#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# Dashboard to display the impact of UADD's data literacy training

#

# To visualise Attendee Characteristics, Feedback Poll data, and Summary Statistics

#

# Written by: Esther Heggie August 2024

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# Install libraries

#install.packages("shinydashboard")

# Import libraries

library(shiny)

library(ggplot2)

library(tidyverse)

library(readxl)

library(dplyr)

library(shinydashboard)

library(scales)

library(plotly)

library(mathjaxr)

# Import Data===================================================================

#===============================================================================

# Feedback data-----------------------------------------------------------------

Master\_feedback <- read\_xlsx("2024\_Master\_Feedback.xlsx", sheet = "Poll", col\_names = TRUE)

TTAS\_feedback <- read\_xlsx("2024\_10\_Things\_Feedback.xlsx", sheet = "Poll", col\_names = TRUE)

WCwD\_feedback <- read\_xlsx("2024\_WCwD\_Feedback.xlsx", sheet = "Poll", col\_names = TRUE)

DV1\_feedback <- read\_xlsx("2024\_Data\_Vis\_1\_Feedback.xlsx", sheet = "Poll", col\_names = TRUE)

DV2\_feedback <- read\_xlsx("2024\_Data\_Vis\_2\_Feedback.xlsx", sheet = "Poll", col\_names = TRUE)

QUIZ\_feedback <- read\_xlsx("2024\_QUIZ\_Feedback.xlsx", sheet = "Poll", col\_names = TRUE)

# Summary Statistics data-------------------------------------------------------

Master\_summary\_statistics <- read\_xlsx("2024\_Master\_Attendance.xlsx", sheet = "Attendance", col\_names = TRUE)

TTAS\_summary\_statistics <- read\_xlsx("2024\_10\_Things\_Attendance.xlsx", sheet = "Attendance", col\_names = TRUE)

WCwD\_summary\_statistics <- read\_xlsx("2024\_WCwD\_Attendance.xlsx", sheet = "Attendance", col\_names = TRUE)

DV1\_summary\_statistics <- read\_xlsx("2024\_Data\_Vis\_1\_Attendance.xlsx", sheet = "Attendance", col\_names = TRUE)

DV2\_summary\_statistics <- read\_xlsx("2024\_Data\_Vis\_2\_Attendance.xlsx", sheet = "Attendance", col\_names = TRUE)

QUIZ\_summary\_statistics <- read\_xlsx("2024\_QUIZ\_Attendance.xlsx", sheet = "Attendance", col\_names = TRUE)

# Attendee Characteristics data-------------------------------------------------

Master\_attendee\_characteristics <- read\_xlsx("2024\_Master\_Attendance.xlsx", sheet = "Registration", col\_names = TRUE)

TTAS\_attendee\_characteristics <- read\_xlsx("2024\_10\_Things\_Attendance.xlsx", sheet = "Registration", col\_names = TRUE)

WCwD\_attendee\_characteristics <- read\_xlsx("2024\_WCwD\_Attendance.xlsx", sheet = "Registration", col\_names = TRUE)

DV1\_attendee\_characteristics <- read\_xlsx("2024\_Data\_Vis\_1\_Attendance.xlsx", sheet = "Registration", col\_names = TRUE)

DV2\_attendee\_characteristics <- read\_xlsx("2024\_Data\_Vis\_2\_Attendance.xlsx", sheet = "Registration", col\_names = TRUE)

QUIZ\_attendee\_characteristics <- read\_xlsx("2024\_QUIZ\_Attendance.xlsx", sheet = "Registration", col\_names = TRUE)

# User Interface================================================================

#===============================================================================

ui <- dashboardPage(

skin = "green",

dashboardHeader(title = "Impact of Training"),

# Creates side bar to select which training you want to look into

dashboardSidebar(

sidebarMenu(

style = "white-space: normal;", # Makes side bar text stay in side bar

# Page names

menuItem("Introduction", tabName = "Introduction"),

menuItem("Defra Demographics", tabName = "Defra\_demographics"),

menuItem("All Courses", tabName = "All\_Courses"),

menuItem("10 Things You Need to Know About Statistics", tabName = "10\_Things"),

menuItem("Working Confidently with Data", tabName = "WCwD"),

menuItem("Introduction to Data Visualisation : Part 1", tabName = "data\_vis1"),

menuItem("Introduction to Data Visualisation : Part 2", tabName = "data\_vis2"),

menuItem("QUIZ! Understanding Analytical Guidance", tabName = "barbie\_quiz")

)),

# Creates main dashboard body

dashboardBody(

tabItems(

# Introduction page=======================================================

tabItem(tabName = "Introduction",

fluidPage(

# Introduction to dashboard

titlePanel("Welcome to the Impact of Training dashboard")),

fluidRow(

column(4,

tags$p("This app has been produced by the Use of Analysis and Data

in Decision making (UADD) team."),

tags$p("It contains data for the attendee characteristics,

feedback polls, and summary statistics from each of the

training session run by UADD. (Personal data is not used).

For one of our training sessions we also collect the slido

data from the quiz element of the training."),

tags$strong("UADD TRAINING - SESSION DESCRIPTIONS"),

br(),

tags$strong("10 Things You Need to Know about Statistics"),

tags$p("This session presents common pitfalls and issues

around understanding and communicating data. It will

help you build your skills in:"),

tags$ul(

tags$li("interpreting statistical evidence, "),

tags$li("flagging and understanding potential issues, and "),

tags$li("thinking about effective ways to explain and

present findings to a decision maker. ")

),

tags$p("Intended audience: Non-specialists who want to feel

more confident in understanding and drawing together

statistical evidence. This course is particularly

relevant to those in the policy profession and those

who need to interpret data as part of their role. "),

tags$strong("Working Confidently with Data"),

tags$p("An introduction to data literacy and why it’s important.

This session will explain how to find meaning in numbers

(understanding stories and quality) and help you

communicate data insights effectively."),

tags$p("Intended audience: Non-specialists who want to feel

more confident understanding and talking about data."),

tags$strong("Introduction to Data Visualisation"),

tags$p("This session introduces five principles to follow when

using data to inform decisions. They will help you

develop effective data visualisations that convey your

messages clearly and with impact."),

tags$p("Intended audience: Non-specialists who use data in

their work, whether developing visualisations themselves

or commissioning them from others. Also, specialists

looking for a refresher in presenting analysis impactfully."),

tags$strong("QUIZ! Understanding Analytical Guidance"),

tags$p("A fun and engaging way to understand and learn more

about analytical guidance and good practice. It covers

both cross-Government and Defra-specific guidance. "),

tags$p("Intended audience: Defra group staff who are members

of an analytical profession or anyone who wants to

understand more about analytical guidance and good practice. "),

tags$strong("Eligibility for all training sessions:"),

tags$p("All Defra group staff."),

tags$strong("IMPORTANT NOTE:"),

tags$p("During the process of creating this dashboard (August 2024),

the registration form and attendance data collection for our

training has been improved and standardised for the dashboard

to use. Data collected prior to this has a slightly lower

data quality but it's overall effect on how we measure the

impact of training will decrease as we run the sessions more."),

tags$strong("Formulas used in the dashboard"),

uiOutput("formulas"),

tags$strong("How to use this app:"),

tags$ul(

tags$li("Make the side-bar appear and disappear using the

three lines at the top."),

tags$li("Use the side bar to select which page to view, i.e.

which training."),

tags$li("Use the tab headers along the top of the page to

view different areas of analysis of impact of

training for each training."),

tags$li("If you use any statistics or charts from this

dashboard in your work, please reference this tool

as: UADD Training Impact dashboard and the date on

which you extracted the data."),

),

tags$strong("Important information:"),

tags$p("Data for 2024 is not complete as we update after a

training session is run. "),

tags$strong("If you have any questions about this dashboard,

please contact:"),

tags$ul(

tags$li("UADD team: UADD@defra.gov.uk"),

tags$li("Hannah Bateman: Hannah.Bateman@defra.gov.uk"),

),

tags$p("App built by: Esther Heggie, August 2024"),

),

column(width = 8, imageOutput("defra\_logo")),

),

),

# Defra Demographics======================================================

tabItem(tabName = "Defra\_demographics",

fluidPage(

titlePanel("Defra Demographics")),

fluidRow(

tags$strong("Proportion of attendance to Core Defra headcount"),

tags$p("These number were taken from Civil Service Statistics

bulletin (from the excel sheet attached to Figure 3.3 then table 20)

on 20/08/2024 and have been coded as static numbers.

These will needed to be updated as and when the statistical bulletin is updated."),

uiOutput("headcount"),

tags$p("Data caveats:"),

tags$ul(

tags$li("The data won't capture the movement of Core Defra

employees that leave after undertaking training"),

tags$li("The data doesn't account for the movement of employees

within Core Defar between grades"),

tags$li("The grades of Core Defra and ALBs do not align,

Grade 6s and 7s are the other way around. Therefore,

this slightly impacts the data quality when analysising

attendee characteristics for each training type"),

tags$li("Our training data for grades has been grouped in the

same way at the statistical bulletin for the donut

graphs below for better comparisons"),

tags$li("This doesn't account for employees undertaking

multiple trainings. Since we do not use personal ID

data, we can't track whether someone is being tracked

multiple times."),

),

tags$p("We will use the donut graphs below to monitor whether

the proportion of each grade in attendance at our training

is representative of the proportion of Defra headcount

that the grade makes up."),

h3("Proportion of each grade that makes up attendees per training type"),

box(title = "Grade proportion of Core Defra",

background = "light-blue", solidHeader = TRUE, plotlyOutput("Master\_prop\_grade\_donut"),),

box(title = "10 Things You Need to Know about Statistics",

background = "teal", solidHeader = TRUE, plotlyOutput("TTAS\_prop\_grade\_donut"),),

box(title = "Working Confidently with Data",

background = "teal", solidHeader = TRUE, plotlyOutput("WCwD\_prop\_grade\_donut"),),

box(title = "Introduction to Data Visualisations : Part 1",

background = "teal", solidHeader = TRUE, plotlyOutput("DV1\_prop\_grade\_donut"),),

box(title = "Introduction to Data Visualisations : Part 2",

background = "teal", solidHeader = TRUE, plotlyOutput("DV2\_prop\_grade\_donut"),),

box(title = "QUIZ! Understanding Analytical Guidance",

background = "teal", solidHeader = TRUE, plotlyOutput("QUIZ\_prop\_grade\_donut"),),

),

),

# All Courses=============================================================

tabItem(tabName = "All\_Courses",

fluidPage(

titlePanel("Feedback Polls from All Courses")),

tabsetPanel(

# Attendee Characteristics tab----------------------------------

tabPanel("Attendee Characteristics",

fluidRow(

box(title = "Grade Analysis", background = "teal",

solidHeader = TRUE,

actionButton("grade\_caveat1", "Show data caveat"),

plotOutput("Master\_grade")),

box(title = "DG Group Analysis", background = "light-blue",

solidHeader = TRUE,

actionButton("DG\_caveat1", "Show data caveat"),

plotOutput("Master\_DG\_group")),

box(title = "Profession Analysis", background = "light-blue",

solidHeader = TRUE,

actionButton("profession\_caveat1", "Show data caveat"),

plotOutput("Master\_profession")),

box(title = "ALB Analysis", background = "teal",

solidHeader = TRUE,

actionButton("ALB\_caveat1", "Show data caveat"),

plotOutput("Master\_ALB")),

),

),

# Feedback Poll data analysis tab

tabPanel("Feedback Poll",

fluidRow(

valueBoxOutput("Master\_num\_responses"),

valueBoxOutput("Master\_feedback\_response\_rate"),

box(title = "Did you find this session useful?", background = "teal",

solidHeader = TRUE, plotOutput("Master\_useful\_plot")),

box(title = "Did you find this session engaging?", background = "light-blue",

solidHeader = TRUE, plotOutput("Master\_engaging\_plot")),

box(title = "Did you learn something from this session?", background = "light-blue",

solidHeader = TRUE, plotOutput("Master\_learning\_plot")),

box(title = "Would you recommend this session to a colleague", background = "teal",

solidHeader = TRUE, plotOutput("Master\_recommend\_plot")),

),

),

# Summary Statistics tab----------------------------------------

tabPanel("Summary Statistics",

fluidRow(

valueBoxOutput("Master\_prop\_attend\_box"),

valueBoxOutput("Master\_num\_attendees\_box"),

valueBoxOutput("Master\_num\_registrations\_box"),

valueBoxOutput("Master\_prop\_register\_box"),

valueBoxOutput("Master\_num\_page\_views\_box"),

valueBoxOutput("Master\_num\_cancel\_box"),

valueBoxOutput("Master\_avg\_attend\_time\_box"),

valueBoxOutput("Master\_train\_duration\_box"),

valueBoxOutput("Master\_meet\_duration\_box"),

),

),

),

),

# 10 Things page==========================================================

tabItem(tabName = "10\_Things",

fluidPage(

titlePanel("10 Things You Need to Know about Statistics")),

tabsetPanel(

# Attendee Characteristics tab----------------------------------

tabPanel("Attendee Characteristics",

fluidRow(

box(title = "Grade Analysis", background = "teal",

solidHeader = TRUE,

actionButton("grade\_caveat2", "Show data caveat"),

plotOutput("TTAS\_grade")),

box(title = "DG Group Analysis", background = "light-blue",

solidHeader = TRUE,

actionButton("DG\_caveat2", "Show data caveat"),

plotOutput("TTAS\_DG\_group")),

box(title = "Profession Analysis", background = "light-blue",

solidHeader = TRUE,

actionButton("profession\_caveat2", "Show data caveat"),

plotOutput("TTAS\_profession")),

box(title = "ALB Analysis", background = "teal",

solidHeader = TRUE,

actionButton("ALB\_caveat2", "Show data caveat"),

plotOutput("TTAS\_ALB")),

),

),

# Feedback Polls tab--------------------------------------------

tabPanel("Feedback Polls",

fluidRow(

valueBoxOutput("TTAS\_num\_responses"),

valueBoxOutput("TTAS\_feedback\_response\_rate"),

box(title = "Did you find this session useful?", background = "teal",

solidHeader = TRUE, plotOutput("TTAS\_useful\_plot")),

box(title = "Did you find this session engaging?", background = "light-blue",

solidHeader = TRUE, plotOutput("TTAS\_engaging\_plot")),

box(title = "Did you learn something from this session?", background = "light-blue",

solidHeader = TRUE, plotOutput("TTAS\_learning\_plot")),

box(title = "Would you recommend this session to a colleague", background = "teal",

solidHeader = TRUE, plotOutput("TTAS\_recommend\_plot")),

),

),

# Summary Statistics tab----------------------------------------

tabPanel("Summary Statistics",

fluidRow(

valueBoxOutput("TTAS\_prop\_attend\_box"),

valueBoxOutput("TTAS\_num\_attendees\_box"),

valueBoxOutput("TTAS\_num\_registrations\_box"),

valueBoxOutput("TTAS\_prop\_register\_box"),

valueBoxOutput("TTAS\_num\_page\_views\_box"),

valueBoxOutput("TTAS\_num\_cancel\_box"),

valueBoxOutput("TTAS\_avg\_attend\_time\_box"),

valueBoxOutput("TTAS\_train\_duration\_box"),

valueBoxOutput("TTAS\_meet\_duration\_box"),

),

),

),

),

# WCwD page===============================================================

tabItem(tabName = "WCwD",

fluidPage(

titlePanel("Working Confidently with Data")),

tabsetPanel(

# Attendee Characteristics tab----------------------------------

tabPanel("Attendee Characteristics",

fluidRow(

box(title = "Grade Analysis", background = "teal",

solidHeader = TRUE,

actionButton("grade\_caveat3", "Show data caveat"),

plotOutput("WCwD\_grade")),

box(title = "DG Group Analysis", background = "light-blue",

solidHeader = TRUE,

actionButton("DG\_caveat3", "Show data caveat"),

plotOutput("WCwD\_DG\_group")),

box(title = "Profession Analysis", background = "light-blue",

solidHeader = TRUE,

actionButton("profession\_caveat3", "Show data caveat"),

plotOutput("WCwD\_profession")),

box(title = "ALB Analysis", background = "teal",

solidHeader = TRUE,

actionButton("ALB\_caveat3", "Show data caveat"),

plotOutput("WCwD\_ALB")),

),

),

# Feedback Polls tab--------------------------------------------

tabPanel("Feedback Polls",

fluidRow(

valueBoxOutput("WCwD\_num\_responses"),

valueBoxOutput("WCwD\_feedback\_response\_rate"),

box(title = "Did you find this session useful?", background = "teal",

solidHeader = TRUE, plotOutput("WCwD\_useful\_plot")),

box(title = "Did you find this session engaging?", background = "light-blue",

solidHeader = TRUE, plotOutput("WCwD\_engaging\_plot")),

box(title = "Did you learn something from this session?", background = "light-blue",

solidHeader = TRUE, plotOutput("WCwD\_learning\_plot")),

box(title = "Would you recommend this session to a colleague", background = "teal",

solidHeader = TRUE, plotOutput("WCwD\_recommend\_plot")),

),

),

# Summary Statistics tab----------------------------------------

tabPanel("Summary Statistics",

fluidRow(

valueBoxOutput("WCwD\_prop\_attend\_box"),

valueBoxOutput("WCwD\_num\_attendees\_box"),

valueBoxOutput("WCwD\_num\_registrations\_box"),

valueBoxOutput("WCwD\_prop\_register\_box"),

valueBoxOutput("WCwD\_num\_page\_views\_box"),

valueBoxOutput("WCwD\_num\_cancel\_box"),

valueBoxOutput("WCwD\_avg\_attend\_time\_box"),

valueBoxOutput("WCwD\_train\_duration\_box"),

valueBoxOutput("WCwD\_meet\_duration\_box"),

),

),

),

),

# Data vis 1 page=========================================================

tabItem(tabName = "data\_vis1",

fluidPage(

titlePanel("Introduction to Data Visualisations : Part 1")),

tabsetPanel(

# Attendee Characteristics tab----------------------------------

tabPanel("Attendee Characteristics",

fluidRow(

box(title = "Grade Analysis", background = "teal",

solidHeader = TRUE,

actionButton("grade\_caveat4", "Show data caveat"),

plotOutput("DV1\_grade")),

box(title = "DG Group Analysis", background = "light-blue",

solidHeader = TRUE,

actionButton("DG\_caveat4", "Show data caveat"),

plotOutput("DV1\_DG\_group")),

box(title = "Profession Analysis", background = "light-blue",

solidHeader = TRUE,

actionButton("profession\_caveat4", "Show data caveat"),

plotOutput("DV1\_profession")),

box(title = "ALB Analysis", background = "teal",

solidHeader = TRUE,

actionButton("ALB\_caveat4", "Show data caveat"),

plotOutput("DV1\_ALB")),

),

),

# Feedback Polls tab--------------------------------------------

tabPanel("Feedback Polls",

fluidRow(

valueBoxOutput("DV1\_num\_responses"),

valueBoxOutput("DV1\_feedback\_response\_rate"),

box(title = "Did you find this session useful?", background = "teal",

solidHeader = TRUE, plotOutput("DV1\_useful\_plot")),

box(title = "Did you find this session engaging?", background = "light-blue",

solidHeader = TRUE, plotOutput("DV1\_engaging\_plot")),

box(title = "Did you learn something from this session?", background = "light-blue",

solidHeader = TRUE, plotOutput("DV1\_learning\_plot")),

box(title = "Would you recommend this session to a colleague", background = "teal",

solidHeader = TRUE, plotOutput("DV1\_recommend\_plot")),

),

),

# Summary Statistics tab----------------------------------------

tabPanel("Summary Statistics",

fluidRow(

valueBoxOutput("DV1\_prop\_attend\_box"),

valueBoxOutput("DV1\_num\_attendees\_box"),

valueBoxOutput("DV1\_num\_registrations\_box"),

valueBoxOutput("DV1\_prop\_register\_box"),

valueBoxOutput("DV1\_num\_page\_views\_box"),

valueBoxOutput("DV1\_num\_cancel\_box"),

valueBoxOutput("DV1\_avg\_attend\_time\_box"),

valueBoxOutput("DV1\_train\_duration\_box"),

valueBoxOutput("DV1\_meet\_duration\_box"),

),

),

),

),

# Data vis 2 page=========================================================

tabItem(tabName = "data\_vis2",

fluidPage(

titlePanel("Introduction to Data Visualisations : Part 2")),

tabsetPanel(

# Attendee Characteristics tab----------------------------------

tabPanel("Attendee Characteristics",

fluidRow(

box(title = "Grade Analysis", background = "teal",

solidHeader = TRUE,

actionButton("grade\_caveat5", "Show data caveat"),

plotOutput("DV2\_grade")),

box(title = "DG Group Analysis", background = "light-blue",

solidHeader = TRUE,

actionButton("DG\_caveat5", "Show data caveat"),

plotOutput("DV2\_DG\_group")),

box(title = "Profession Analysis", background = "light-blue",

solidHeader = TRUE,

actionButton("profession\_caveat5", "Show data caveat"),

plotOutput("DV2\_profession")),

box(title = "ALB Analysis", background = "teal",

solidHeader = TRUE,

actionButton("ALB\_caveat5", "Show data caveat"),

plotOutput("DV2\_ALB")),

),

),

# Feedback Polls tab--------------------------------------------

tabPanel("Feedback Polls",

fluidRow(

valueBoxOutput("DV2\_num\_responses"),

valueBoxOutput("DV2\_feedback\_response\_rate"),

box(title = "Did you find this session useful?", background = "teal",

solidHeader = TRUE, plotOutput("DV2\_useful\_plot")),

box(title = "Did you find this session engaging?", background = "light-blue",

solidHeader = TRUE, plotOutput("DV2\_engaging\_plot")),

box(title = "Did you learn something from this session?", background = "light-blue",

solidHeader = TRUE, plotOutput("DV2\_learning\_plot")),

box(title = "Would you recommend this session to a colleague", background = "teal",

solidHeader = TRUE, plotOutput("DV2\_recommend\_plot")),

),

),

# Summary Statistics tab----------------------------------------

tabPanel("Summary Statistics",

fluidRow(

valueBoxOutput("DV2\_prop\_attend\_box"),

valueBoxOutput("DV2\_num\_attendees\_box"),

valueBoxOutput("DV2\_num\_registrations\_box"),

valueBoxOutput("DV2\_prop\_register\_box"),

valueBoxOutput("DV2\_num\_page\_views\_box"),

valueBoxOutput("DV2\_num\_cancel\_box"),

valueBoxOutput("DV2\_avg\_attend\_time\_box"),

valueBoxOutput("DV2\_train\_duration\_box"),

valueBoxOutput("DV2\_meet\_duration\_box"),

),

),

),

),

# Barbie quiz page========================================================

tabItem(tabName = "barbie\_quiz",

fluidPage(

titlePanel("QUIZ! Understanding Analytical Guidance")),

tabsetPanel(

# Attendee Characteristics tab----------------------------------

tabPanel("Attendee Characteristics",

fluidRow(

box(title = "Grade Analysis", background = "teal",

solidHeader = TRUE,

actionButton("grade\_caveat6", "Show data caveat"),

plotOutput("QUIZ\_grade")),

box(title = "DG Group Analysis", background = "light-blue",

solidHeader = TRUE,

actionButton("DG\_caveat6", "Show data caveat"),

plotOutput("QUIZ\_DG\_group")),

box(title = "Profession Analysis", background = "light-blue",

solidHeader = TRUE,

actionButton("profession\_caveat6", "Show data caveat"),

plotOutput("QUIZ\_profession")),

box(title = "ALB Analysis", background = "teal",

solidHeader = TRUE,

actionButton("ALB\_caveat6", "Show data caveat"),

plotOutput("QUIZ\_ALB")),

),

),

# Feedback Polls tab--------------------------------------------

tabPanel("Feedback Polls",

fluidRow(

valueBoxOutput("QUIZ\_num\_responses"),

valueBoxOutput("QUIZ\_feedback\_response\_rate"),

box(title = "Did you find this session useful?", background = "teal",

solidHeader = TRUE, plotOutput("QUIZ\_useful\_plot")),

box(title = "Did you find this session engaging?", background = "light-blue",

solidHeader = TRUE, plotOutput("QUIZ\_engaging\_plot")),

box(title = "Did you learn something from this session?", background = "light-blue",

solidHeader = TRUE, plotOutput("QUIZ\_learning\_plot")),

box(title = "Would you recommend this session to a colleague", background = "teal",

solidHeader = TRUE, plotOutput("QUIZ\_recommend\_plot")),

),

),

# Summary Statistics tab----------------------------------------

tabPanel("Summary Statistics",

fluidRow(

valueBoxOutput("QUIZ\_prop\_attend\_box"),

valueBoxOutput("QUIZ\_num\_attendees\_box"),

valueBoxOutput("QUIZ\_num\_registrations\_box"),

valueBoxOutput("QUIZ\_prop\_register\_box"),

valueBoxOutput("QUIZ\_num\_page\_views\_box"),

valueBoxOutput("QUIZ\_num\_cancel\_box"),

valueBoxOutput("QUIZ\_avg\_attend\_time\_box"),

valueBoxOutput("QUIZ\_train\_duration\_box"),

valueBoxOutput("QUIZ\_meet\_duration\_box"),

),

),

# Slido data tab

tabPanel("Slido data",

fluidRow(

)),

),

)

) # end of tab items

), # end of dashboard body

) # end of ui bracket

# Server Logic==================================================================

#===============================================================================

server <- function(input, output) {

# Functions used within code

# RAG rating for value boxes (because they only accept certain colours)

RAG <- function(value){

if (value <= 0.5){

RAG = "maroon"

}

else if (value < 0.75){

RAG = "orange"

}

else {

RAG = "olive"

}

}

# Doughnut graph code with inbuilt RAG rating

big\_number\_donut\_plot <- function(title, value) {

# RAG rating (using DEFRA style guide colours)

RAG\_donut <- function(value){

if (value <= 0.5){

RAG\_donut = "#BD0A1B"

}

else if (value < 0.75){

RAG\_donut = "#FFCC00"

}

else {

RAG\_donut = "#77BC1F"

}

}

# Wrangle data to get a data frame in the format we need it in to make our donut chart

df <- tibble(x = 1, y = value) %>%

mutate(y\_negative = 1 - y) %>%

pivot\_longer(cols = -x)

# Create a nicely formatted big number to go in the donut hole

big\_number\_text\_label <- percent(value, accuracy = 1)

# Create our plot

ggplot(df,

aes(x = x,

y = value,

fill = name)) +

# Add a bar, but don't add the legend

geom\_col(show.legend = FALSE) +

# Add title

ggtitle(label = title) +

# A pie/donut chart is a bar chart with polar coordinates

# Add polar coordinates and set the direction to -1

# so the filled in part starts at the top and goes clockwise

coord\_polar(theta = "y",

direction = -1) +

# Set the limits, which is important for adding the hole

xlim(c(-2, 2)) +

# Set a color scale with the highlighted section in whatever color

# is chosen with the highlight\_color argument and the rest in a light gray

scale\_fill\_manual(values = c(RAG\_donut(value), "grey90")) +

# Set theme\_void() to remove grid lines and everything else from the plot

theme\_void() +

# Add the big number in the center of the hole

annotate("text",

label = big\_number\_text\_label,

family = "Serif",

fontface = "bold",

color = RAG\_donut(value),

size = 12,

x = -2,

y = 0)

}

# Data caveats================================================================

# Grade Analysis

observeEvent(input$grade\_caveat1 | input$grade\_caveat2 | input$grade\_caveat3 |

input$grade\_caveat4 | input$grade\_caveat5 | input$grade\_caveat6, {

showModal(modalDialog(

title = "Grade data caveat",

"Grades are defined differently between DG Groups and ALBs. This difference

is that Grade 6s and Grade 7s are the opposite way round to DG Groups in ALBs.",

easyClose = TRUE, footer = NULL

))

})

# DG Group Analysis

observeEvent(input$DG\_caveat1 | input$DG\_caveat2 | input$DG\_caveat3 |

input$DG\_caveat4 | input$DG\_caveat5 | input$DG\_caveat6, {

showModal(modalDialog(

title = "DG Group data caveat",

"Since we are unable to root questions in the registration form, some ALB

attendees select a DG Group rather than N/A when filling it in.",

easyClose = TRUE, footer = NULL

))

})

# Profession Analysis

observeEvent(input$profession\_caveat1 | input$profession\_caveat2 | input$profession\_caveat3 |

input$profession\_caveat4 | input$profession\_caveat5 | input$profession\_caveat6, {

showModal(modalDialog(

title = "Profession data caveat",

"ALB attendees won't have specific professions. However, due to being

unable to root questions in the registration form, some ALB attendees might

answer this question with a profession rather than clickly the N/A option.",

easyClose = TRUE, footer = NULL

))

})

# ALB Analysis

observeEvent(input$ALB\_caveat1 | input$ALB\_caveat2 | input$ALB\_caveat3 |

input$ALB\_caveat4 | input$ALB\_caveat5 | input$ALB\_caveat6, {

showModal(modalDialog(

title = "ALB data caveat",

"The collection of ALB data was not standardised until August 2024.

Data collected before the standardisation of the registration form will be

missing data entries that were not inputted in the way the code reads it.",

easyClose = TRUE, footer = NULL

))

})

# Introduction================================================================

output$formulas <- renderUI({

withMathJax(

helpText('Feedback response rate: $$\\frac{Number of responses}{Number in attendance} \* 100$$'),

helpText('Attendees as a proportion of those who registered: $$\\frac{Number of attendees}{Number of registrations} \* 100$$'),

helpText('Proportion of people who viewed registration page that then signed up: $$\\frac{Number of registrations}{Number of page views} \* 100$$')

)

})

output$defra\_logo <- renderImage({

list(src = "Defra-logo.png",

width = 300, height = 200)

}, deleteFile = F)

# DEFRA DEMOGRAPHICS==========================================================

url <- a("Civil Service statistical bulletin",

href="https://www.gov.uk/government/statistics/civil-service-statistics-2023/statistical-bulletin-civil-service-statistics-2023#grade")

output$headcount <- renderUI({tagList("Link to data source:", url)})

# Grade proportion analysis by training type

# Master

output$Master\_prop\_grade\_donut <- renderPlotly({

# Donut graph of Grade

# create a data frame with category labels and corresponding values

data <- data.frame(

category = c("AO/AA", "EO", "HEO/SEO", "Grade 7/Grade 6", "SCS"),

value = c(2540, 2240, 5030, 2990, 235)

)

# define the colors for each category

colors <- c("#E6F2F8", "#CADAE4", "#8DA8BF", "#44729A", "#01538A")

# set the marker properties, including the colors and line width

marker <- list(colors = colors)

# create the pie chart with a hole in the center

plot\_ly(data, labels = ~category, values = ~value, type = "pie",

hole = 0.5, marker = marker)

})

# TTAS

output$TTAS\_prop\_grade\_donut <- renderPlotly({

# Donut graph of Grade

# create a data frame with category labels and corresponding values

data <- data.frame(

category = c("AO/AA", "EO", "HEO/SEO", "Grade 7/Grade 6", "SCS"),

value = c(length(which(TTAS\_attendee\_characteristics[2] == "AO/AA")),

length(which(TTAS\_attendee\_characteristics[2] == "EO")),

length(which(TTAS\_attendee\_characteristics[2] == "HEO")) + length(which(TTAS\_attendee\_characteristics[2] == "SEO")),

length(which(TTAS\_attendee\_characteristics[2] == "Grade 7")) + length(which(TTAS\_attendee\_characteristics[2] == "Grade 6")),

length(which(TTAS\_attendee\_characteristics[2] == "SCS")))

)

# define the colors for each category

colors <- c("#E6F2F8", "#CADAE4", "#8DA8BF", "#44729A", "#01538A")

# set the marker properties, including the colors and line width

marker <- list(colors = colors)

# create the pie chart with a hole in the center

plot\_ly(data, labels = ~category, values = ~value, type = "pie",

hole = 0.5, marker = marker)

})

# WCwD

output$WCwD\_prop\_grade\_donut <- renderPlotly({

# Donut graph of Grade

# create a data frame with category labels and corresponding values

data <- data.frame(

category = c("AO/AA", "EO", "HEO/SEO", "Grade 7/Grade 6", "SCS"),

value = c(length(which(WCwD\_attendee\_characteristics[2] == "AO/AA")),

length(which(WCwD\_attendee\_characteristics[2] == "EO")),

length(which(WCwD\_attendee\_characteristics[2] == "HEO")) + length(which(WCwD\_attendee\_characteristics[2] == "SEO")),

length(which(WCwD\_attendee\_characteristics[2] == "Grade 7")) + length(which(WCwD\_attendee\_characteristics[2] == "Grade 6")),

length(which(WCwD\_attendee\_characteristics[2] == "SCS")))

)

# define the colors for each category

colors <- c("#E6F2F8", "#CADAE4", "#8DA8BF", "#44729A", "#01538A")

# set the marker properties, including the colors and line width

marker <- list(colors = colors)

# create the pie chart with a hole in the center

plot\_ly(data, labels = ~category, values = ~value, type = "pie",

hole = 0.5, marker = marker)

})

# DV1

output$DV1\_prop\_grade\_donut <- renderPlotly({

# Donut graph of Grade

# create a data frame with category labels and corresponding values

data <- data.frame(

category = c("AO/AA", "EO", "HEO/SEO", "Grade 7/Grade 6", "SCS"),

value = c(length(which(DV1\_attendee\_characteristics[2] == "AO/AA")),

length(which(DV1\_attendee\_characteristics[2] == "EO")),

length(which(DV1\_attendee\_characteristics[2] == "HEO")) + length(which(DV1\_attendee\_characteristics[2] == "SEO")),

length(which(DV1\_attendee\_characteristics[2] == "Grade 7")) + length(which(DV1\_attendee\_characteristics[2] == "Grade 6")),

length(which(DV1\_attendee\_characteristics[2] == "SCS")))

)

# define the colors for each category

colors <- c("#E6F2F8", "#CADAE4", "#8DA8BF", "#44729A", "#01538A")

# set the marker properties, including the colors and line width

marker <- list(colors = colors)

# create the pie chart with a hole in the center

plot\_ly(data, labels = ~category, values = ~value, type = "pie",

hole = 0.5, marker = marker)

})

# DV2

output$DV2\_prop\_grade\_donut <- renderPlotly({

# Donut graph of Grade

# create a data frame with category labels and corresponding values

data <- data.frame(

category = c("AO/AA", "EO", "HEO/SEO", "Grade 7/Grade 6", "SCS"),

value = c(length(which(DV2\_attendee\_characteristics[2] == "AO/AA")),

length(which(DV2\_attendee\_characteristics[2] == "EO")),

length(which(DV2\_attendee\_characteristics[2] == "HEO")) + length(which(DV2\_attendee\_characteristics[2] == "SEO")),

length(which(DV2\_attendee\_characteristics[2] == "Grade 7")) + length(which(DV2\_attendee\_characteristics[2] == "Grade 6")),

length(which(DV2\_attendee\_characteristics[2] == "SCS")))

)

# define the colors for each category

colors <- c("#E6F2F8", "#CADAE4", "#8DA8BF", "#44729A", "#01538A")

# set the marker properties, including the colors and line width

marker <- list(colors = colors)

# create the pie chart with a hole in the center

plot\_ly(data, labels = ~category, values = ~value, type = "pie",

hole = 0.5, marker = marker)

})

# QUIZ

output$QUIZ\_prop\_grade\_donut <- renderPlotly({

# Donut graph of Grade

# create a data frame with category labels and corresponding values

data <- data.frame(

category = c("AO/AA", "EO", "HEO/SEO", "Grade 7/Grade 6", "SCS"),

value = c(length(which(QUIZ\_attendee\_characteristics[2] == "AO/AA")),

length(which(QUIZ\_attendee\_characteristics[2] == "EO")),

length(which(QUIZ\_attendee\_characteristics[2] == "HEO")) + length(which(QUIZ\_attendee\_characteristics[2] == "SEO")),

length(which(QUIZ\_attendee\_characteristics[2] == "Grade 7")) + length(which(QUIZ\_attendee\_characteristics[2] == "Grade 6")),

length(which(QUIZ\_attendee\_characteristics[2] == "SCS")))

)

# define the colors for each category

colors <- c("#E6F2F8", "#CADAE4", "#8DA8BF", "#44729A", "#01538A")

# set the marker properties, including the colors and line width

marker <- list(colors = colors)

# create the pie chart with a hole in the center

plot\_ly(data, labels = ~category, values = ~value, type = "pie",

hole = 0.5, marker = marker)

})

# ALL COURSES=================================================================

# Attendee Characteristics----------------------------------------------------

# Grade analysis

output$Master\_grade <- renderPlot({

x = c("AO/AA", "EO", "HEO", "SEO", "Grade 7", "Grade 6", "SCS")

y = c(length(which(Master\_attendee\_characteristics[2] == "AO/AA")),

length(which(Master\_attendee\_characteristics[2] == "EO")),

length(which(Master\_attendee\_characteristics[2] == "HEO")),

length(which(Master\_attendee\_characteristics[2] == "SEO")),

length(which(Master\_attendee\_characteristics[2] == "Grade 7")),

length(which(Master\_attendee\_characteristics[2] == "Grade 6")),

length(which(Master\_attendee\_characteristics[2] == "SCS")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

xlab("Grade") + ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# DG group analysis

output$Master\_DG\_group <- renderPlot({

x = c("Environment", "Food, Biosecurity, and Trade", "Portfolio Delivery",

"Strategy", "Science and Analysis")

y = c(length(which(Master\_attendee\_characteristics[3] == "Environment")),

length(which(Master\_attendee\_characteristics[3] == "Food, Biosecurity, and Trade")),

length(which(Master\_attendee\_characteristics[3] == "Portfolio Delivery")),

length(which(Master\_attendee\_characteristics[3] == "Strategy")),

length(which(Master\_attendee\_characteristics[3] == "Science and Analysis")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) +

geom\_bar(stat = "identity", fill = "navy") + coord\_flip() + xlab("DG Group") +

ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Profession analysis

output$Master\_profession <- renderPlot({

x = c("Operational Delivery",

"Policy",

"Commercial",

"Communications",

"Digital and Data",

"Economics",

"Finance",

"Geography",

"Human Resources",

"Project Delivery",

"Property",

"Operational Research",

"Social Research",

"Statistics",

"Legal",

"Science and Engineering",

"Veterinary",

"Other")

y = c(length(which(Master\_attendee\_characteristics[1] == "Operational Delivery")),

length(which(Master\_attendee\_characteristics[1] == "Policy")),

length(which(Master\_attendee\_characteristics[1] == "Commercial")),

length(which(Master\_attendee\_characteristics[1] == "Communications")),

length(which(Master\_attendee\_characteristics[1] == "Digital and Data")),

length(which(Master\_attendee\_characteristics[1] == "Economics")),

length(which(Master\_attendee\_characteristics[1] == "Finance")),

length(which(Master\_attendee\_characteristics[1] == "Geography")),

length(which(Master\_attendee\_characteristics[1] == "Human Resources")),

length(which(Master\_attendee\_characteristics[1] == "Project Delivery")),

length(which(Master\_attendee\_characteristics[1] == "Property")),

length(which(Master\_attendee\_characteristics[1] == "Operational Research")),

length(which(Master\_attendee\_characteristics[1] == "Social Research")),

length(which(Master\_attendee\_characteristics[1] == "Statistics")),

length(which(Master\_attendee\_characteristics[1] == "Legal")),

length(which(Master\_attendee\_characteristics[1] == "Science and Engineering")),

length(which(Master\_attendee\_characteristics[1] == "Veterinary")),

length(which(Master\_attendee\_characteristics[1] == "Other")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) +

geom\_bar(stat = "identity", fill = "navy") + coord\_flip() + xlab("Profession") +

ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# ALB Analysis

output$Master\_ALB <- renderPlot({

x = c("Animal and Plant Health Agency",

"Centre for Environment, Fisheries, and Aquaculture Science",

"Environment Agency", "Forestry Commission", "Marine Management Organisation",

"Natural England", "Rural Payments Agency", "Veterinary Medicines Directorate",

"Other", "N/A")

y = c(length(which(Master\_attendee\_characteristics[4] == "Animal and Plant Health Agenc")),

length(which(Master\_attendee\_characteristics[4] == "Centre for Environment, Fisheries, and Aquaculture Science")),

length(which(Master\_attendee\_characteristics[4] == "Environment Agency")),

length(which(Master\_attendee\_characteristics[4] == "Forestry Commission")),

length(which(Master\_attendee\_characteristics[4] == "Marine Management Organisation")),

length(which(Master\_attendee\_characteristics[4] == "Natural England")),

length(which(Master\_attendee\_characteristics[4] == "Rural Payments Agency")),

length(which(Master\_attendee\_characteristics[4] == "Veterinary Medicines Directorate")),

length(which(Master\_attendee\_characteristics[4] == "Other")),

length(which(Master\_attendee\_characteristics[4] == "N/A")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) + geom\_bar(stat = "identity", fill = "navy") + coord\_flip() +

xlab("ALB") + ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Feedback Polls -------------------------------------------------------------

# Number of responses = number of rows

master\_num\_responses <- function(){nrow(Master\_feedback)}

output$Master\_num\_responses <- renderValueBox({

valueBox(paste0(master\_num\_responses()), "Number of responses", color = "olive")})

# Feedback response rate -

master\_feedback\_response\_rate <- function(){

round(((nrow(Master\_feedback)) / sum(Master\_summary\_statistics[4])) \* 100, digits = 0)}

output$Master\_feedback\_response\_rate <- renderValueBox({

valueBox(paste0(master\_feedback\_response\_rate(), "%"), "Feedback response rate", color = RAG(master\_feedback\_response\_rate() / 100))})

# QUESTION: Did you find this session useful?

output$Master\_useful\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(Master\_feedback[6] == "Strongly Disagree")) + length(which(Master\_feedback[6] == "Strongly disagree")),

length(which(Master\_feedback[6] == "Disagree")),

length(which(Master\_feedback[6] == "Neutral")),

length(which(Master\_feedback[6] == "Agree")),

length(which(Master\_feedback[6] == "Strongly Agree")) + length(which(Master\_feedback[6] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Did you find this session engaging?

output$Master\_engaging\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(Master\_feedback[7] == "Strongly Disagree")) + length(which(Master\_feedback[7] == "Strongly disagree")),

length(which(Master\_feedback[7] == "Disagree")),

length(which(Master\_feedback[7] == "Neutral")),

length(which(Master\_feedback[7] == "Agree")),

length(which(Master\_feedback[7] == "Strongly Agree")) + length(which(Master\_feedback[7] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Did you learn something from this session?

output$Master\_learning\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(Master\_feedback[8] == "Strongly Disagree")) + length(which(Master\_feedback[8] == "Strongly disagree")),

length(which(Master\_feedback[8] == "Disagree")),

length(which(Master\_feedback[8] == "Neutral")),

length(which(Master\_feedback[8] == "Agree")),

length(which(Master\_feedback[8] == "Strongly Agree")) + length(which(Master\_feedback[8] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Would you recommend this session to a colleague

output$Master\_recommend\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(Master\_feedback[9] == "Strongly Disagree")) + length(which(Master\_feedback[9] == "Strongly disagree")),

length(which(Master\_feedback[9] == "Disagree")),

length(which(Master\_feedback[9] == "Neutral")),

length(which(Master\_feedback[9] == "Agree")),

length(which(Master\_feedback[9] == "Strongly Agree")) + length(which(Master\_feedback[9] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Summary Statistics----------------------------------------------------------

# Attendees as a proportion of those who registered

prop\_attend <- function(){

round((as.numeric(sum(Master\_summary\_statistics[4])) /

as.numeric(sum(Master\_summary\_statistics[2]))) \* 100, digits = 0)}

output$Master\_prop\_attend\_box <- renderValueBox({

valueBox(paste0(prop\_attend(), "%"),

"Attendees as a proportion of those who registered", color = RAG(prop\_attend() / 100))})

# Number of attendees

num\_attendees <- function(){sum(Master\_summary\_statistics[4])}

output$Master\_num\_attendees\_box <- renderValueBox({

valueBox(paste0(num\_attendees()), "Number of attendees", color = "light-blue")})

# Number of registrations

num\_registrations <- function(){sum(Master\_summary\_statistics[2])}

output$Master\_num\_registrations\_box <- renderValueBox({

valueBox(paste0(num\_registrations()), "Number of registrations", color = "light-blue")})

# Proportion of people who viewed registration page that then signed up, (num registered + num cancelled) / num views

prop\_register <- function(){

round(((as.numeric(sum(Master\_summary\_statistics[2])) +

as.numeric(sum(Master\_summary\_statistics[3]))) /

as.numeric(sum(Master\_summary\_statistics[1]))) \* 100, digits = 0)}

output$Master\_prop\_register\_box <- renderValueBox({

valueBox(paste0(prop\_register(), "%"),

"Proportion of people who viewed registration page that then signed up",

color = RAG(prop\_register() / 100))})

# Number of page views

num\_page\_views <- function(){sum(Master\_summary\_statistics[1])}

output$Master\_num\_page\_views\_box <- renderValueBox({

valueBox(paste0(num\_page\_views()), "Number of page views", color = "light-blue")})

# Number of cancellations

num\_cancel <- function(){as.numeric(sum(Master\_summary\_statistics[3]))}

output$Master\_num\_cancel\_box <- renderValueBox({

valueBox(paste0(num\_cancel()), "Number of cancellations", color = "light-blue")})

# 10 THINGS YOU NEED TO KNOW ABOUT STATISTICS=================================

# Attendee Characteristics----------------------------------------------------

# Grade analysis

output$TTAS\_grade <- renderPlot({

x = c("AO/AA", "EO", "HEO", "SEO", "Grade 7", "Grade 6", "SCS")

y = c(length(which(TTAS\_attendee\_characteristics[2] == "AO/AA")),

length(which(TTAS\_attendee\_characteristics[2] == "EO")),

length(which(TTAS\_attendee\_characteristics[2] == "HEO")),

length(which(TTAS\_attendee\_characteristics[2] == "SEO")),

length(which(TTAS\_attendee\_characteristics[2] == "Grade 7")),

length(which(TTAS\_attendee\_characteristics[2] == "Grade 6")),

length(which(TTAS\_attendee\_characteristics[2] == "SCS")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

xlab("Grade") + ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# DG group analysis

output$TTAS\_DG\_group <- renderPlot({

x = c("Environment", "Food, Biosecurity, and Trade", "Portfolio Delivery",

"Strategy", "Science and Analysis")

y = c(length(which(TTAS\_attendee\_characteristics[3] == "Environment")),

length(which(TTAS\_attendee\_characteristics[3] == "Food, Biosecurity, and Trade")),

length(which(TTAS\_attendee\_characteristics[3] == "Portfolio Delivery")),

length(which(TTAS\_attendee\_characteristics[3] == "Strategy")),

length(which(TTAS\_attendee\_characteristics[3] == "Science and Analysis")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) +

geom\_bar(stat = "identity", fill = "navy") + coord\_flip() + xlab("DG Group") +

ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Profession analysis

output$TTAS\_profession <- renderPlot({

x = c("Operational Delivery",

"Policy",

"Commercial",

"Communications",

"Digital and Data",

"Economics",

"Finance",

"Geography",

"Human Resources",

"Project Delivery",

"Property",

"Operational Research",

"Social Research",

"Statistics",

"Legal",

"Science and Engineering",

"Veterinary",

"Other")

y = c(length(which(TTAS\_attendee\_characteristics[1] == "Operational Delivery")),

length(which(TTAS\_attendee\_characteristics[1] == "Policy")),

length(which(TTAS\_attendee\_characteristics[1] == "Commercial")),

length(which(TTAS\_attendee\_characteristics[1] == "Communications")),

length(which(TTAS\_attendee\_characteristics[1] == "Digital and Data")),

length(which(TTAS\_attendee\_characteristics[1] == "Economics")),

length(which(TTAS\_attendee\_characteristics[1] == "Finance")),

length(which(TTAS\_attendee\_characteristics[1] == "Geography")),

length(which(TTAS\_attendee\_characteristics[1] == "Human Resources")),

length(which(TTAS\_attendee\_characteristics[1] == "Project Delivery")),

length(which(TTAS\_attendee\_characteristics[1] == "Property")),

length(which(TTAS\_attendee\_characteristics[1] == "Operational Research")),

length(which(TTAS\_attendee\_characteristics[1] == "Social Research")),

length(which(TTAS\_attendee\_characteristics[1] == "Statistics")),

length(which(TTAS\_attendee\_characteristics[1] == "Legal")),

length(which(TTAS\_attendee\_characteristics[1] == "Science and Engineering")),

length(which(TTAS\_attendee\_characteristics[1] == "Veterinary")),

length(which(TTAS\_attendee\_characteristics[1] == "Other")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) +

geom\_bar(stat = "identity", fill = "navy") + coord\_flip() + xlab("Profession") +

ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# ALB Analysis

output$TTAS\_ALB <- renderPlot({

x = c("Animal and Plant Health Agency",

"Centre for Environment, Fisheries, and Aquaculture Science",

"Environment Agency", "Forestry Commission", "Marine Management Organisation",

"Natural England", "Rural Payments Agency", "Veterinary Medicines Directorate",

"Other", "N/A")

y = c(length(which(TTAS\_attendee\_characteristics[4] == "Animal and Plant Health Agenc")),

length(which(TTAS\_attendee\_characteristics[4] == "Centre for Environment, Fisheries, and Aquaculture Science")),

length(which(TTAS\_attendee\_characteristics[4] == "Environment Agency")),

length(which(TTAS\_attendee\_characteristics[4] == "Forestry Commission")),

length(which(TTAS\_attendee\_characteristics[4] == "Marine Management Organisation")),

length(which(TTAS\_attendee\_characteristics[4] == "Natural England")),

length(which(TTAS\_attendee\_characteristics[4] == "Rural Payments Agency")),

length(which(TTAS\_attendee\_characteristics[4] == "Veterinary Medicines Directorate")),

length(which(TTAS\_attendee\_characteristics[4] == "Other")),

length(which(TTAS\_attendee\_characteristics[4] == "N/A")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) + geom\_bar(stat = "identity", fill = "navy") + coord\_flip() +

xlab("ALB") + ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Feedback polls--------------------------------------------------------------

# Number of responses = number of rows

TTAS\_num\_responses <- function(){nrow(TTAS\_feedback)}

output$TTAS\_num\_responses <- renderValueBox({

valueBox(paste0(TTAS\_num\_responses()), "Number of responses", color = "olive")})

# Feedback response rate

TTAS\_feedback\_response\_rate <- function(){

round(((nrow(TTAS\_feedback)) / as.numeric(TTAS\_summary\_statistics[4])) \* 100, digits = 0)}

output$TTAS\_feedback\_response\_rate <- renderValueBox({

valueBox(paste0(TTAS\_feedback\_response\_rate(), "%"), "Feedback response rate", color = RAG(TTAS\_feedback\_response\_rate() / 100))})

# QUESTION: Did you find this session useful?

output$TTAS\_useful\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(TTAS\_feedback[6] == "Strongly Disagree")) + length(which(TTAS\_feedback[6] == "Strongly disagree")),

length(which(TTAS\_feedback[6] == "Disagree")),

length(which(TTAS\_feedback[6] == "Neutral")),

length(which(TTAS\_feedback[6] == "Agree")),

length(which(TTAS\_feedback[6] == "Strongly Agree")) + length(which(TTAS\_feedback[6] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Did you find this session engaging?

output$TTAS\_engaging\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(TTAS\_feedback[7] == "Strongly Disagree")) + length(which(TTAS\_feedback[7] == "Strongly disagree")),

length(which(TTAS\_feedback[7] == "Disagree")),

length(which(TTAS\_feedback[7] == "Neutral")),

length(which(TTAS\_feedback[7] == "Agree")),

length(which(TTAS\_feedback[7] == "Strongly Agree")) + length(which(TTAS\_feedback[7] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Did you learn something from this session?

output$TTAS\_learning\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(TTAS\_feedback[8] == "Strongly Disagree")) + length(which(TTAS\_feedback[8] == "Strongly disagree")),

length(which(TTAS\_feedback[8] == "Disagree")),

length(which(TTAS\_feedback[8] == "Neutral")),

length(which(TTAS\_feedback[8] == "Agree")),

length(which(TTAS\_feedback[8] == "Strongly Agree")) + length(which(TTAS\_feedback[8] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Would you recommend this session to a colleague

output$TTAS\_recommend\_plot <- renderPlot({

# Create x vector to five survey response options

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(TTAS\_feedback[9] == "Strongly Disagree")) + length(which(TTAS\_feedback[9] == "Strongly disagree")),

length(which(TTAS\_feedback[9] == "Disagree")),

length(which(TTAS\_feedback[9] == "Neutral")),

length(which(TTAS\_feedback[9] == "Agree")),

length(which(TTAS\_feedback[9] == "Strongly Agree")) + length(which(TTAS\_feedback[9] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Summary Statistics----------------------------------------------------------

# Attendees as a proportion of those who registered

TTAS\_prop\_attend <- function(){

round((as.numeric(TTAS\_summary\_statistics[4]) /

as.numeric(TTAS\_summary\_statistics[2])) \* 100, digits = 0)}

output$TTAS\_prop\_attend\_box <- renderValueBox({

valueBox(paste0(TTAS\_prop\_attend(), "%"),

"Attendees as a proportion of those who registered", color = RAG(TTAS\_prop\_attend() / 100))})

# Number of attendees

TTAS\_num\_attendees <- function(){TTAS\_summary\_statistics[4]}

output$TTAS\_num\_attendees\_box <- renderValueBox({

valueBox(paste0(TTAS\_num\_attendees()), "Number of attendees", color = "light-blue")})

# Number of registrations

TTAS\_num\_registrations <- function(){TTAS\_summary\_statistics[2]}

output$TTAS\_num\_registrations\_box <- renderValueBox({

valueBox(paste0(TTAS\_num\_registrations()), "Number of registrations", color = "light-blue")})

# Proportion of people who viewed registration page that then signed up, (num registered + num cancelled) / num views

TTAS\_prop\_register <- function(){

round(((as.numeric(TTAS\_summary\_statistics[2]) +

as.numeric(TTAS\_summary\_statistics[3])) /

as.numeric(TTAS\_summary\_statistics[1])) \* 100, digits = 0)}

output$TTAS\_prop\_register\_box <- renderValueBox({

valueBox(paste0(TTAS\_prop\_register(), "%"),

"Proportion of people who viewed registration page that then signed up",

color = RAG(TTAS\_prop\_register() / 100))})

# Number of page views

TTAS\_num\_page\_views <- function(){TTAS\_summary\_statistics[1]}

output$TTAS\_num\_page\_views\_box <- renderValueBox({

valueBox(paste0(TTAS\_num\_page\_views()), "Number of page views", color = "light-blue")})

# Number of cancellations

TTAS\_num\_cancel <- function(){as.numeric(TTAS\_summary\_statistics[3])}

output$TTAS\_num\_cancel\_box <- renderValueBox({

valueBox(paste0(TTAS\_num\_cancel()), "Number of cancellations", color = "light-blue")})

# Average attendance time

TTAS\_avg\_attend\_time <- function(){TTAS\_summary\_statistics[6]}

output$TTAS\_avg\_attend\_time\_box <- renderValueBox({

valueBox(paste0(TTAS\_avg\_attend\_time()), "Average attendance duration", color = "light-blue")})

# Training duration

TTAS\_train\_duration <- function(){"1h 30m 00s"}

output$TTAS\_train\_duration\_box <- renderValueBox({

valueBox(paste0(TTAS\_train\_duration()), "Target training duration", color = "light-blue")})

# Meeting duration

TTAS\_meet\_duration <- function(){TTAS\_summary\_statistics[5]}

output$TTAS\_meet\_duration\_box <- renderValueBox({

valueBox(paste0(TTAS\_meet\_duration()), "Actual training duration", color = "light-blue")})

# WORKING CONFIDENTLY WITH DATA===============================================

# Attendee Characteristics----------------------------------------------------

# Grade analysis

output$WCwD\_grade <- renderPlot({

x = c("AO/AA", "EO", "HEO", "SEO", "Grade 7", "Grade 6", "SCS")

y = c(length(which(WCwD\_attendee\_characteristics[2] == "AO/AA")),

length(which(WCwD\_attendee\_characteristics[2] == "EO")),

length(which(WCwD\_attendee\_characteristics[2] == "HEO")),

length(which(WCwD\_attendee\_characteristics[2] == "SEO")),

length(which(WCwD\_attendee\_characteristics[2] == "Grade 7")),

length(which(WCwD\_attendee\_characteristics[2] == "Grade 6")),

length(which(WCwD\_attendee\_characteristics[2] == "SCS")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

xlab("Grade") + ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# DG group analysis

output$WCwD\_DG\_group <- renderPlot({

x = c("Environment", "Food, Biosecurity, and Trade", "Portfolio Delivery",

"Strategy", "Science and Analysis")

y = c(length(which(WCwD\_attendee\_characteristics[3] == "Environment")),

length(which(WCwD\_attendee\_characteristics[3] == "Food, Biosecurity, and Trade")),

length(which(WCwD\_attendee\_characteristics[3] == "Portfolio Delivery")),

length(which(WCwD\_attendee\_characteristics[3] == "Strategy")),

length(which(WCwD\_attendee\_characteristics[3] == "Science and Analysis")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) +

geom\_bar(stat = "identity", fill = "navy") + coord\_flip() + xlab("DG Group") +

ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Profession analysis

output$WCwD\_profession <- renderPlot({

x = c("Operational Delivery",

"Policy",

"Commercial",

"Communications",

"Digital and Data",

"Economics",

"Finance",

"Geography",

"Human Resources",

"Project Delivery",

"Property",

"Operational Research",

"Social Research",

"Statistics",

"Legal",

"Science and Engineering",

"Veterinary",

"Other")

y = c(length(which(WCwD\_attendee\_characteristics[1] == "Operational Delivery")),

length(which(WCwD\_attendee\_characteristics[1] == "Policy")),

length(which(WCwD\_attendee\_characteristics[1] == "Commercial")),

length(which(WCwD\_attendee\_characteristics[1] == "Communications")),

length(which(WCwD\_attendee\_characteristics[1] == "Digital and Data")),

length(which(WCwD\_attendee\_characteristics[1] == "Economics")),

length(which(WCwD\_attendee\_characteristics[1] == "Finance")),

length(which(WCwD\_attendee\_characteristics[1] == "Geography")),

length(which(WCwD\_attendee\_characteristics[1] == "Human Resources")),

length(which(WCwD\_attendee\_characteristics[1] == "Project Delivery")),

length(which(WCwD\_attendee\_characteristics[1] == "Property")),

length(which(WCwD\_attendee\_characteristics[1] == "Operational Research")),

length(which(WCwD\_attendee\_characteristics[1] == "Social Research")),

length(which(WCwD\_attendee\_characteristics[1] == "Statistics")),

length(which(WCwD\_attendee\_characteristics[1] == "Legal")),

length(which(WCwD\_attendee\_characteristics[1] == "Science and Engineering")),

length(which(WCwD\_attendee\_characteristics[1] == "Veterinary")),

length(which(WCwD\_attendee\_characteristics[1] == "Other")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) +

geom\_bar(stat = "identity", fill = "navy") + coord\_flip() + xlab("Profession") +

ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# ALB Analysis

output$WCwD\_ALB <- renderPlot({

x = c("Animal and Plant Health Agency",

"Centre for Environment, Fisheries, and Aquaculture Science",

"Environment Agency", "Forestry Commission", "Marine Management Organisation",

"Natural England", "Rural Payments Agency", "Veterinary Medicines Directorate",

"Other", "N/A")

y = c(length(which(WCwD\_attendee\_characteristics[4] == "Animal and Plant Health Agenc")),

length(which(WCwD\_attendee\_characteristics[4] == "Centre for Environment, Fisheries, and Aquaculture Science")),

length(which(WCwD\_attendee\_characteristics[4] == "Environment Agency")),

length(which(WCwD\_attendee\_characteristics[4] == "Forestry Commission")),

length(which(WCwD\_attendee\_characteristics[4] == "Marine Management Organisation")),

length(which(WCwD\_attendee\_characteristics[4] == "Natural England")),

length(which(WCwD\_attendee\_characteristics[4] == "Rural Payments Agency")),

length(which(WCwD\_attendee\_characteristics[4] == "Veterinary Medicines Directorate")),

length(which(WCwD\_attendee\_characteristics[4] == "Other")),

length(which(WCwD\_attendee\_characteristics[4] == "N/A")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) + geom\_bar(stat = "identity", fill = "navy") + coord\_flip() +

xlab("ALB") + ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Feedback polls--------------------------------------------------------------

# Number of responses = number of rows

WCwD\_num\_responses <- function(){nrow(WCwD\_feedback)}

output$WCwD\_num\_responses <- renderValueBox({

valueBox(paste0(WCwD\_num\_responses()), "Number of responses", color = "olive")})

# Feedback response rate

WCwD\_feedback\_response\_rate <- function(){

round(((nrow(WCwD\_feedback)) / as.numeric(WCwD\_summary\_statistics[4])) \* 100, digits = 0)}

output$WCwD\_feedback\_response\_rate <- renderValueBox({

valueBox(paste0(WCwD\_feedback\_response\_rate(), "%"), "Feedback response rate", color = RAG(WCwD\_feedback\_response\_rate() / 100))})

# QUESTION: Did you find this session useful?

output$WCwD\_useful\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(WCwD\_feedback[6] == "Strongly Disagree")) + length(which(WCwD\_feedback[6] == "Strongly disagree")),

length(which(WCwD\_feedback[6] == "Disagree")),

length(which(WCwD\_feedback[6] == "Neutral")),

length(which(WCwD\_feedback[6] == "Agree")),

length(which(WCwD\_feedback[6] == "Strongly Agree")) + length(which(WCwD\_feedback[6] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Did you find this session engaging?

output$WCwD\_engaging\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(WCwD\_feedback[7] == "Strongly Disagree")) + length(which(WCwD\_feedback[7] == "Strongly disagree")),

length(which(WCwD\_feedback[7] == "Disagree")),

length(which(WCwD\_feedback[7] == "Neutral")),

length(which(WCwD\_feedback[7] == "Agree")),

length(which(WCwD\_feedback[7] == "Strongly Agree")) + length(which(WCwD\_feedback[7] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Did you learn something from this session?

output$WCwD\_learning\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(WCwD\_feedback[8] == "Strongly Disagree")) + length(which(WCwD\_feedback[8] == "Strongly disagree")),

length(which(WCwD\_feedback[8] == "Disagree")),

length(which(WCwD\_feedback[8] == "Neutral")),

length(which(WCwD\_feedback[8] == "Agree")),

length(which(WCwD\_feedback[8] == "Strongly Agree")) + length(which(WCwD\_feedback[8] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Would you recommend this session to a colleague

output$WCwD\_recommend\_plot <- renderPlot({

# Create x vector to five survey response options

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(WCwD\_feedback[9] == "Strongly Disagree")) + length(which(WCwD\_feedback[9] == "Strongly disagree")),

length(which(WCwD\_feedback[9] == "Disagree")),

length(which(WCwD\_feedback[9] == "Neutral")),

length(which(WCwD\_feedback[9] == "Agree")),

length(which(WCwD\_feedback[9] == "Strongly Agree")) + length(which(WCwD\_feedback[9] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Summary Statistics----------------------------------------------------------

# Attendees as a proportion of those who registered

WCwD\_prop\_attend <- function(){

round((as.numeric(WCwD\_summary\_statistics[4]) /

as.numeric(WCwD\_summary\_statistics[2])) \* 100, digits = 0)}

output$WCwD\_prop\_attend\_box <- renderValueBox({

valueBox(paste0(WCwD\_prop\_attend(), "%"),

"Attendees as a proportion of those who registered", color = RAG(WCwD\_prop\_attend() / 100))})

# Number of attendees

WCwD\_num\_attendees <- function(){WCwD\_summary\_statistics[4]}

output$WCwD\_num\_attendees\_box <- renderValueBox({

valueBox(paste0(WCwD\_num\_attendees()), "Number of attendees", color = "light-blue")})

# Number of registrations

WCwD\_num\_registrations <- function(){WCwD\_summary\_statistics[2]}

output$WCwD\_num\_registrations\_box <- renderValueBox({

valueBox(paste0(WCwD\_num\_registrations()), "Number of registrations", color = "light-blue")})

# Proportion of people who viewed registration page that then signed up, (num registered + num cancelled) / num views

WCwD\_prop\_register <- function(){

round(((as.numeric(WCwD\_summary\_statistics[2]) +

as.numeric(WCwD\_summary\_statistics[3])) /

as.numeric(WCwD\_summary\_statistics[1])) \* 100, digits = 0)}

output$WCwD\_prop\_register\_box <- renderValueBox({

valueBox(paste0(WCwD\_prop\_register(), "%"),

"Proportion of people who viewed registration page that then signed up",

color = RAG(WCwD\_prop\_register() / 100))})

# Number of page views

WCwD\_num\_page\_views <- function(){WCwD\_summary\_statistics[1]}

output$WCwD\_num\_page\_views\_box <- renderValueBox({

valueBox(paste0(WCwD\_num\_page\_views()), "Number of page views", color = "light-blue")})

# Number of cancellations

WCwD\_num\_cancel <- function(){as.numeric(WCwD\_summary\_statistics[3])}

output$WCwD\_num\_cancel\_box <- renderValueBox({

valueBox(paste0(WCwD\_num\_cancel()), "Number of cancellations", color = "light-blue")})

# Average attendance time

WCwD\_avg\_attend\_time <- function(){WCwD\_summary\_statistics[6]}

output$WCwD\_avg\_attend\_time\_box <- renderValueBox({

valueBox(paste0(WCwD\_avg\_attend\_time()), "Average attendance duration", color = "light-blue")})

# Training duration

WCwD\_train\_duration <- function(){"1h 00m 00s"}

output$WCwD\_train\_duration\_box <- renderValueBox({

valueBox(paste0(WCwD\_train\_duration()), "Target training duration", color = "light-blue")})

# Meeting duration

WCwD\_meet\_duration <- function(){WCwD\_summary\_statistics[5]}

output$WCwD\_meet\_duration\_box <- renderValueBox({

valueBox(paste0(WCwD\_meet\_duration()), "Actual training duration", color = "light-blue")})

# INTRODUCTION TO DATA VISUALISATIONS : PART 1================================

# Attendee Characteristics----------------------------------------------------

# Grade analysis

output$DV1\_grade <- renderPlot({

x = c("AO/AA", "EO", "HEO", "SEO", "Grade 7", "Grade 6", "SCS")

y = c(length(which(DV1\_attendee\_characteristics[2] == "AO/AA")),

length(which(DV1\_attendee\_characteristics[2] == "EO")),

length(which(DV1\_attendee\_characteristics[2] == "HEO")),

length(which(DV1\_attendee\_characteristics[2] == "SEO")),

length(which(DV1\_attendee\_characteristics[2] == "Grade 7")),

length(which(DV1\_attendee\_characteristics[2] == "Grade 6")),

length(which(DV1\_attendee\_characteristics[2] == "SCS")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

xlab("Grade") + ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# DG group analysis

output$DV1\_DG\_group <- renderPlot({

x = c("Environment", "Food, Biosecurity, and Trade", "Portfolio Delivery",

"Strategy", "Science and Analysis")

y = c(length(which(DV1\_attendee\_characteristics[3] == "Environment")),

length(which(DV1\_attendee\_characteristics[3] == "Food, Biosecurity, and Trade")),

length(which(DV1\_attendee\_characteristics[3] == "Portfolio Delivery")),

length(which(DV1\_attendee\_characteristics[3] == "Strategy")),

length(which(DV1\_attendee\_characteristics[3] == "Science and Analysis")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) +

geom\_bar(stat = "identity", fill = "navy") + coord\_flip() + xlab("DG Group") +

ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Profession analysis

output$DV1\_profession <- renderPlot({

x = c("Operational Delivery",

"Policy",

"Commercial",

"Communications",

"Digital and Data",

"Economics",

"Finance",

"Geography",

"Human Resources",

"Project Delivery",

"Property",

"Operational Research",

"Social Research",

"Statistics",

"Legal",

"Science and Engineering",

"Veterinary",

"Other")

y = c(length(which(DV1\_attendee\_characteristics[1] == "Operational Delivery")),

length(which(DV1\_attendee\_characteristics[1] == "Policy")),

length(which(DV1\_attendee\_characteristics[1] == "Commercial")),

length(which(DV1\_attendee\_characteristics[1] == "Communications")),

length(which(DV1\_attendee\_characteristics[1] == "Digital and Data")),

length(which(DV1\_attendee\_characteristics[1] == "Economics")),

length(which(DV1\_attendee\_characteristics[1] == "Finance")),

length(which(DV1\_attendee\_characteristics[1] == "Geography")),

length(which(DV1\_attendee\_characteristics[1] == "Human Resources")),

length(which(DV1\_attendee\_characteristics[1] == "Project Delivery")),

length(which(DV1\_attendee\_characteristics[1] == "Property")),

length(which(DV1\_attendee\_characteristics[1] == "Operational Research")),

length(which(DV1\_attendee\_characteristics[1] == "Social Research")),

length(which(DV1\_attendee\_characteristics[1] == "Statistics")),

length(which(DV1\_attendee\_characteristics[1] == "Legal")),

length(which(DV1\_attendee\_characteristics[1] == "Science and Engineering")),

length(which(DV1\_attendee\_characteristics[1] == "Veterinary")),

length(which(DV1\_attendee\_characteristics[1] == "Other")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) +

geom\_bar(stat = "identity", fill = "navy") + coord\_flip() + xlab("Profession") +

ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# ALB Analysis

output$DV1\_ALB <- renderPlot({

x = c("Animal and Plant Health Agency",

"Centre for Environment, Fisheries, and Aquaculture Science",

"Environment Agency", "Forestry Commission", "Marine Management Organisation",

"Natural England", "Rural Payments Agency", "Veterinary Medicines Directorate",

"Other", "N/A")

y = c(length(which(DV1\_attendee\_characteristics[4] == "Animal and Plant Health Agenc")),

length(which(DV1\_attendee\_characteristics[4] == "Centre for Environment, Fisheries, and Aquaculture Science")),

length(which(DV1\_attendee\_characteristics[4] == "Environment Agency")),

length(which(DV1\_attendee\_characteristics[4] == "Forestry Commission")),

length(which(DV1\_attendee\_characteristics[4] == "Marine Management Organisation")),

length(which(DV1\_attendee\_characteristics[4] == "Natural England")),

length(which(DV1\_attendee\_characteristics[4] == "Rural Payments Agency")),

length(which(DV1\_attendee\_characteristics[4] == "Veterinary Medicines Directorate")),

length(which(DV1\_attendee\_characteristics[4] == "Other")),

length(which(DV1\_attendee\_characteristics[4] == "N/A")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) + geom\_bar(stat = "identity", fill = "navy") + coord\_flip() +

xlab("ALB") + ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Feedback polls--------------------------------------------------------------

# Number of responses = number of rows

DV1\_num\_responses <- function(){nrow(DV1\_feedback)}

output$DV1\_num\_responses <- renderValueBox({

valueBox(paste0(DV1\_num\_responses()), "Number of responses", color = "olive")})

# Feedback response rate

DV1\_feedback\_response\_rate <- function(){

round(((nrow(DV1\_feedback)) / as.numeric(DV1\_summary\_statistics[4])) \* 100, digits = 0)}

output$DV1\_feedback\_response\_rate <- renderValueBox({

valueBox(paste0(DV1\_feedback\_response\_rate(), "%"), "Feedback response rate", color = RAG(DV1\_feedback\_response\_rate() / 100))})

# QUESTION: Did you find this session useful?

output$DV1\_useful\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(DV1\_feedback[6] == "Strongly Disagree")) + length(which(DV1\_feedback[6] == "Strongly disagree")),

length(which(DV1\_feedback[6] == "Disagree")),

length(which(DV1\_feedback[6] == "Neutral")),

length(which(DV1\_feedback[6] == "Agree")),

length(which(DV1\_feedback[6] == "Strongly Agree")) + length(which(DV1\_feedback[6] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Did you find this session engaging?

output$DV1\_engaging\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(DV1\_feedback[7] == "Strongly Disagree")) + length(which(DV1\_feedback[7] == "Strongly disagree")),

length(which(DV1\_feedback[7] == "Disagree")),

length(which(DV1\_feedback[7] == "Neutral")),

length(which(DV1\_feedback[7] == "Agree")),

length(which(DV1\_feedback[7] == "Strongly Agree")) + length(which(DV1\_feedback[7] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Did you learn something from this session?

output$DV1\_learning\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(DV1\_feedback[8] == "Strongly Disagree")) + length(which(DV1\_feedback[8] == "Strongly disagree")),

length(which(DV1\_feedback[8] == "Disagree")),

length(which(DV1\_feedback[8] == "Neutral")),

length(which(DV1\_feedback[8] == "Agree")),

length(which(DV1\_feedback[8] == "Strongly Agree")) + length(which(DV1\_feedback[8] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Would you recommend this session to a colleague

output$DV1\_recommend\_plot <- renderPlot({

# Create x vector to five survey response options

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(DV1\_feedback[9] == "Strongly Disagree")) + length(which(DV1\_feedback[9] == "Strongly disagree")),

length(which(DV1\_feedback[9] == "Disagree")),

length(which(DV1\_feedback[9] == "Neutral")),

length(which(DV1\_feedback[9] == "Agree")),

length(which(DV1\_feedback[9] == "Strongly Agree")) + length(which(DV1\_feedback[9] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Summary Statistics----------------------------------------------------------

# Attendees as a proportion of those who registered

DV1\_prop\_attend <- function(){

round((as.numeric(DV1\_summary\_statistics[4]) /

as.numeric(DV1\_summary\_statistics[2])) \* 100, digits = 0)}

output$DV1\_prop\_attend\_box <- renderValueBox({

valueBox(paste0(DV1\_prop\_attend(), "%"),

"Attendees as a proportion of those who registered", color = RAG(DV1\_prop\_attend() / 100))})

# Number of attendees

DV1\_num\_attendees <- function(){DV1\_summary\_statistics[4]}

output$DV1\_num\_attendees\_box <- renderValueBox({

valueBox(paste0(DV1\_num\_attendees()), "Number of attendees", color = "light-blue")})

# Number of registrations

DV1\_num\_registrations <- function(){DV1\_summary\_statistics[2]}

output$DV1\_num\_registrations\_box <- renderValueBox({

valueBox(paste0(DV1\_num\_registrations()), "Number of registrations", color = "light-blue")})

# Proportion of people who viewed registration page that then signed up, (num registered + num cancelled) / num views

DV1\_prop\_register <- function(){

round(((as.numeric(DV1\_summary\_statistics[2]) +

as.numeric(DV1\_summary\_statistics[3])) /

as.numeric(DV1\_summary\_statistics[1])) \* 100, digits = 0)}

output$DV1\_prop\_register\_box <- renderValueBox({

valueBox(paste0(DV1\_prop\_register(), "%"),

"Proportion of people who viewed registration page that then signed up",

color = RAG(DV1\_prop\_register() / 100))})

# Number of page views

DV1\_num\_page\_views <- function(){DV1\_summary\_statistics[1]}

output$DV1\_num\_page\_views\_box <- renderValueBox({

valueBox(paste0(DV1\_num\_page\_views()), "Number of page views", color = "light-blue")})

# Number of cancellations

DV1\_num\_cancel <- function(){as.numeric(DV1\_summary\_statistics[3])}

output$DV1\_num\_cancel\_box <- renderValueBox({

valueBox(paste0(DV1\_num\_cancel()), "Number of cancellations", color = "light-blue")})

# Average attendance time

DV1\_avg\_attend\_time <- function(){DV1\_summary\_statistics[6]}

output$DV1\_avg\_attend\_time\_box <- renderValueBox({

valueBox(paste0(DV1\_avg\_attend\_time()), "Average attendance duration", color = "light-blue")})

# Training duration

DV1\_train\_duration <- function(){"1h 00m 00s"}

output$DV1\_train\_duration\_box <- renderValueBox({

valueBox(paste0(DV1\_train\_duration()), "Target training duration", color = "light-blue")})

# Meeting duration

DV1\_meet\_duration <- function(){DV1\_summary\_statistics[5]}

output$DV1\_meet\_duration\_box <- renderValueBox({

valueBox(paste0(DV1\_meet\_duration()), "Actual training duration", color = "light-blue")})

# INTRODUCTION TO DATA VISUALISATIONS : PART 2================================

# Attendee Characteristics----------------------------------------------------

# Grade analysis

output$DV2\_grade <- renderPlot({

x = c("AO/AA", "EO", "HEO", "SEO", "Grade 7", "Grade 6", "SCS")

y = c(length(which(DV2\_attendee\_characteristics[2] == "AO/AA")),

length(which(DV2\_attendee\_characteristics[2] == "EO")),

length(which(DV2\_attendee\_characteristics[2] == "HEO")),

length(which(DV2\_attendee\_characteristics[2] == "SEO")),

length(which(DV2\_attendee\_characteristics[2] == "Grade 7")),

length(which(DV2\_attendee\_characteristics[2] == "Grade 6")),

length(which(DV2\_attendee\_characteristics[2] == "SCS")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

xlab("Grade") + ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# DG group analysis

output$DV2\_DG\_group <- renderPlot({

x = c("Environment", "Food, Biosecurity, and Trade", "Portfolio Delivery",

"Strategy", "Science and Analysis")

y = c(length(which(DV2\_attendee\_characteristics[3] == "Environment")),

length(which(DV2\_attendee\_characteristics[3] == "Food, Biosecurity, and Trade")),

length(which(DV2\_attendee\_characteristics[3] == "Portfolio Delivery")),

length(which(DV2\_attendee\_characteristics[3] == "Strategy")),

length(which(DV2\_attendee\_characteristics[3] == "Science and Analysis")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) +

geom\_bar(stat = "identity", fill = "navy") + coord\_flip() + xlab("DG Group") +

ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Profession analysis

output$DV2\_profession <- renderPlot({

x = c("Operational Delivery",

"Policy",

"Commercial",

"Communications",

"Digital and Data",

"Economics",

"Finance",

"Geography",

"Human Resources",

"Project Delivery",

"Property",

"Operational Research",

"Social Research",

"Statistics",

"Legal",

"Science and Engineering",

"Veterinary",

"Other")

y = c(length(which(DV2\_attendee\_characteristics[1] == "Operational Delivery")),

length(which(DV2\_attendee\_characteristics[1] == "Policy")),

length(which(DV2\_attendee\_characteristics[1] == "Commercial")),

length(which(DV2\_attendee\_characteristics[1] == "Communications")),

length(which(DV2\_attendee\_characteristics[1] == "Digital and Data")),

length(which(DV2\_attendee\_characteristics[1] == "Economics")),

length(which(DV2\_attendee\_characteristics[1] == "Finance")),

length(which(DV2\_attendee\_characteristics[1] == "Geography")),

length(which(DV2\_attendee\_characteristics[1] == "Human Resources")),

length(which(DV2\_attendee\_characteristics[1] == "Project Delivery")),

length(which(DV2\_attendee\_characteristics[1] == "Property")),

length(which(DV2\_attendee\_characteristics[1] == "Operational Research")),

length(which(DV2\_attendee\_characteristics[1] == "Social Research")),

length(which(DV2\_attendee\_characteristics[1] == "Statistics")),

length(which(DV2\_attendee\_characteristics[1] == "Legal")),

length(which(DV2\_attendee\_characteristics[1] == "Science and Engineering")),

length(which(DV2\_attendee\_characteristics[1] == "Veterinary")),

length(which(DV2\_attendee\_characteristics[1] == "Other")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) +

geom\_bar(stat = "identity", fill = "navy") + coord\_flip() + xlab("Profession") +

ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# ALB Analysis

output$DV2\_ALB <- renderPlot({

x = c("Animal and Plant Health Agency",

"Centre for Environment, Fisheries, and Aquaculture Science",

"Environment Agency", "Forestry Commission", "Marine Management Organisation",

"Natural England", "Rural Payments Agency", "Veterinary Medicines Directorate",

"Other", "N/A")

y = c(length(which(DV2\_attendee\_characteristics[4] == "Animal and Plant Health Agenc")),

length(which(DV2\_attendee\_characteristics[4] == "Centre for Environment, Fisheries, and Aquaculture Science")),

length(which(DV2\_attendee\_characteristics[4] == "Environment Agency")),

length(which(DV2\_attendee\_characteristics[4] == "Forestry Commission")),

length(which(DV2\_attendee\_characteristics[4] == "Marine Management Organisation")),

length(which(DV2\_attendee\_characteristics[4] == "Natural England")),

length(which(DV2\_attendee\_characteristics[4] == "Rural Payments Agency")),

length(which(DV2\_attendee\_characteristics[4] == "Veterinary Medicines Directorate")),

length(which(DV2\_attendee\_characteristics[4] == "Other")),

length(which(DV2\_attendee\_characteristics[4] == "N/A")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) + geom\_bar(stat = "identity", fill = "navy") + coord\_flip() +

xlab("ALB") + ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Feedback polls--------------------------------------------------------------

# Number of responses = number of rows

DV2\_num\_responses <- function(){nrow(DV2\_feedback)}

output$DV2\_num\_responses <- renderValueBox({

valueBox(paste0(DV2\_num\_responses()), "Number of responses", color = "olive")})

# Feedback response rate

DV2\_feedback\_response\_rate <- function(){

round(((nrow(DV2\_feedback)) / as.numeric(DV2\_summary\_statistics[4])) \* 100, digits = 0)}

output$DV2\_feedback\_response\_rate <- renderValueBox({

valueBox(paste0(DV2\_feedback\_response\_rate(), "%"), "Feedback response rate", color = RAG(DV2\_feedback\_response\_rate() / 100))})

# QUESTION: Did you find this session useful?

output$DV2\_useful\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(DV2\_feedback[6] == "Strongly Disagree")) + length(which(DV2\_feedback[6] == "Strongly disagree")),

length(which(DV2\_feedback[6] == "Disagree")),

length(which(DV2\_feedback[6] == "Neutral")),

length(which(DV2\_feedback[6] == "Agree")),

length(which(DV2\_feedback[6] == "Strongly Agree")) + length(which(DV2\_feedback[6] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Did you find this session engaging?

output$DV2\_engaging\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(DV2\_feedback[7] == "Strongly Disagree")) + length(which(DV2\_feedback[7] == "Strongly disagree")),

length(which(DV2\_feedback[7] == "Disagree")),

length(which(DV2\_feedback[7] == "Neutral")),

length(which(DV2\_feedback[7] == "Agree")),

length(which(DV2\_feedback[7] == "Strongly Agree")) + length(which(DV2\_feedback[7] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Did you learn something from this session?

output$DV2\_learning\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(DV2\_feedback[8] == "Strongly Disagree")) + length(which(DV2\_feedback[8] == "Strongly disagree")),

length(which(DV2\_feedback[8] == "Disagree")),

length(which(DV2\_feedback[8] == "Neutral")),

length(which(DV2\_feedback[8] == "Agree")),

length(which(DV2\_feedback[8] == "Strongly Agree")) + length(which(DV2\_feedback[8] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Would you recommend this session to a colleague

output$DV2\_recommend\_plot <- renderPlot({

# Create x vector to five survey response options

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(DV2\_feedback[9] == "Strongly Disagree")) + length(which(DV2\_feedback[9] == "Strongly disagree")),

length(which(DV2\_feedback[9] == "Disagree")),

length(which(DV2\_feedback[9] == "Neutral")),

length(which(DV2\_feedback[9] == "Agree")),

length(which(DV2\_feedback[9] == "Strongly Agree")) + length(which(DV2\_feedback[9] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Summary Statistics----------------------------------------------------------

# Attendees as a proportion of those who registered

DV2\_prop\_attend <- function(){

round((as.numeric(DV2\_summary\_statistics[4]) /

as.numeric(DV2\_summary\_statistics[2])) \* 100, digits = 0)}

output$DV2\_prop\_attend\_box <- renderValueBox({

valueBox(paste0(DV2\_prop\_attend(), "%"),

"Attendees as a proportion of those who registered", color = RAG(DV2\_prop\_attend() / 100))})

# Number of attendees

DV2\_num\_attendees <- function(){DV2\_summary\_statistics[4]}

output$DV2\_num\_attendees\_box <- renderValueBox({

valueBox(paste0(DV2\_num\_attendees()), "Number of attendees", color = "light-blue")})

# Number of registrations

DV2\_num\_registrations <- function(){DV2\_summary\_statistics[2]}

output$DV2\_num\_registrations\_box <- renderValueBox({

valueBox(paste0(DV2\_num\_registrations()), "Number of registrations", color = "light-blue")})

# Proportion of people who viewed registration page that then signed up, (num registered + num cancelled) / num views

DV2\_prop\_register <- function(){

round(((as.numeric(DV2\_summary\_statistics[2]) +

as.numeric(DV2\_summary\_statistics[3])) /

as.numeric(DV2\_summary\_statistics[1])) \* 100, digits = 0)}

output$DV2\_prop\_register\_box <- renderValueBox({

valueBox(paste0(DV2\_prop\_register(), "%"),

"Proportion of people who viewed registration page that then signed up",

color = RAG(DV2\_prop\_register() / 100))})

# Number of page views

DV2\_num\_page\_views <- function(){DV2\_summary\_statistics[1]}

output$DV2\_num\_page\_views\_box <- renderValueBox({

valueBox(paste0(DV2\_num\_page\_views()), "Number of page views", color = "light-blue")})

# Number of cancellations

DV2\_num\_cancel <- function(){as.numeric(DV2\_summary\_statistics[3])}

output$DV2\_num\_cancel\_box <- renderValueBox({

valueBox(paste0(DV2\_num\_cancel()), "Number of cancellations", color = "light-blue")})

# Average attendance time

DV2\_avg\_attend\_time <- function(){DV2\_summary\_statistics[6]}

output$DV2\_avg\_attend\_time\_box <- renderValueBox({

valueBox(paste0(DV2\_avg\_attend\_time()), "Average attendance duration", color = "light-blue")})

# Training duration

DV2\_train\_duration <- function(){"1h 00m 00s"}

output$DV2\_train\_duration\_box <- renderValueBox({

valueBox(paste0(DV2\_train\_duration()), "Target training duration", color = "light-blue")})

# Meeting duration

DV2\_meet\_duration <- function(){DV2\_summary\_statistics[5]}

output$DV2\_meet\_duration\_box <- renderValueBox({

valueBox(paste0(DV2\_meet\_duration()), "Actual training duration", color = "light-blue")})

# QUIZ! UNDERSTANDING ANALYTICAL GUIDANCE=====================================

# Attendee Characteristics----------------------------------------------------

# Grade analysis

output$QUIZ\_grade <- renderPlot({

x = c("AO/AA", "EO", "HEO", "SEO", "Grade 7", "Grade 6", "SCS")

y = c(length(which(QUIZ\_attendee\_characteristics[2] == "AO/AA")),

length(which(QUIZ\_attendee\_characteristics[2] == "EO")),

length(which(QUIZ\_attendee\_characteristics[2] == "HEO")),

length(which(QUIZ\_attendee\_characteristics[2] == "SEO")),

length(which(QUIZ\_attendee\_characteristics[2] == "Grade 7")),

length(which(QUIZ\_attendee\_characteristics[2] == "Grade 6")),

length(which(QUIZ\_attendee\_characteristics[2] == "SCS")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

xlab("Grade") + ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# DG group analysis

output$QUIZ\_DG\_group <- renderPlot({

x = c("Environment", "Food, Biosecurity, and Trade", "Portfolio Delivery",

"Strategy", "Science and Analysis")

y = c(length(which(QUIZ\_attendee\_characteristics[3] == "Environment")),

length(which(QUIZ\_attendee\_characteristics[3] == "Food, Biosecurity, and Trade")),

length(which(QUIZ\_attendee\_characteristics[3] == "Portfolio Delivery")),

length(which(QUIZ\_attendee\_characteristics[3] == "Strategy")),

length(which(QUIZ\_attendee\_characteristics[3] == "Science and Analysis")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) +

geom\_bar(stat = "identity", fill = "navy") + coord\_flip() + xlab("DG Group") +

ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Profession analysis

output$QUIZ\_profession <- renderPlot({

x = c("Operational Delivery",

"Policy",

"Commercial",

"Communications",

"Digital and Data",

"Economics",

"Finance",

"Geography",

"Human Resources",

"Project Delivery",

"Property",

"Operational Research",

"Social Research",

"Statistics",

"Legal",

"Science and Engineering",

"Veterinary",

"Other")

y = c(length(which(QUIZ\_attendee\_characteristics[1] == "Operational Delivery")),

length(which(QUIZ\_attendee\_characteristics[1] == "Policy")),

length(which(QUIZ\_attendee\_characteristics[1] == "Commercial")),

length(which(QUIZ\_attendee\_characteristics[1] == "Communications")),

length(which(QUIZ\_attendee\_characteristics[1] == "Digital and Data")),

length(which(QUIZ\_attendee\_characteristics[1] == "Economics")),

length(which(QUIZ\_attendee\_characteristics[1] == "Finance")),

length(which(QUIZ\_attendee\_characteristics[1] == "Geography")),

length(which(QUIZ\_attendee\_characteristics[1] == "Human Resources")),

length(which(QUIZ\_attendee\_characteristics[1] == "Project Delivery")),

length(which(QUIZ\_attendee\_characteristics[1] == "Property")),

length(which(QUIZ\_attendee\_characteristics[1] == "Operational Research")),

length(which(QUIZ\_attendee\_characteristics[1] == "Social Research")),

length(which(QUIZ\_attendee\_characteristics[1] == "Statistics")),

length(which(QUIZ\_attendee\_characteristics[1] == "Legal")),

length(which(QUIZ\_attendee\_characteristics[1] == "Science and Engineering")),

length(which(QUIZ\_attendee\_characteristics[1] == "Veterinary")),

length(which(QUIZ\_attendee\_characteristics[1] == "Other")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) +

geom\_bar(stat = "identity", fill = "navy") + coord\_flip() + xlab("Profession") +

ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# ALB Analysis

output$QUIZ\_ALB <- renderPlot({

x = c("Animal and Plant Health Agency",

"Centre for Environment, Fisheries, and Aquaculture Science",

"Environment Agency", "Forestry Commission", "Marine Management Organisation",

"Natural England", "Rural Payments Agency", "Veterinary Medicines Directorate",

"Other", "N/A")

y = c(length(which(QUIZ\_attendee\_characteristics[4] == "Animal and Plant Health Agenc")),

length(which(QUIZ\_attendee\_characteristics[4] == "Centre for Environment, Fisheries, and Aquaculture Science")),

length(which(QUIZ\_attendee\_characteristics[4] == "Environment Agency")),

length(which(QUIZ\_attendee\_characteristics[4] == "Forestry Commission")),

length(which(QUIZ\_attendee\_characteristics[4] == "Marine Management Organisation")),

length(which(QUIZ\_attendee\_characteristics[4] == "Natural England")),

length(which(QUIZ\_attendee\_characteristics[4] == "Rural Payments Agency")),

length(which(QUIZ\_attendee\_characteristics[4] == "Veterinary Medicines Directorate")),

length(which(QUIZ\_attendee\_characteristics[4] == "Other")),

length(which(QUIZ\_attendee\_characteristics[4] == "N/A")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = reorder(fct\_inorder(x, ordered = NA), y), y = y)) + geom\_bar(stat = "identity", fill = "navy") + coord\_flip() +

xlab("ALB") + ylab("Number in attendance") + theme(axis.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Feedback polls--------------------------------------------------------------

# Number of responses = number of rows

QUIZ\_num\_responses <- function(){nrow(QUIZ\_feedback)}

output$QUIZ\_num\_responses <- renderValueBox({

valueBox(paste0(QUIZ\_num\_responses()), "Number of responses", color = "olive")})

# Feedback response rate

QUIZ\_feedback\_response\_rate <- function(){

round(((nrow(QUIZ\_feedback)) / as.numeric(QUIZ\_summary\_statistics[4])) \* 100, digits = 0)}

output$QUIZ\_feedback\_response\_rate <- renderValueBox({

valueBox(paste0(QUIZ\_feedback\_response\_rate(), "%"), "Feedback response rate", color = RAG(QUIZ\_feedback\_response\_rate() / 100))})

# QUESTION: Did you find this session useful?

output$QUIZ\_useful\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(QUIZ\_feedback[6] == "Strongly Disagree")) + length(which(QUIZ\_feedback[6] == "Strongly disagree")),

length(which(QUIZ\_feedback[6] == "Disagree")),

length(which(QUIZ\_feedback[6] == "Neutral")),

length(which(QUIZ\_feedback[6] == "Agree")),

length(which(QUIZ\_feedback[6] == "Strongly Agree")) + length(which(QUIZ\_feedback[6] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Did you find this session engaging?

output$QUIZ\_engaging\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(QUIZ\_feedback[7] == "Strongly Disagree")) + length(which(QUIZ\_feedback[7] == "Strongly disagree")),

length(which(QUIZ\_feedback[7] == "Disagree")),

length(which(QUIZ\_feedback[7] == "Neutral")),

length(which(QUIZ\_feedback[7] == "Agree")),

length(which(QUIZ\_feedback[7] == "Strongly Agree")) + length(which(QUIZ\_feedback[7] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Did you learn something from this session?

output$QUIZ\_learning\_plot <- renderPlot({

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(QUIZ\_feedback[8] == "Strongly Disagree")) + length(which(QUIZ\_feedback[8] == "Strongly disagree")),

length(which(QUIZ\_feedback[8] == "Disagree")),

length(which(QUIZ\_feedback[8] == "Neutral")),

length(which(QUIZ\_feedback[8] == "Agree")),

length(which(QUIZ\_feedback[8] == "Strongly Agree")) + length(which(QUIZ\_feedback[8] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

#QUESTION: Would you recommend this session to a colleague

output$QUIZ\_recommend\_plot <- renderPlot({

# Create x vector to five survey response options

# Create x vector to five survey response options

x = c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree")

# Create y vector that counts the number of each response type

y = c(length(which(QUIZ\_feedback[9] == "Strongly Disagree")) + length(which(QUIZ\_feedback[9] == "Strongly disagree")),

length(which(QUIZ\_feedback[9] == "Disagree")),

length(which(QUIZ\_feedback[9] == "Neutral")),

length(which(QUIZ\_feedback[9] == "Agree")),

length(which(QUIZ\_feedback[9] == "Strongly Agree")) + length(which(QUIZ\_feedback[9] == "Strongly agree")))

# Create data frame

df <- data.frame(x, y)

# Column plot results

ggplot(df, aes(x = fct\_inorder(x, ordered = NA), y = y)) + geom\_col(fill = "navy") +

ggtitle(paste("[", sum(y), " responses]")) +

geom\_text(aes(label = y), vjust = -0.2) +

ylab("Number of response") + theme(axis.title.x = element\_blank(),

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

axis.title = element\_text(size = 14, face = "bold"),

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

})

# Summary Statistics----------------------------------------------------------

# Attendees as a proportion of those who registered

QUIZ\_prop\_attend <- function(){

round((as.numeric(QUIZ\_summary\_statistics[4]) /

as.numeric(QUIZ\_summary\_statistics[2])) \* 100, digits = 0)}

output$QUIZ\_prop\_attend\_box <- renderValueBox({

valueBox(paste0(QUIZ\_prop\_attend(), "%"),

"Attendees as a proportion of those who registered", color = RAG(QUIZ\_prop\_attend() / 100))})

# Number of attendees

QUIZ\_num\_attendees <- function(){QUIZ\_summary\_statistics[4]}

output$QUIZ\_num\_attendees\_box <- renderValueBox({

valueBox(paste0(QUIZ\_num\_attendees()), "Number of attendees", color = "light-blue")})

# Number of registrations

QUIZ\_num\_registrations <- function(){QUIZ\_summary\_statistics[2]}

output$QUIZ\_num\_registrations\_box <- renderValueBox({

valueBox(paste0(QUIZ\_num\_registrations()), "Number of registrations", color = "light-blue")})

# Proportion of people who viewed registration page that then signed up, (num registered + num cancelled) / num views

QUIZ\_prop\_register <- function(){

round(((as.numeric(QUIZ\_summary\_statistics[2]) +

as.numeric(QUIZ\_summary\_statistics[3])) /

as.numeric(QUIZ\_summary\_statistics[1])) \* 100, digits = 0)}

output$QUIZ\_prop\_register\_box <- renderValueBox({

valueBox(paste0(QUIZ\_prop\_register(), "%"),

"Proportion of people who viewed registration page that then signed up",

color = RAG(QUIZ\_prop\_register() / 100))})

# Number of page views

QUIZ\_num\_page\_views <- function(){QUIZ\_summary\_statistics[1]}

output$QUIZ\_num\_page\_views\_box <- renderValueBox({

valueBox(paste0(QUIZ\_num\_page\_views()), "Number of page views", color = "light-blue")})

# Number of cancellations

QUIZ\_num\_cancel <- function(){as.numeric(QUIZ\_summary\_statistics[3])}

output$QUIZ\_num\_cancel\_box <- renderValueBox({

valueBox(paste0(QUIZ\_num\_cancel()), "Number of cancellations", color = "light-blue")})

# Average attendance time

QUIZ\_avg\_attend\_time <- function(){QUIZ\_summary\_statistics[6]}

output$QUIZ\_avg\_attend\_time\_box <- renderValueBox({

valueBox(paste0(QUIZ\_avg\_attend\_time()), "Average attendance duration", color = "light-blue")})

# Training duration

QUIZ\_train\_duration <- function(){"1h 00m 00s"}

output$QUIZ\_train\_duration\_box <- renderValueBox({

valueBox(paste0(QUIZ\_train\_duration()), "Target training duration", color = "light-blue")})

# Meeting duration

QUIZ\_meet\_duration <- function(){QUIZ\_summary\_statistics[5]}

output$QUIZ\_meet\_duration\_box <- renderValueBox({

valueBox(paste0(QUIZ\_meet\_duration()), "Actual training duration", color = "light-blue")})

} # end of server bracket

# Run the application

shinyApp(ui = ui, server = server)