

STATS 780 Assignment 1 (Supplementary Material)

Anna Ly (400254496)

2024-09-24

Supplementary Material

Code for Data Transformation and Preprocessing Steps

```
library(terra)
library(tidyverse)

# SASKatchewan Data
SASK_2009 = "MapData/SASKatchewan/aci_2009_sk_v1.tif"
SASK_2010 = "MapData/SASKatchewan/aci_2010_sk_v1.tif"
SASK_2011 = "MapData/SASKatchewan/aci_2011_sk_v3.tif"
SASK_2012 = "MapData/SASKatchewan/aci_2012_sk_v3.tif"
SASK_2013 = "MapData/SASKatchewan/aci_2013_sk_v3.tif"

# MANItoba Data
MANI_2009 = "MapData/MANItoba/aci_2009_mb_v1.tif"
MANI_2010 = "MapData/MANItoba/aci_2010_mb_v1.tif"
MANI_2011 = "MapData/MANItoba/aci_2011_mb_v3.tif"
MANI_2012 = "MapData/MANItoba/aci_2012_mb_v3.tif"
MANI_2013 = "MapData/MANItoba/aci_2013_mb_v3.tif"

SRCS_SASK_2009 = rast(SASK_2009)
SRCS_SASK_2009 = values(SRCS_SASK_2009[[1]])
SRCS_SASK_2010 = rast(SASK_2010)
SRCS_SASK_2010 = values(SRCS_SASK_2010[[1]])
SRCS_SASK_2011 = rast(SASK_2011)
SRCS_SASK_2011 = values(SRCS_SASK_2011[[1]])
SRCS_SASK_2012 = rast(SASK_2012)
SRCS_SASK_2012 = values(SRCS_SASK_2012[[1]])
SRCS_SASK_2013 = rast(SASK_2013)
SRCS_SASK_2013 = values(SRCS_SASK_2013[[1]])
SRCS_MANI_2009 = rast(MANI_2009)
SRCS_MANI_2009 = values(SRCS_MANI_2009[[1]])
SRCS_MANI_2010 = rast(MANI_2010)
SRCS_MANI_2010 = values(SRCS_MANI_2010[[1]])
SRCS_MANI_2011 = rast(MANI_2011)
SRCS_MANI_2011 = values(SRCS_MANI_2011[[1]])
SRCS_MANI_2012 = rast(MANI_2012)
SRCS_MANI_2012 = values(SRCS_MANI_2012[[1]])
```

```

SRCS_MANI_2013 = rast(MANI_2013)
SRCS_MANI_2013 = values(SRCS_MANI_2013[[1]])

# Extract Codes (these are associated with the crops used in the website)
# Data that contains the code & meanings
COLOUR_MAP = read.csv("aci_crop_classifications.csv", header = TRUE, fileEncoding =
  ↪ "Latin1")

# reference of the dataset we want to make
VALUES_DATASET = data.frame(
  Year = rep(c(2009, 2010, 2011, 2012, 2013), 2),
  Region = rep(c("Saskatchewan", "Manitoba"), each = 5) # Region labels
)

add_crop_data = function(dataset, crop_id = "Peas"){
  code_id = COLOUR_MAP$Code[which(COLOUR_MAP$Label == crop_id)]
  code_id = as.numeric(code_id) # ensuring we have an numeric val
  # Values for Saskatchewan
  extra_sask_2009 = sum(SRCS_SASK_2009 == code_id, na.rm = TRUE)
  extra_sask_2010 = sum(SRCS_SASK_2010 == code_id, na.rm = TRUE)
  extra_sask_2011 = sum(SRCS_SASK_2011 == code_id, na.rm = TRUE)
  extra_sask_2012 = sum(SRCS_SASK_2012 == code_id, na.rm = TRUE)
  extra_sask_2013 = sum(SRCS_SASK_2013 == code_id, na.rm = TRUE)
  # Values for Manitoba
  extra_mani_2009 = sum(SRCS_MANI_2009 == code_id, na.rm = TRUE)
  extra_mani_2010 = sum(SRCS_MANI_2010 == code_id, na.rm = TRUE)
  extra_mani_2011 = sum(SRCS_MANI_2011 == code_id, na.rm = TRUE)
  extra_mani_2012 = sum(SRCS_MANI_2012 == code_id, na.rm = TRUE)
  extra_mani_2013 = sum(SRCS_MANI_2013 == code_id, na.rm = TRUE)
  # Getting the new column name for the dataset
  new_col_name = paste0(crop_id, "_Production")
  # Adding values to the dataset
  dataset = dataset %>%
    mutate(!sym(new_col_name) := c(extra_sask_2009, extra_sask_2010, extra_sask_2011,
                                   extra_sask_2012, extra_sask_2013,
                                   extra_mani_2009, extra_mani_2010, extra_mani_2011,
                                   extra_mani_2012, extra_mani_2013))

  return(dataset)
}

ALL_CROP_TYPES = c("Cereals", "Barley", "Millet", "Oats", "Rye", "Spelt", "Triticale",
  "Wheat", "Sorghum", "Quinoa", "Corn", "Soybeans", "Peas",
  "Chickpeas", "Beans", "Fababeans", "Lentils")

for(crop in ALL_CROP_TYPES){
  VALUES_DATASET = add_crop_data(VALUES_DATASET, crop_id = crop)
}

write.csv(VALUES_DATASET, "CropData.csv", row.names=FALSE)

```

Code for Creating Plots for Single and Multi Variable Analysis

```
library(tidyverse)

graph_data = read.csv("CropData.csv", header = TRUE)
ggplot(graph_data, aes(x = factor(Year), y = Corn_Production, fill = Region)) +
  geom_bar(stat = "identity") +
  labs(title = "Corn Production: Saskatchewan and Manitoba",
       x = "Year", y = "Corn Production") +
  theme_minimal() +
  scale_fill_manual(values = c("#0092d6", "#da1a1a"))

ggplot(graph_data, aes(x = Year, y = Soybeans_Production, color = Region)) +
  geom_line(linewidth = 1) +
  geom_point(size = 3) +
  labs(title = "Soybeans Production: Saskatchewan and Manitoba",
       x = "Year", y = "Soybeans Production") +
  scale_color_manual(values = c("#FF6666", "#6699FF")) +
  theme_minimal()
```

Code for the R Shiny Website

The project is actually fairly huge, so it will require multiple files to run. I've separated it based off of files. For easier access, you may visit the repository on GitHub: <https://github.com/annahuynhly/stats780assignment1>

app.R

```
library(shiny)
library(shinycssloaders) # for loading screens
library(colourpicker)
library(tidyverse)

source("datasets.R")
source("functions.R")
source("contact_page.R")
source("graph_page.R")

ui = navbarPage(
  title = "Annual Crop Inventory of Saskatchewan and Manitoba between 2009 and 2013",
  tabPanel("Creating Graphs of Different Crops", page_home),
  tabPanel("Contact & Credits", page_contact_and_credits),
  id = "navbarID",
  theme = shinythemes::shinytheme("flatly"), # may want to change theme
)

server = function(input, output) {
  source("graphs_server.R", local = TRUE)$value
}
```

```
shinyApp(ui = ui, server = server)
```

datasets.R

```
COLOUR_MAP = read.csv("aci_crop_classifications.csv", header = TRUE, fileEncoding =
  ↪ "Latin1")

GRAPHING_DATASET = read.csv("CropData.csv", header = TRUE)

ALL_CROP_TYPES = c("Cereals", "Barley", "Millet", "Oats", "Rye", "Spelt", "Triticale",
  "Wheat", "Corn", "Soybeans", "Peas", "Beans", "Lentils")

COLOUR_THEME_LIST = list("Default Theme 1" = 'default1',
  "Default Theme 2" = 'default2',
  "Default Theme 3" = 'default3',
  "Lovely Mei" = 'lovelymei',
  "Jack in, Execute!" = "jackin",
  "Manually Insert" = 'manual')

DEFAULT1_COLOUR = c("#FF6666", "#6699FF", "#05DEB2", "#947aff", "#3333FF", "#5b10a7")
DEFAULT2_COLOUR = c("blue", "green", "red", "#b3bfff", "royalblue1", "#81ddff")
DEFAULT3_COLOUR = c("#EE4266", "#3cbbb1", "#b33c86", "#403f4c", "#0a0f0d", "#3185fc")
LOVELYMEI_COLOUR = c("#3800c2", "#676bf8", "#58887a", "#e69eb7", "#372f66", "#a2cda3")
EXECUTE_COLOUR = c("#0092d6", "#212c57", "#f85210", "#ffc710", "#0092d6", "#da1a1a")

COLOUR_TRANSLATION = list("default1" = DEFAULT1_COLOUR,
  "default2" = DEFAULT2_COLOUR,
  "default3" = DEFAULT3_COLOUR,
  "lovelymei" = LOVELYMEI_COLOUR,
  "jackin" = EXECUTE_COLOUR)
```

functions.R

```
convert_to_hex = function(hex_colour){
  #' Ensures the hex codes are in the correct format for plot building.
  hex_colour = gsub(" ", "", hex_colour)
  first_char = substr(hex_colour, 1, 1)
  if(first_char != "#"){
    return(paste("#", hex_colour, sep = ""))
  } else {
    return(hex_colour)
  }
}

comparison_barplot = function(dataset, crop_id, colours = c("#0092d6", "#da1a1a")){
  production_type = paste0(crop_id, "_Production")
  title_name = paste0("Stacked Bar Plot of ", crop_id, " Production from Manitoba and
  ↪ Saskatchewan")
}
```

```

# (GitHub Copilot helped with the skeleton)
ggplot(dataset, aes(x = factor(Year), y = !!sym(production_type), fill = Region)) +
  geom_bar(stat = "identity") +
  labs(title = title_name, x = "Year",
        y = paste0("Amount of Plots Used for ", crop_id, " Production")) +
  theme_minimal() +
  scale_fill_manual(values = colours[c(1, 2)])
}

comparison_lineplot = function(dataset, crop_id, colours = c("#0092d6", "#d9a1a1")){
  production_type = paste0(crop_id, "_Production")
  title_name = paste0("Line Plots of ", crop_id, " Production from Manitoba and
↪ Saskatchewan")
  # (GitHub Copilot helped with the skeleton)
  ggplot(dataset, aes(x = Year, y = !!sym(production_type), color = Region)) +
    geom_line(linewidth = 1) +
    geom_point(size = 3) +
    labs(title = title_name, x = "Year",
          y = paste0("Amount of Plots Used for ", crop_id, " Production")) +
    scale_color_manual(values = colours[c(1, 2)]) +
    theme_minimal()
}

individual_barplot = function(dataset, type, crop_id, colour = c("#0092d6")){
  production_type = paste0(crop_id, "_Production")
  if(type == "Manitoba"){
    graph_title = paste0("Stacked Bar Plot of ", crop_id, " Production from Manitoba")
    individual_data = subset(dataset, Region == "Manitoba")
  } else if (type == "Saskatchewan"){
    graph_title = paste0("Stacked Bar Plot of ", crop_id, " Production from
↪ Saskatchewan")
    individual_data = subset(dataset, Region == "Saskatchewan")
  }
  # (GitHub Copilot helped with the skeleton)
  ggplot(individual_data, aes(x = factor(Year), y = !!sym(production_type))) +
    geom_bar(stat = "identity", fill = colour[1]) +
    labs(title = graph_title, x = "Year",
          y = paste0("Amount of Plots Used for ", crop_id, " Production")) +
    theme_minimal()
}

individual_lineplot = function(dataset, type, crop_id, colour = c("#0092d6")){
  production_type = paste0(crop_id, "_Production")
  if(type == "Manitoba"){
    graph_title = paste0("Line Plot of ", crop_id, " Production from Manitoba")
    individual_data = subset(dataset, Region == "Manitoba")
  } else if (type == "Saskatchewan"){
    graph_title = paste0("Line Plot of ", crop_id, " Production from Saskatchewan")
    individual_data = subset(dataset, Region == "Saskatchewan")
  }
  # (GitHub Copilot helped with the skeleton)
  ggplot(individual_data, aes(x = Year, y = !!sym(production_type))) +
    geom_line(linewidth = 1, color = colour[1]) +

```

```

geom_point(size = 3, color = colour[1]) +
labs(title = graph_title, x = "Year",
      y = paste0("Amount of Plots Used for ", crop_id, " Production")) +
theme_minimal()
}

```

contact_page.R

```

page_contact = div(
  titlePanel("Contact"),

  p('This website is maintained by Anna Ly. If you find any bugs on this website, please
    ↪ contact
      lya19@mcmaster.ca. '),

  tags$style("#project-grid {
    display: grid;
    grid-template-columns: 120px 1fr;
    grid-gap: 10px;
  }"),
  div(id = "project-grid",
    div(id = "AnnaImg", img(src = "me.jpg", style = 'border-radius: 50%', width =
      ↪ '120px')),
    div(h3('Anna Ly'),
      h4('Graduate Student, M.Sc. in Statistics at McMaster University'),
      p("I like reading otome isekai. Also I did my undergrad at UofT.
        One day I want to make my own webtoon.", style = "color:#61646b"),
      tags$script(src = "https://kit.fontawesome.com/5e940c3ade.js"),
      tags$div(
        tags$i(class = "fa-brands fa-github"),
        tags$a(href="https://github.com/annahuynhly", "Github"), " | ",
        tags$i(class = "fa-brands fa-linkedin"),
        tags$a(href="https://www.linkedin.com/in/anna-ly-statistics-specialist/",
      ↪ "Linkedin"), " | ",
        tags$i(class = "fa-solid fa-graduation-cap"),
        tags$a(href="https://scholar.google.ca/citations?user=9w41oS8AAAAAJ&hl=en",
      ↪ "Google Scholar")
      ),
    ),
  ), # End of Project Grid
)

page_credit = div(
  titlePanel("Credits"),
  tags$div(
    "I constructed this website using
    ↪ ", tags$a(href="https://www.r-project.org/about.html", "R."), "Specifically, I
    ↪ used the ", tags$a(href="https://shiny.rstudio.com/", "R Shiny "), "package. The
    ↪ website theme is flatly from ",
    ↪ tags$a(href="https://rstudio.github.io/shinythemes/", "shinythemes."),
  ),
)

```

```

    br(),
    tags$div("I used the following additional Shiny packages: ",
    ↪ tags$a(href="https://cran.r-project.org/web/packages/shinycssloaders/index.html",
    ↪ "shinycssloaders"), " (for loading screens), ",
    ↪ tags$a(href="https://cran.r-project.org/web/packages/colourpicker/index.html",
    ↪ "colourpicker"), " (for users to manually select a colour).",
    ),
    br(),
    tags$div("I also used ",
    ↪ tags$a(href="https://cran.r-project.org/web/packages/tidyverse/index.html",
    ↪ "tidyverse"), " specifically ggplot2 to make the graphs."
    ),
    br(),
    tags$div("I also occasionally used ",
    ↪ tags$a(href="https://github.com/features/copilot","GitHub Copilot"), " to help me
    ↪ write code."
    ),
    br(),
    tags$div("The colour themes were self-chosen, except for \"Jack in, Execute!\" which is
    ↪ a reference to the ",
    ↪ tags$a(href="https://en.wikipedia.org/wiki/Mega_Man_Battle_Network", "Mega Man Battle
    ↪ Network Series.")
    )
  )
)

page_contact_and_credits = div(
  titlePanel("Contact & Credits"),
  tabsetPanel(type = "tabs",
    tabPanel("Credit", page_credit),
    tabPanel("Contact", page_contact),
  )
)

```

graph_page.R

```

page_home = div(
  titlePanel("Creating Graphs of Different Crops"),
  sidebarLayout(
    sidebarPanel(
      # GitHub Copilot helped with the skeleton for this section, particularly, the
      ↪ possible inputs.
      selectInput(inputId = "select_crop",
        label = "Select a crop!",
        choices = ALL_CROP_TYPES,
        selected = "Soybeans"),

      selectInput(inputId = "type_graph",
        label = "Select a type of graph to analyze.",
        choices = list("Stacked Bar Plot" = 1, "Line Plot" = 2),
        selected = 2),
    )
  )

```

```

selectInput(inputId = "compare_type", label = "Choose whether you prefer a
↳ comparison plot between two provinces or individual graphs.",
            choices = list("Comparison" = "comp", "Individual" = "ind")),

conditionalPanel(
  condition = "input.compare_type == 'ind'",

  selectInput(inputId = "select_region",
              label = "Select a province",
              choices = c("Saskatchewan", "Manitoba")),

  colourInput(inputId = "ind_colour",
              label = "Choose a colour for the plot",
              value = "6699FF"),
),

conditionalPanel(
  condition = "input.compare_type == 'comp'",

  selectInput(inputId = "colour_scheme_type",
              label = "What colour scheme would you prefer to use?",
              choices = COLOUR_THEME_LIST,
              selected = "manual"),

  conditionalPanel(
    condition = "input.colour_scheme_type == 'manual'",
    colourInput(inputId = "comp_sask_colour",
                label = "Choose a colour to represent Saskatchewan",
                value = "6699FF"),
    colourInput(inputId = "comp_mani_colour",
                label = "Choose a colour to represent Manitoba",
                value = "05DEB2"),
  ),

), # End of conditional Panel

), # End of sidebarPanel
mainPanel(

  conditionalPanel(
    condition = "input.compare_type == 'comp'",
    withSpinner(plotOutput("comparison_crop_plot"))
  ),

  conditionalPanel(
    condition = "input.compare_type == 'ind'",
    withSpinner(plotOutput("individual_crop_plot"))
  ),

) # End of mainPanel
)
)

```


graphs_server.R

```
use_colours = reactive({
  if(input$colour_scheme_type == "manual"){
    c(convert_to_hex(input$comp_mani_colour), convert_to_hex(input$comp_sask_colour))
  } else {
    COLOUR_TRANSLATION[[input$colour_scheme_type]]
  }
})

individual_barplot_value = reactive({
  individual_barplot(dataset = GRAPHING_DATASET,
    type = input$select_region,
    crop_id = input$select_crop,
    colour = convert_to_hex(input$ind_colour))
})

individual_lineplot_value = reactive({
  individual_lineplot(dataset = GRAPHING_DATASET,
    type = input$select_region,
    crop_id = input$select_crop,
    colour = convert_to_hex(input$ind_colour))
})

comparison_barplot_value = reactive({
  comparison_barplot(dataset = GRAPHING_DATASET,
    crop_id = input$select_crop,
    colours = use_colours())
})

comparison_lineplot_value = reactive({
  comparison_lineplot(dataset = GRAPHING_DATASET,
    crop_id = input$select_crop,
    colours = use_colours())
})

# Making the comparison plots
output$comparison_crop_plot = renderPlot({
  if(input$type_graph == 1){ # Stacked Bar Plot case
    comparison_barplot_value()
  } else if (input$type_graph == 2){ # Line Plot case
    comparison_lineplot_value()
  }
})

output$individual_crop_plot = renderPlot({
  if(input$type_graph == 1){ # Stacked Bar Plot case
    individual_barplot_value()
  } else if (input$type_graph == 2){ # Line Plot case
    individual_lineplot_value()
  }
})
```