STARTING APP

# ---------------------------------------------------------------------------- #

# ACTIVITY 1.4e TIMESERIES\_2 SERVER

# ---------------------------------------------------------------------------- #

# Load in libraries

library(shiny)

library(dplyr)

library(ggplot2)

library(RColorBrewer)

library(lubridate)

# Load in the raw data

raw\_data <- read.csv("data/raw\_data.csv", stringsAsFactors=FALSE)

# Create a colour palette

col\_palette <- brewer.pal(name="Dark2", n=8)

# Collect list of years

raw\_data$date <- as.Date(raw\_data$date) # Change the structure of 'date' to a date

yrs <- c(unique(year(raw\_data$date)), 2015) #Extract year and take only the unique years

# Set plot breaks

plot\_breaks = seq(from=0, to=12\*length(yrs)-1, by=12)

# Summarise for 'all' data - this will always be plotted

overall\_summary <- raw\_data %>%

group\_by(month) %>%

summarise(n = length(month)) %>%

mutate(region="All data",

sex="All data")

# Summarise for 'all' sexes, divided by region data

region\_allsexes\_summary <- raw\_data %>%

group\_by(month, region) %>%

summarise(n = length(month)) %>%

mutate(sex="Both sexes")

# Summarise for 'all' region, divided by sex data

sex\_allregions\_summary <- raw\_data %>%

group\_by(month, sex) %>%

summarise(n = length(month)) %>%

mutate(region="All Regions")

# Summarise region and sex data

region\_sexes\_summary <- raw\_data %>%

group\_by(month, region, sex) %>%

summarise(n = length(month))

# Join summary data together

summary\_data <- bind\_rows(overall\_summary, region\_allsexes\_summary, sex\_allregions\_summary, region\_sexes\_summary)

#------------------------------------------------------------------------------#

# Begin server section

shinyServer(function(input, output) {

# Subset for the chosen region and sexes

data\_subset <- reactive({

# Subset for region

data\_sub = summary\_data %>%

filter(region==input$select\_region | region=="All data")

# Subset for sex

if(length(input$select\_sex)>0){

data\_sub = data\_sub %>%

filter(sex %in% input$select\_sex | sex=="Both sexes" | sex=="All data")

} else {

data\_sub = data\_sub %>%

filter(sex=="Both sexes" | sex=="All data")

}

# Summarise data using subsets created above

data\_sub = data\_sub %>%

group\_by(month, region, sex) %>%

summarise(n = sum(n))

as.data.frame(data\_sub)

})

# Produce plot

output$tsPlot <- renderPlot({

ggplot() +

geom\_path(data=data\_subset(), aes(x=month, y=n, color=sex), size=1) +

scale\_color\_manual(name="Sex", values=col\_palette) +

labs(title=input$select\_region, x="Date (Month)", y="Number of records") +

# Extra plotting code to control appearence

scale\_x\_continuous(breaks=plot\_breaks, labels=yrs, limits=c(min(overall\_summary$month), max(overall\_summary$month))) +

theme\_classic() +

theme(axis.text = element\_text(size=14),

axis.title = element\_text(size=18),

plot.title = element\_text(size=20),

legend.title = element\_text(size=18),

legend.text = element\_text(size=14))

})

})

STARTING APP

# ---------------------------------------------------------------------------- #

# ACTIVITY 1.4e TIMESERIES\_2 UI

# ---------------------------------------------------------------------------- #

# Load in libraries

library(shiny)

library(dplyr)

library(ggplot2)

# Load in the raw data

raw\_data <- read.csv("data/raw\_data.csv", stringsAsFactors=FALSE)

# Collect a list of regions for the dropdown menu

region\_list <- c("All Regions", sort(unique(raw\_data$region)))

#------------------------------------------------------------------------------#

# Begin ui section

shinyUI(fluidPage(

# Application title

titlePanel("Day 1 - Timeseries\_2"),

# Add a line break

br(),

# Add text section

h4("This app is identical to the last, with a new widget: checkboxGroupInput"),

h4("Using these widgets together, we can change the region and the sex we want to view on the plot. The line showing 'all data' will always be visible!"),

# Add a line break

br(),

# Sidebar with a slider input for number of bins

sidebarLayout(

sidebarPanel(

# Add a dropdown menu widget

selectInput("select\_region", label = h3("Select a Region:"),

choices = region\_list,

selected = 1),

br(),

# Add a checkbox widget

checkboxGroupInput("select\_sex", label = h3("Select a sex"),

choices = list("Male" = "M", "Female" = "F"),

selected = c("M", "F"))

),

# Show plot

mainPanel(

plotOutput("tsPlot", height=700)

)

)

))

COMPLETED APP

# ---------------------------------------------------------------------------- #

# ACTIVITY 1.4e TIMESERIES\_2 SERVER

# ---------------------------------------------------------------------------- #

# Load in libraries

library(shiny)

library(dplyr)

library(ggplot2)

library(RColorBrewer)

library(lubridate)

# Load in the raw data

raw\_data <- read.csv("data/raw\_data.csv", stringsAsFactors=FALSE)

# Create a colour palette

col\_palette <- brewer.pal(name="Dark2", n=8)

# Collect list of years

raw\_data$date <- as.Date(raw\_data$date) # Change the structure of 'date' to a date

yrs <- c(unique(year(raw\_data$date)), 2015) #Extract year and take only the unique years

# Set plot breaks

plot\_breaks = seq(from=0, to=12\*length(yrs)-1, by=12)

# Summarise for 'all' data - this will always be plotted

overall\_summary <- raw\_data %>%

group\_by(month) %>%

summarise(n = length(month)) %>%

mutate(region="All data",

species="All data")

# Summarise for 'all' species, divided by region data

region\_allspecies\_summary <- raw\_data %>%

group\_by(month, region) %>%

summarise(n = length(month)) %>%

mutate(species="All species")

# Summarise for 'all' region, divided by sex data

species\_allregions\_summary <- raw\_data %>%

group\_by(month, species) %>%

summarise(n = length(month)) %>%

mutate(region="All Regions")

# Summarise region and species data

region\_species\_summary <- raw\_data %>%

group\_by(month, region, species) %>%

summarise(n = length(month))

# Join summary data together

summary\_data <- bind\_rows(overall\_summary, region\_allspecies\_summary, species\_allregions\_summary, region\_species\_summary)

#------------------------------------------------------------------------------#

# Begin server section

shinyServer(function(input, output) {

# Subset for the chosen region and sexes

data\_subset <- reactive({

# Subset for region

data\_sub = summary\_data %>%

filter(region==input$select\_region | region=="All data")

# Subset for sex

if(length(input$select\_species)>0){

data\_sub = data\_sub %>%

filter(species %in% input$select\_species | species=="All species" | species=="All data")

} else {

data\_sub = data\_sub %>%

filter(species=="All species" | species=="All data")

}

# Summarise data using subsets created above

data\_sub = data\_sub %>%

group\_by(month, region, species) %>%

summarise(n = sum(n))

as.data.frame(data\_sub)

})

# Produce plot

output$tsPlot <- renderPlot({

ggplot() +

geom\_path(data=data\_subset(), aes(x=month, y=n, color=species), size=1) +

scale\_color\_manual(name="Species", values=col\_palette) +

labs(title=input$select\_region, x="Date (Month)", y="Number of records") +

# Extra plotting code to control appearence

scale\_x\_continuous(breaks=plot\_breaks, labels=yrs, limits=c(min(overall\_summary$month), max(overall\_summary$month))) +

theme\_classic() +

theme(axis.text = element\_text(size=14),

axis.title = element\_text(size=18),

plot.title = element\_text(size=20),

legend.title = element\_text(size=18),

legend.text = element\_text(size=14))

})

})

COMPLETED APP

# ---------------------------------------------------------------------------- #

# ACTIVITY 1.4e TIMESERIES\_2 UI

# ---------------------------------------------------------------------------- #

# Load in libraries

library(shiny)

library(dplyr)

library(ggplot2)

# Load in the raw data

raw\_data <- read.csv("data/raw\_data.csv", stringsAsFactors=FALSE)

# Collect a list of regions for the dropdown menu

region\_list <- c("All Regions", sort(unique(raw\_data$region)))

#------------------------------------------------------------------------------#

# Begin ui section

shinyUI(fluidPage(

# Application title

titlePanel("Exploratory plots: Timeseries\_2 (Master)"),

# Add a line break

br(),

# Add text section

h4("This app is identical to the last, with a new widget: checkboxGroupInput"),

h4("Using these widgets, we can change the region and the sex we want to view on the plot. The line showing 'all data' will always be visible!"),

# Add a line break

br(),

# Sidebar with a slider input for number of bins

sidebarLayout(

sidebarPanel(

# Add a dropdown menu widget

selectInput("select\_region", label = h3("Select a Region:"),

choices = region\_list,

selected = 1),

br(),

# Add a checkbox widget

checkboxGroupInput("select\_species", label = h3("Select a Species"),

choices = list("Cat" = "cat", "Dog" = "dog", "Human"="human", "Jackal"="jackal", "Lion"="lion"),

selected = c("cat", "dog", "human", "jackal", "lion"))

),

# Show plot

mainPanel(

plotOutput("tsPlot", height=700)

)

)

))