C:\Users\User A\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\8654C40E.tmp button to run the code and load the app

**Activity: Create an app using multiple R scripts**

1. On the top menu, click on:    File > New File > Shiny Web App
2. Type in a name for your app, click on *Multiple File (ui.R/server.R)*, check you are happy with the directory, then click Create!
3. Run the app as before

**1.4b   Add an option to a radio button widget**

Please navigate to the **Apps/barplot\_1** folder and open up the **ui.R** and **server.R** scripts in R Studio.   
  
This is a simple app that uses our dataset to show a barplot.   
There is only 1 widget - selectInput() - to show a Dropdown menu that selects a variable to show on the x-axis.   
  
Please take a moment to have a look at the server and ui scripts, then run the app.   
We’ve added comments to the 2 scripts to help explain what the code does, but some of it should be familiar. 

**Activity: Add age as an option to view on the x-axis**  
*Hints:*

* *You only need to change the ui file*
* *Remember to explore the dataset to find the correct column name*

**1.4c   Add a new text output**

Please navigate to the **Apps/barplot\_2** folder and open up the **ui.R** and **server.R** scripts in R Studio.   
  
This is the same simple barplot app as before, but with a new verbatimtextOutput() below.   
*Verbatim* simply mean “as is”" - it prints the output to look like code.   
  
Please take a moment to have a look at the server and ui scripts, then run the app. 

**Questions:**

1. How many inputs does the app use?
2. How many outputs/renders does the app use?

**Activity: Add another text output to the side panel**

*Hints:*

* *You will need to change both files*
* *You could show a summary of your data e.g. head(), summary()*
* *You can reference the input as a column name using a set of double square brackets e.g. raw\_data[[input$new\_text]]*

**1.4d   Make a simple change to the data used by a widget**

Please navigate to the **Apps/Day 1 - timeseries\_1** folder and open up the **ui.R** and **server.R** scripts in R Studio.   
  
This is a new app that shows a timeseries plot with a Dropdown menu, which allows the user to select the species plotted. There is an option for “All species”, but if the user selects another option, “All species” is still plotted.   
  
Please take a moment to have a look at the server and ui scripts, then run the app. 

**Questions:**

1. What function is used to subset in the reactive() function?
2. What does the “|” mean in the reactive() function?

**Activity: Change the dropdown from Species to Region**  
*Hints:*

* *You will need to change both files*
* *We have named objects descriptively e.g. input$select\_species; use this to identify where you should make edits*

**1.4e   Make a more complicated change to the data used by a widget**

Please navigate to the **Apps/Day 1 - timeseries\_2** folder and open up the **ui.R** and **server.R** scripts in R Studio.   
  
This is the same timeseries app as before, but with a new checkboxGroupInput() below.   
  
Please take a moment to have a look at the server and ui scripts, then run the app.   
  
**Questions:**

1. Why do we create categories for “All data”, “Both sexes” and “All regions” when summarising the data?
2. In the checkboxGroupInput() function in the ui, we provide a list for the choices given to the user.  
   Which of these (Male, Female, M, F) are present in the data, and which have we set to appear in the app?

**Activity: Change the checkboxes from Sex to Species**

*Hints:*

* *You will need to change both files*
* *When listing options for the widgets, you can provide a list (e.g. region\_list) or set the choices manually (e.g. list("Dog"="dog", "Cat"="cat"...))*
* *For checkboxes, you do not need an option for “Select all” in the ui because the user can automatically select multiple options*

**1.4f   Change a widget**

Please navigate to the **Apps/Day 1 - timeseries\_3** folder and open up the **ui.R** and **server.R** scripts in R Studio.   
  
This is the same timeseries app as before, but with a new sliderInput() below the others.   
  
Please take a moment to have a look at the server and ui scripts, then run the app. 

**Questions:**

1. What does the ”<=“ mean in the reactive() function?
2. What does the value argument in the sliderInput() set?

**Activity: Change from a 1-sided to a 2-sided slider**  
*Hints:*

* *You will need to change both files*
* *Remember to use the ? to view a helpfile if you are stuck*
* *If an input produces multiple values, they appear in the ui as a character string (vector)*

**1.4g   Explore a “Go” button**

Please navigate to the **Apps/Day 1 - timeseries\_4** folder and open up the **ui.R** and **server.R** scripts in R Studio.   
  
This is the same timeseries app as before, but with 2 new actionButton() below the others.   
  
Please take a moment to have a look at the server and ui scripts, then run the app. 

**Questions:**

We’ll discuss this app as a group, so please take a moment to think about these questions:

1. What reactive elements have changed in the ui code and the server code?
2. Can you spot any new functions in the shinyServer that we have not introduced yet? What do these do?
3. Can you think of any scenarios where it would be helpful to use an actionButton() to trigger a reactive event?