~~~
---
title: "Case Study 2: AKSTA Statistical Computing"
author: "Zuhaib Yousaf Zai"
date: "`r Sys.Date()`"
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
library(readxl)
library(tidyr)
~~~

Introduction

This case study analyzes world data from 2020, focusing on education expenditure, youth unemployment rate, and net migration rate. The data is sourced from the CIA World Factbook Archives. The objective is to clean, merge, and analyze the data to provide insights into global trends.

Task a: Data Import and Cleaning

Education Data

The education data is loaded from a text file. We skip the first two lines and use two or more spaces as the delimiter.

```
# Load education data
education_data <- read.table("rawdata_369.txt", header = FALSE, skip = 2, sep = "", fill = TRUE,
stringsAsFactors = FALSE)
names(education_data) <- c("country", "education_expenditure", "year")
education_data <- education_data[, c("country", "education_expenditure")]
```

Unemployment Data

The unemployment data is imported from an Excel file. We assign clear column names for consistency.

```
# Load unemployment data from Excel
unemployment_data <- read_excel("rawdata 373.xls")
names(unemployment_data) <- c("country", "youth_unemployment_rate")
```

Migration Data

Migration data is also loaded from a text file. We ensure only relevant columns are retained.

```
# Load migration data
migration_data <- read.table("rawdata_347.txt", header = FALSE, sep = "", fill = TRUE,
stringsAsFactors = FALSE)
names(migration_data) <- c("country", "net_migration_rate", "year")
migration_data <- migration_data[, c("country", "net_migration_rate")]
```

Data Cleaning

We convert country columns to character type to ensure consistency across datasets.

```
# Convert country columns to character
education_data$country <- as.character(education_data$country)
unemployment_data$country <- as.character(unemployment_data$country)
migration_data$country <- as.character(migration_data$country)
```

Task b: Merging Raw Data

We merge the datasets using the country column as the key, ensuring we retain all observations.

```
# Merge datasets using full join
merged_data <- full_join(education_data, unemployment_data, by = "country")
merged_data <- full_join(merged_data, migration_data, by = "country")

# Return dimensions of the merged dataset
dim(merged_data)
```

Task c: Enriching Data with Income Classification

Load Income Data

Income classification data is obtained from the World Bank and merged using standardized country names.

```
# Load income classification data
income_data <- read_excel("CLASS.xlsx")
names(income_data) <- c("country", "iso_code", "region", "income_group", "lending_category")
```

Merge Income Data

We standardize country names to ensure successful merging.

```
# Clean and standardize country names
standardize_country_names <- function(df) {
  df$country <- trimws(tolower(gsub("[^[:alnum:] ]", "", as.character(df$country))))
  return(df)
}

merged_data <- standardize_country_names(merged_data)
income_data <- standardize_country_names(income_data)

# Merge datasets
enriched_data <- left_join(merged_data, income_data, by = "country")
```

Task d: Adding Geographical Information

Load and Merge Continent Data

We introduce continent and subcontinent data, ensuring no countries are lost in the merge.

```
# Load the continent dataset
continent_data <- read.csv("continents2.csv")
names(continent_data)[names(continent_data) == "region"] <- "continent"
continent_data <- standardize_country_names(continent_data)

# Merge datasets
df_vars <- left_join(enriched_data, continent_data, by = "country")
```

Task e: Data Tidiness and Summary Statistics

Evaluate Tidiness

We remove duplicates and filter out missing values to tidy the data.

```
# Remove duplicates and filter missing values
df_vars <- df_vars %>% distinct() %>% filter(!is.na(income_group), !is.na(continent))
```

Frequency Table

We create a frequency table for the income status variable to interpret the distribution of countries.

```
# Frequency table for income status
income_status_freq <- table(df_vars$income_group)
income_status_freq
```

Task f: Further Summary Statistics and Insights

Summary Statistics

We calculate mean and median values for each variable, grouped by income status.

```
# Calculate mean and median values
average_stats <- df_vars %>%
  group_by(income_group) %>%
  summarise(
    mean_expenditure = mean(education_expenditure, na.rm = TRUE),
    median_expenditure = median(education_expenditure, na.rm = TRUE),
    mean_youth_unemployment = mean(youth_unemployment_rate, na.rm = TRUE),
    median_youth_unemployment = median(youth_unemployment_rate, na.rm = TRUE),
    mean_net_migration = mean(net_migration_rate, na.rm = TRUE),
    median_net_migration = median(net_migration_rate, na.rm = TRUE)
  ) %>%
  arrange(factor(income_group, levels = c("L", "LM", "UM", "H")))
average_stats
```

Task g: Conditional Probabilities

We estimate probabilities based on observed frequencies, focusing on high income and migration conditions.

```
# Calculate conditional probabilities
prior_high_income <- df_vars %>%
  summarise(prior_probability = mean(income_group == "high", na.rm = TRUE))

conditional_high_income_europe <- df_vars %>%
  filter(continent == "Europe") %>%
  summarise(conditional_probability = mean(income_group == "high", na.rm = TRUE))

conditional_negative_migration <- df_vars %>%
  filter(youth_unemployment_rate > 25) %>%
  summarise(conditional_probability = mean(net_migration_rate < 0, na.rm = TRUE))

prior_high_income
conditional_high_income_europe
conditional_negative_migration
```

Task h: Simpson’s Paradox Analysis

We investigate trends in youth unemployment rates across income groups and continents.

```
# Calculate average youth unemployment rates by income group and continent
continent_unemployment <- df_vars %>%
  group_by(continent, income_group) %>%
  summarise(continent_average_unemployment = mean(youth_unemployment_rate, na.rm =
TRUE), .groups = 'drop')
continent_unemployment
```

Task i: Data Export

Finally, we export the tidy dataset as a CSV file with specified formatting.

```
# Export the final tidy dataset as a CSV
write.table(df_vars, file = "final_dataset.csv", sep = ";", na = ".", row.names = FALSE, quote = FALSE)
```

Conclusion

This document outlines the steps taken to analyze world data from the CIA World Factbook, focusing on education expenditure, youth unemployment, and migration rates. The analysis includes data cleaning, merging, and statistical analysis, culminating in a tidy dataset ready for submission.