An Introduction to R Shiny

an interactive web-based application for data visualization, analysis and more

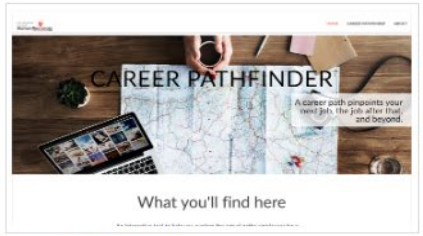
Zhanglong Cao, SAGI West

What is R Shiny?

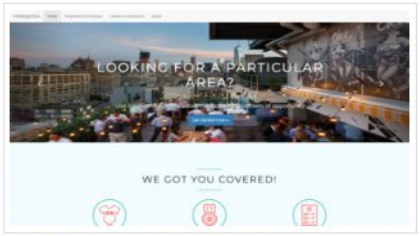
- Shiny is an R package that makes it easy to build interactive web apps straight from R.

Shiny User Showcase https://shiny.rstudio.com/gallery/

Finance / Banking



Career PathFinder



Identifying real estate investment opportunities

Government / Public sector

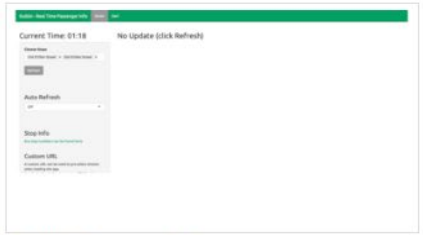
Mostly open data

Freedom of Press Index

Voronoys - Understanding voters' profile in Brazilian elections

Crime Watch

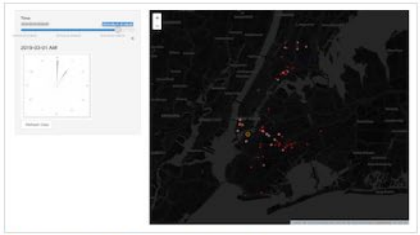
Pasture Potential Tool for improving dairy farm profitability and environmental impact



Dublin Transport Info

Locating Blood Banks in India

Utah Lake Water Quality Profile Dashboard



Animated NYC metro traffic

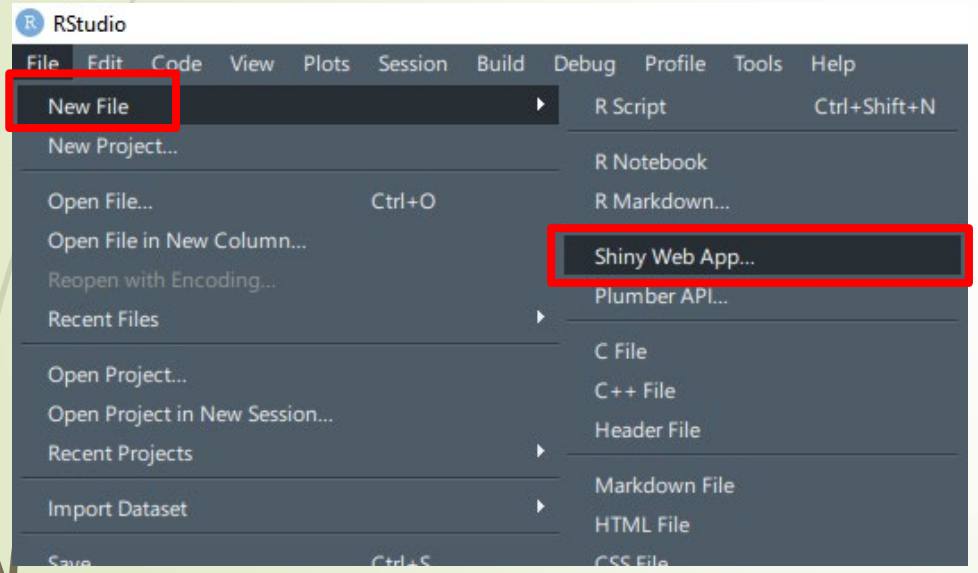
GRDC SAGI Curtin University

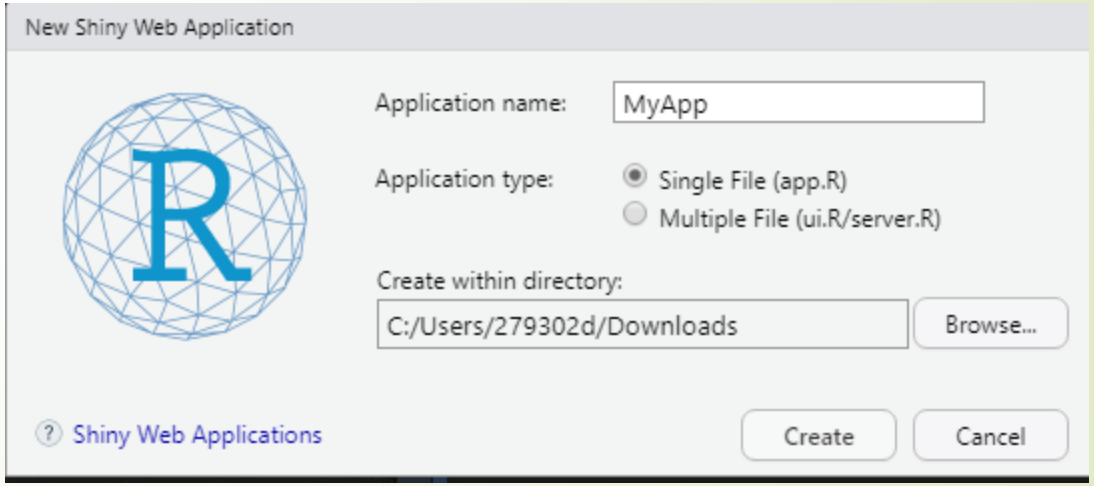
Why R Shiny?

- It's a powerful tool for building interactive for data visualization and analysis.
- It's in the R ecosystem.
- It's easy to deploy and host by R Studio.
- It's easy to customise to meet various of demands.
- No knowledge of HTML, CSS or JavaScript is required.

How to build a R Shiny App?

- install.package("shiny")
- Create a new Shiny Web App

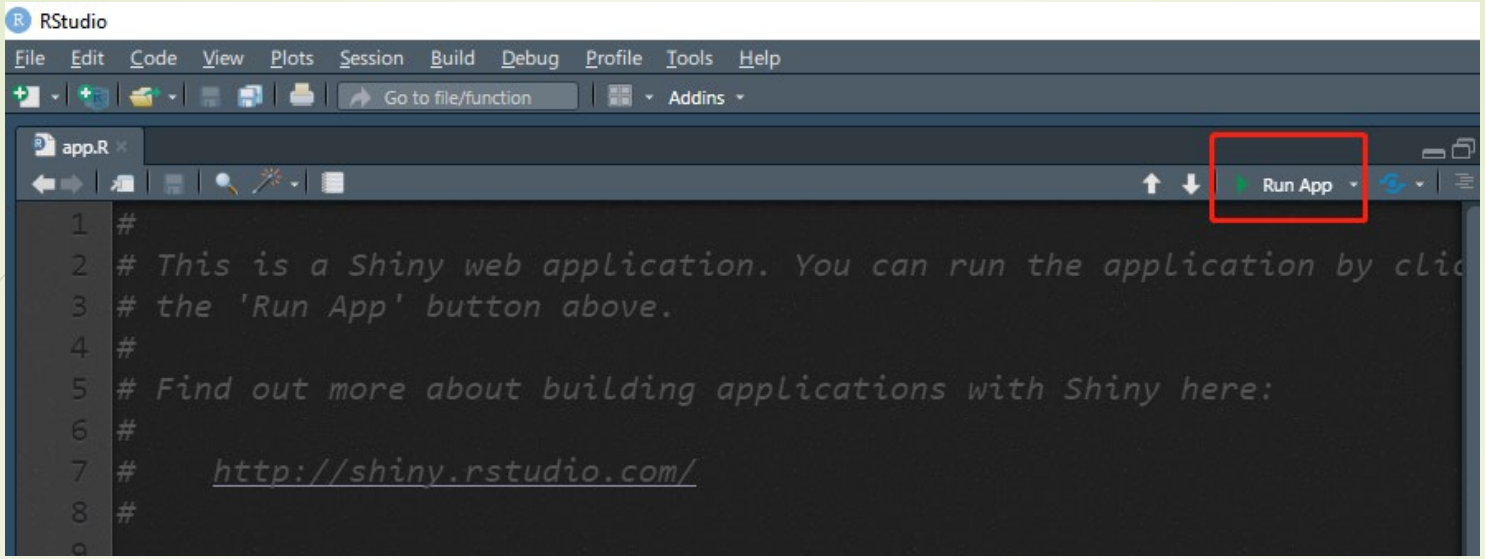




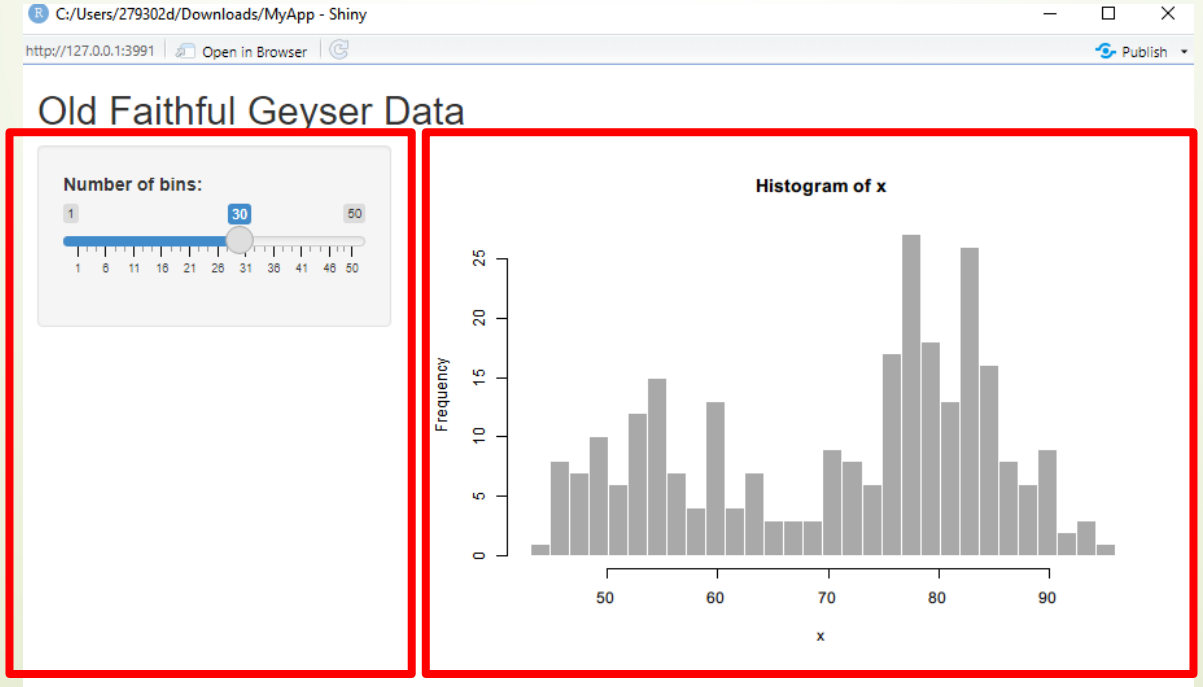
```
# Define UI for application that draws a histogram
ui <- fluidPage(

    # Application title
    titlePanel("Old Faithful Geyser Data"),

    # Sidebar with a slider input for number of bins
    sidebarLayout(
        sidebarPanel(
            sliderInput("bins",
                        "Number of bins:",
                        min = 1,
                        max = 50,
                        value = 30)
        ),

        # Show a plot of the generated distribution
        mainPanel(
           plotOutput("distPlot")
        )
    )
)
```

```
# Define server logic required to draw a histogram
server <- function(input, output) {

    output$distPlot <- renderPlot({
        # generate bins based on input$bins from ui.R
        x    <- faithful[, 2]
        bins <- seq(min(x), max(x), length.out = input$bins + 1)

        # draw the histogram with the specified number of bins
        hist(x, breaks = bins, col = 'darkgray', border = 'white')
    })
}
```

```
# Run the application
shinyApp(ui = ui, server = server)
```



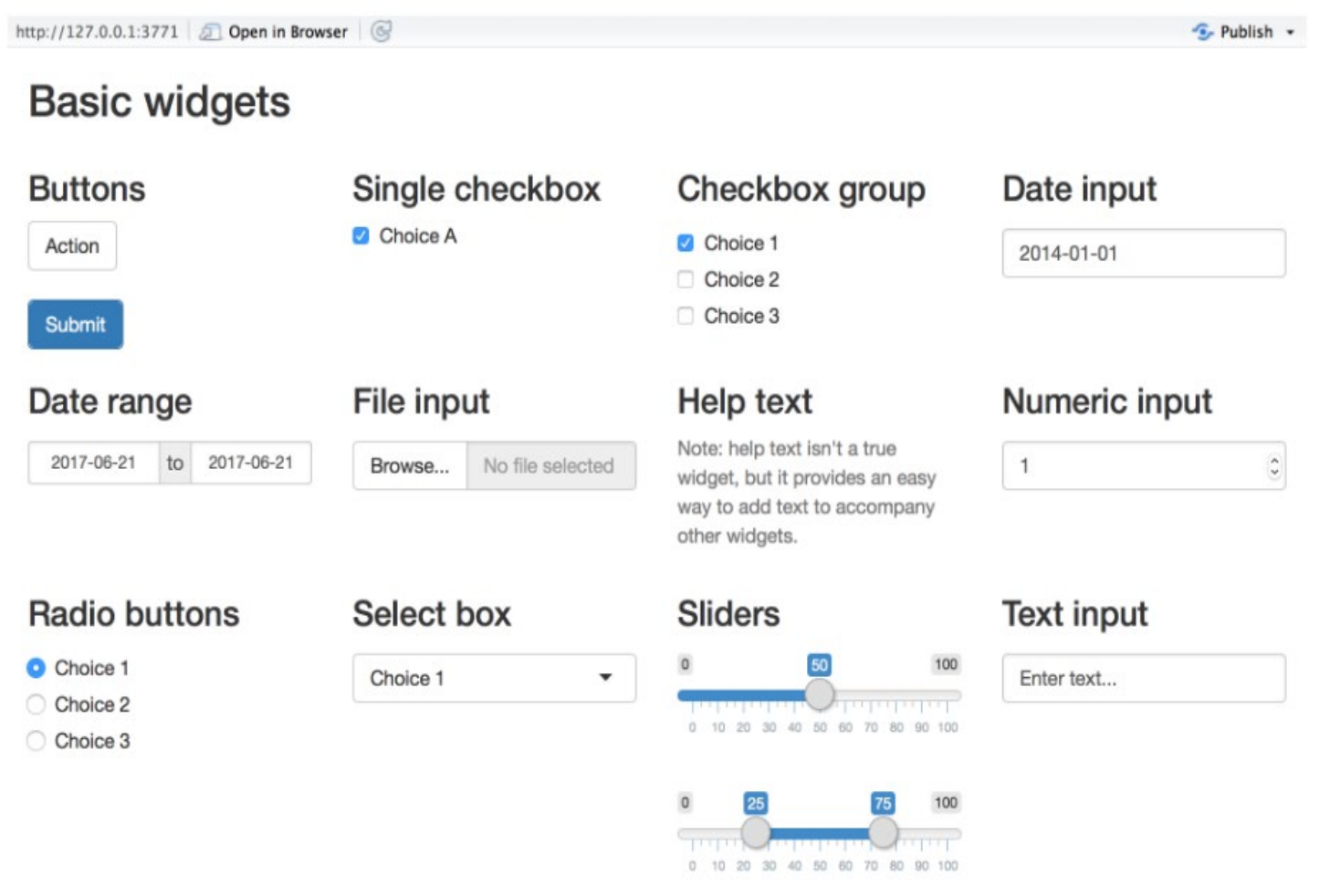
Sidebar panel



Main panel

GRDC SAGI Curtin University

Add more control widgets



- Button (click, radio)
- Checkbox (single, multiple choice)
- Input box (numeric, text)
- Slider
- Select box (drop down menu)

Give it a try

```
# Sidebar with a slider input for number of bins
sidebarLayout(
    sidebarPanel(
        sliderInput("bins",
                    "Number of bins:",
                    min = 1,
                    max = 50,
                    value = 30),

        numericInput("Rand",
                     "Number of observations:",
                     value = 100,
                     min = 1,
                     max = 1000
        )

    ),

    # Show a plot of the generated distribution
    mainPanel(
       plotOutput("distPlot"),
       plotOutput("RandPlot")
    )
)
```

```
# Define server logic required to draw a histogram
server <- function(input, output) {

    output$distPlot <- renderPlot({
        # generate bins based on input$bins from ui.R
        x    <- faithful[, 2]
        bins <- seq(min(x), max(x), length.out = inpu

        # draw the histogram with the specified numbe
        hist(x, breaks = bins, col = 'darkgray', bord
    })

    output$RandPlot <- renderPlot({
      x <- rnorm(input$Rand)
      plot(x)
    })
}
```

Old Faithful Geyser Data

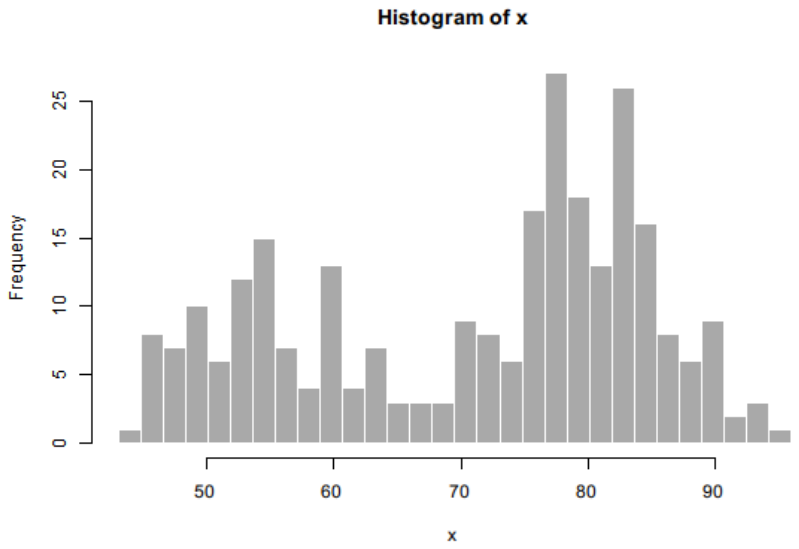
Number of bins:

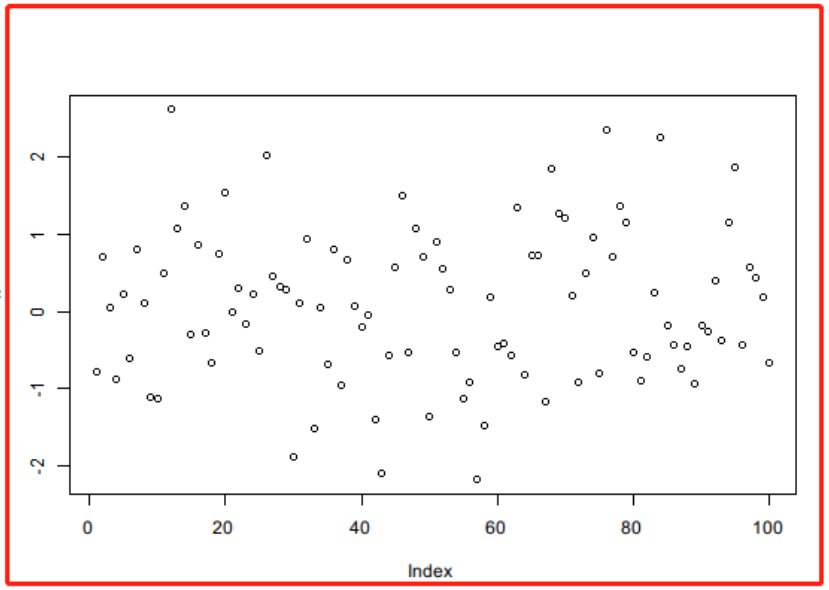
1 30 50

1 6 11 16 21 26 31 36 41 46 50

Number of observations:

100

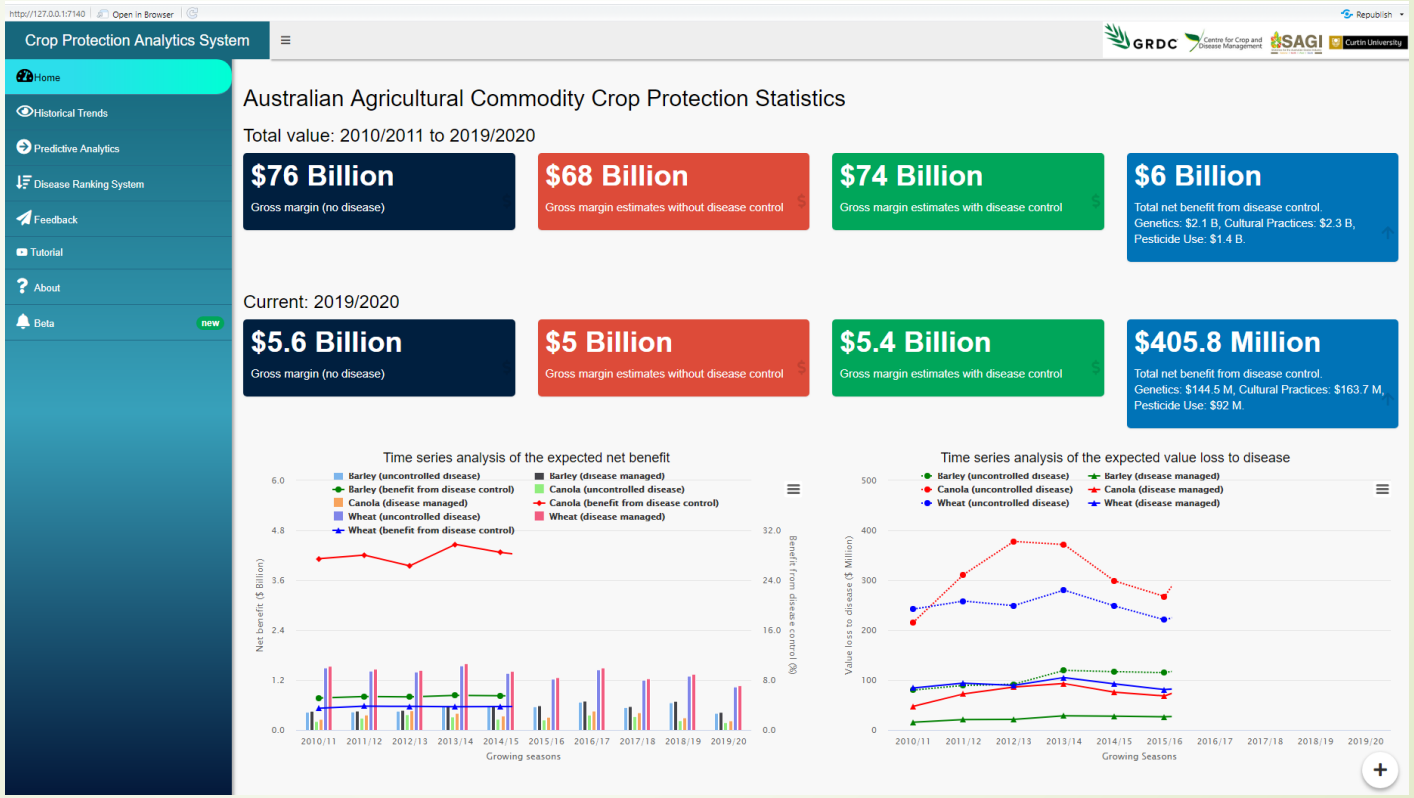




GRDC SAGI Curtin University

The Crop Protection Analytics System (CPAS)

https://www.ccdm.com.au/cpas/



Thank you!


GRDC
SAGI
Statistics for the Australian Grains Industry
Curtin University