1. Dataset Sourcing

This dataset was sourced from the Open Government Portal of Canada, a platform providing public access to a wide range of datasets. To locate the specific dataset: set **Portal Type** to *Open Data*, **Collection Type** to *Open Maps*, **Resource Type** to *Dataset* and **Format** to *GeoTIF*.

The dataset "*Monthly Salinity Climatology of the Northwest Atlantic Ocean from BNAM model (1990-2015)*" (Canada and Oceans 2020) consists of monthly mean salinity measurements derived from the Bedford Institute of Oceanography North Atlantic Model. Variables included in the dataset are monthly salinity averages measured in practical salinity units (PSU) calculated at multiple depths: surface (0m), 110m, 156m, 222m, 318m, 541m, 1062m, and bottom. Each measurement has a spatial resolution of 1/12 degree, latitudes of 8°-75°N and longitudes of 100°W-30°E.

2. Application Description

Salinity levels in the ocean are influenced by various factors including temperature changes and freshwater influx, which are directly linked to climate change. We can analyze long-term salinity trends/variations at various ocean depths and understand the impacts of climatic shifts on ocean salinity. For context, changes in ocean salinity can indicate shifts influenced by climate change. By monitoring these changes, we can develop models to predict climate dynamics, contributing to more accurate climate change predictions.

3. Data Transformation

Data Selection

We included all available data on the Open Government Portal, which included salinity measures encompassing spatial variations in depths and temporal variations in months from 1990 to 2015. This approach allowed us to capture a broad yet detailed perspective of salinity trends and variations.

Data Preprocessing

All data processing, analyses, and visualization was performed in R R Core Team (2020). This involved downloading and loading GeoTIF files and extracting metadata from the file names. We identified missing values directly from the raster, where NA represented areas without salinity measurements. These were excluded from the dataset. Each raster's coordinates and salinity values were compiled into a single dataframe, organized by month and depth. The tidy dataset was saved in an R data format.

In this preprocessing phase, outliers were not specifically addressed because all recorded salinity values (across different ocean depths and months) are valid observations in the context of geological and oceanographic data. Given the dataset's scientific basis, each measurement reflects real-world variations in ocean salinity and is thus integral to the integrity and completeness of the analysis. Therefore, no data points were dismissed as outliers.

4. Single-Variable Analysis

How does *salinity* vary in relation to *ocean depth* over a 25-year period in the Northwest Atlantic Ocean?

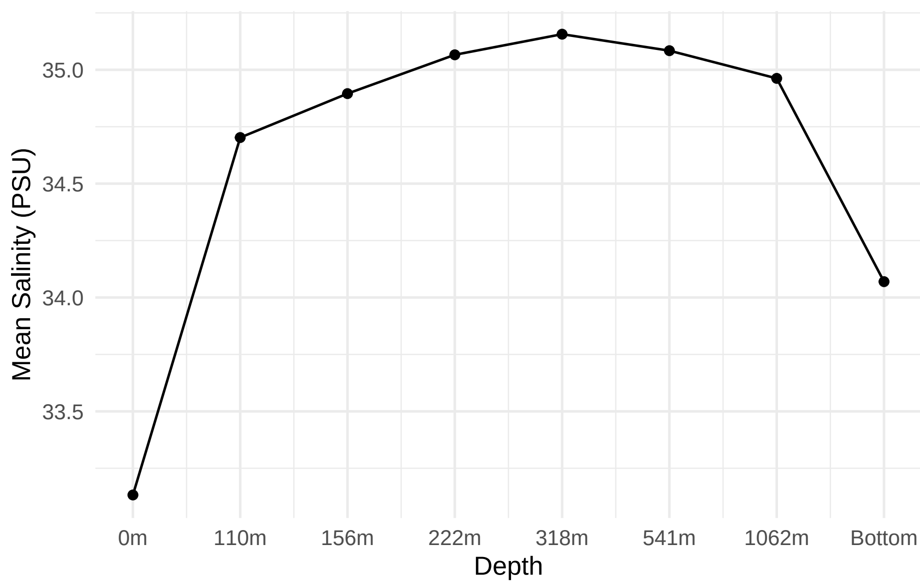

Figure 1: 25-year Average Salinity of 8 ocean levels.

Average salinity increases as the depth increases up to about 541m, suggesting that deeper water layers are saltier, possibly due to lesser influence from freshwater inputs. However, beyond 541m, the salinity begins to decrease, reaching another low at the ocean floor (bottom). This might indicate influences from geological features or deep ocean currents that could introduce less saline waters at these depths.

5. Multi-Variable Analysis

How does the 25-year *average ocean salinity* vary across different *months* and *ocean depths* in the Northwest Atlantic Ocean? (25-Year Average Salinity ~ Month + Depth)

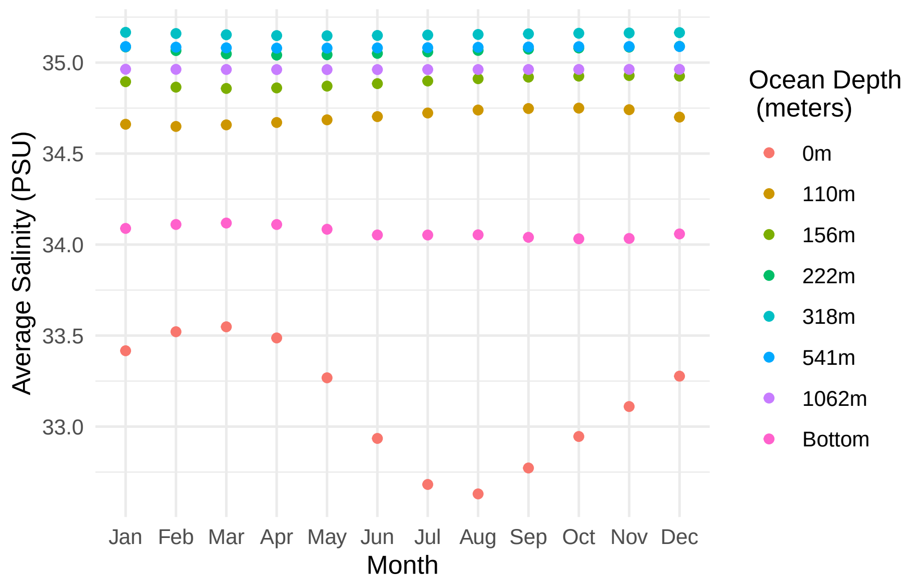

Figure 2: 25-year Average Salinity of 8 ocean levels. All salinity levels are averaged by month.

There is an overall trend where salinity appears to be higher during the summer months (June to August) across deeper depths. The surface layer however, has the opposite trend where salinity is lower in the summer. Salinity values are generally higher in deeper waters, particularly at depths of 541m and below, across all months. This consistency at greater depths might indicate less influence from surface environmental changes, suggesting more stable conditions in deeper ocean depths.

6. Shiny App

Despite the average salinity levels being relatively high (between 32 and 36 PSU), different geographical locations have vastly different salinity levels. The attached Shiny app Chang et al. (2024) features an interactive interface where users can select specific months and depths to explore detailed salinity distributions. The visualization on the right provides a colour-gradient plot that illustrates the available geographical variations in salinity at the selected depth and month. Additionally, users have the option to view either averaged salinity data across all depths and months or a histogram detailing salinity distribution at the chosen depth and month.

References

Canada, Fisheries, and Oceans. 2020. "Monthly Salinity Climatology of the Northwest Atlantic Ocean from BNAM model (1990-2015) - Open Government Portal." https://open.canada.ca/data/en/dataset/c44a8574-9f7d-45b7-afda-27802353a04c.

Chang, Winston, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, and Barbara Borges. 2024. *Shiny: Web Application Framework for r*. https://CRAN.R-project.org/package=shiny.

ChatGPT. 2024. "R Code Extract Word Before the Second Last _ Str_extract(basename(file_names), "(?<=_)[^_]+(?=
.)"." https://chat.opeani.com.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the Tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Supplementary Materials

ChatGPT4 ChatGPT (2024) used on 09/12/2024 for extracting parameter from file name.

The Tidyverse package was loaded and subsequent analyses were done using Tidyverse functions Wickham et al. (2019).

```
################################################################################
# DV: monthly mean salinity across 25 years (1990-2015)
# IV1: depth
# IV2: month

# Load packages
library(tidyverse)
library(terra)
library(plotly)

# Load multiple GeoTIFF files---------------------------------------------------
folder_path <- 'data/NorthwestAtlantic_Salinity/TiffAll'

# List all the tiff files in the folder
file_names <- list.files(
  folder_path,
  pattern = "\\.tif$",
  full.names = TRUE)

# Extract the depth and month from the file names
ocean_depth <- str_extract(basename(file_names), "(?<=_)[^_]+(?=\\.)")
# ChatGPT: R extract word before the second last _
# str_extract(basename(file_names),"(?<=_)[^_]+(?=\\.)"
month <- str_extract(basename(file_names), "(?<=_)[^_]+(?=_[^_]+$)")

df <- data.frame(
```

```
  depth = ocean_depth,
  month = month,
  file_name = file_names
)

# Create empty list to store individual rasters
df_list <- list()

# Loop through file_names to process each raster
for (i in seq_along(df$file_name)) {

  # Load raster file
  raster_data <- rast(df$file_name[i])

  # Extract values from the first layer (assuming single layer raster)
  layer_1_value <- values(raster_data)[,1] |> as.numeric()
  coords <- xyFromCell(raster_data, 1:ncell(raster_data))

  # Find the non-NA values
  non_na_cells <- which(!is.na(layer_1_value))

  # Filter out NA values from the raster values and coordinates
  layer_1_value <- layer_1_value[non_na_cells]
  coords <- coords[non_na_cells, ]

  # Create a temporary dataframe with the current raster's data
  temp <- data.frame(
    value = layer_1_value,
    x = coords[, 1],
    y = coords[, 2],
    month = df$month[i],
```

```
    depth = df$depth[i]
  ) |> as_tibble()

  # Add temp to the list
  df_list[[i]] <- temp
}

# Combine all data frames into one final data frame
df_coords <- bind_rows(df_list)

# Manually order the month and depth columns
df_coords$depth <- factor(
  df_coords$depth,
  levels = c("0m", "110m", "156m", "222m", "318m", "541m", "1062m", "Bottom")
)
df_coords$month <- factor(
  df_coords$month,
  levels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "D
)

# Arrange the data frame by month and depth
df_coords <- df_coords %>%
  arrange(month, depth)

save(df_coords, file = "shiny/df_coords.RData")



# Data analysis
# Summarize the data
df_summary <- df_coords %>%
  group_by(month, depth) %>%
```

```
  summarize(avg_value = mean(value, na.rm = TRUE), .groups = 'drop')

# Create a scatterplot
avg_plot <- ggplot(df_summary, aes(x = month, y = avg_value, colour = depth)) +
  geom_point() +
  labs(title = "Average Salinity by Month and Depth",
       x = "Month",
       y = "Average Salinity") +
  theme_minimal() +
  scale_colour_discrete("Depth")

# Convert to ggplotly
avg_plotly <- ggplotly(avg_plot)
```

Shiny App

```
# Shiny App: STATS780 Assignment 1
# Jackie Zhou

# Imports
library(shiny)
library(plotly)
library(ggplot2)
library(dplyr)
library(viridis)

# Load data
load("shiny/df_coords.RData")

################################################################################
# App UI
ui <- fluidPage(
```

```r
titlePanel("25-Year Average Salinity in Northwest Atlantic Ocean"),

sidebarLayout(

  sidebarPanel(
    # Select month
    selectInput(
      inputId = "month",
      label = "Select a Month",
      choices = unique(df_coords$month),
      selected = "Jan"
    ),

    # Select depth
    selectInput(
      inputId = "depth",
      label = "Select a Depth",
      choices = unique(df_coords$depth),
      selected = "0m"
    ),

    # Select second plot
    selectInput(
      inputId = "plotType",
      label = "Choose Plot Type:",
      choices = c("Average" = "average", "Selected" = "selected"),
      selected = "selected"
    ),
    width = 2
  ),
  mainPanel(
```

```r
      # Plotly output for rendering the interactive plot
      plotlyOutput(outputId = "plot", width = "1000px", height = "400px"),
      width = 8
    )
  )
)

################################################################################
# App Server
server <- function(input, output) {

  # Render ggplotly plot based on ui
  output$plot <- renderPlotly({

    # Calculate average salinity per month and depth across entire dataset
    # For second plot option (average)
    avg_data <- df_coords %>%
      group_by(month, depth) %>%
      summarise(avg_salinity = mean(value, na.rm = TRUE), .groups = 'drop')

    # Subset data based on input month and depth
    filtered_dat <- df_coords %>%
      filter(month == input$month, depth == input$depth)

    # Plots --------------------------------------------------------------------
    # Create raster ggplot -----------------------------------------------------
    rast_plot <- ggplot(filtered_dat, aes(x = x, y = y, fill = value)) +
      geom_tile() +
      scale_fill_viridis(
        "Salinity",
        #colors = rainbow(5),
```

```
    #limits = c(min(df_coords$value), max(df_coords$value))
  ) +
  labs(title = paste0(
    "Ocean Salinity Data: ",
    input$month,
    "_",
    input$depth)
  ) +
  theme_minimal()

# Convert ggplot to plotly plot, add labels
rast_plotly <- ggplotly(rast_plot) %>%
  layout(
    xaxis = list(title = 'Longitude'),
    yaxis = list(title = 'Latitude')
  )

# Create histogram for salinity values ------------------------------------
hist_plot <- ggplot(filtered_dat, aes(x = value)) +
  geom_histogram(bins = 20, fill = "darkgrey") +
  labs(title = "Salinity Distribution") +
  theme_minimal()

# Create ggplotly object for histogram plot and add labels
hist_plotly <- ggplotly(hist_plot) %>%
  layout(xaxis = list(title = 'Salinity'),
         yaxis = list(title = 'Count'),
         legend = list(x = 0))

# Create average plot -----------------------------------------------------
avg_plot <- ggplot(
```

```r
    avg_data,
    aes(x = month, y = avg_salinity, color = depth)
  ) +
    geom_point() +
    labs(title = "Average Salinity per Month and Depth",
         x = "Month",
         y = "Average Salinity") +
    theme_minimal() +
    scale_colour_discrete("Month")

  avg_poltly <- ggplotly(avg_plot)


  # Conditionally display second plot based on plotType-----------------------
  if (input$plotType == "average") {
    subplot(
      rast_plotly,
      avg_plotly,
      nrows = 1,
      widths = c(0.6, 0.4),
      margin = 0.03,
      titleX = TRUE,
      titleY = TRUE
    ) %>%
      layout(legend = list(x = 1.12, y = 0.05))
  } else {
    subplot(
      rast_plotly,
      hist_plotly,
      nrows = 1,
      widths = c(0.6, 0.4),
```

```
        margin = 0.08,
        which_layout = 1,
        titleX = TRUE,
        titleY = TRUE
      )
    }

  })
}

################################################################################
# Run App
shinyApp(ui, server)

################################################################################
```