Recreated School project

Robert Lupo

2024-06-30

# Confirm the current working directory  
print(getwd())

## [1] "C:/Users/treyl/Downloads/WORK/Work Projects"

# Load the male life tables using the full path  
male\_life\_tables <- read.table("C:/Users/treyl/OneDrive/Documents/School/School/2023/Fall 2023/Computational Methods for Applied Statistics-STA4102-001/Project/DATA/lt\_male/mltper\_1x1/USA.mltper\_1x1.txt", header = TRUE, skip = 2)  
  
# Load the female life tables using the full path  
female\_life\_tables <- read.table("C:/Users/treyl/OneDrive/Documents/School/School/2023/Fall 2023/Computational Methods for Applied Statistics-STA4102-001/Project/DATA/lt\_female/fltper\_1x1/USA.fltper\_1x1.txt", header = TRUE, skip = 2)  
  
# Check the first few rows of the loaded data  
head(male\_life\_tables)

## Year Age mx qx ax lx dx Lx Tx ex  
## 1 1933 0 0.06818 0.06486 0.25 100000 6486 95139 5918927 59.19  
## 2 1933 1 0.01004 0.00999 0.50 93514 934 93047 5823788 62.28  
## 3 1933 2 0.00467 0.00466 0.50 92580 431 92364 5730741 61.90  
## 4 1933 3 0.00333 0.00333 0.50 92148 307 91995 5638377 61.19  
## 5 1933 4 0.00254 0.00253 0.50 91842 233 91725 5546382 60.39  
## 6 1933 5 0.00209 0.00209 0.50 91609 191 91513 5454656 59.54

head(female\_life\_tables)

## Year Age mx qx ax lx dx Lx Tx ex  
## 1 1933 0 0.05418 0.05208 0.26 100000 5208 96130 6279695 62.80  
## 2 1933 1 0.00887 0.00883 0.50 94792 837 94374 6183565 65.23  
## 3 1933 2 0.00402 0.00402 0.50 93955 377 93767 6089191 64.81  
## 4 1933 3 0.00287 0.00287 0.50 93578 268 93444 5995425 64.07  
## 5 1933 4 0.00223 0.00223 0.50 93310 208 93206 5901981 63.25  
## 6 1933 5 0.00185 0.00185 0.50 93102 172 93016 5808775 62.39

# Part A) Picked Human Mortality Database  
# Problem Statement: Develop models to predict life expectancy for males and females based on their age and sex  
  
# Part B) Data Cleaning and Wrangling  
  
male\_life\_tables$Gender <- 0 # Add Gender column for males (assign 0)  
  
female\_life\_tables$Gender <- 1 # Add Gender column for females (assign 1)  
  
# Combine male and female life tables  
combined\_life\_tables <- rbind(male\_life\_tables, female\_life\_tables)  
  
# Convert Age to numeric (remove 'm' or 'f' prefix)  
combined\_life\_tables$Age <- as.numeric(gsub("[^0-9]", "", combined\_life\_tables$Age))  
  
# Check the structure of the updated data frame  
str(combined\_life\_tables)

## 'data.frame': 19758 obs. of 11 variables:  
## $ Year : int 1933 1933 1933 1933 1933 1933 1933 1933 1933 1933 ...  
## $ Age : num 0 1 2 3 4 5 6 7 8 9 ...  
## $ mx : num 0.06818 0.01004 0.00467 0.00333 0.00254 ...  
## $ qx : num 0.06486 0.00999 0.00466 0.00333 0.00253 ...  
## $ ax : num 0.25 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...  
## $ lx : int 100000 93514 92580 92148 91842 91609 91418 91244 91085 90938 ...  
## $ dx : int 6486 934 431 307 233 191 173 159 147 137 ...  
## $ Lx : int 95139 93047 92364 91995 91725 91513 91331 91165 91012 90870 ...  
## $ Tx : int 5918927 5823788 5730741 5638377 5546382 5454656 5363143 5271812 5180647 5089635 ...  
## $ ex : num 59.2 62.3 61.9 61.2 60.4 ...  
## $ Gender: num 0 0 0 0 0 0 0 0 0 0 ...

# Select relevant columns  
relevant\_columns <- combined\_life\_tables[c("Gender", "Age", "ex")]  
  
# Check the structure of the relevant data frame  
str(relevant\_columns)

## 'data.frame': 19758 obs. of 3 variables:  
## $ Gender: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Age : num 0 1 2 3 4 5 6 7 8 9 ...  
## $ ex : num 59.2 62.3 61.9 61.2 60.4 ...

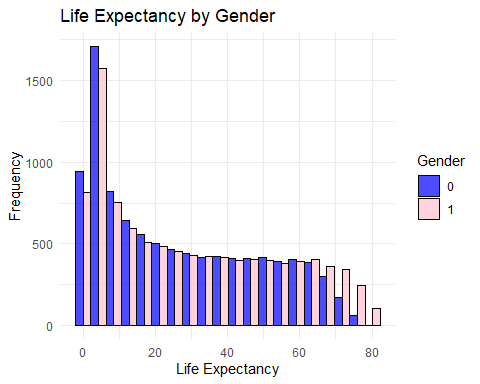
# Check for missing values  
missing\_values <- sum(is.na(relevant\_columns))  
cols\_with\_missing <- colnames(relevant\_columns)[colSums(is.na(relevant\_columns)) > 0]  
  
# Check for duplicates after combining male and female life tables  
duplicates\_after\_combination <- combined\_life\_tables[duplicated(combined\_life\_tables), ]  
  
if (nrow(duplicates\_after\_combination) > 0) {  
 print("Duplicate Rows After Combining Male and Female Life Tables:")  
 print(duplicates\_after\_combination)  
} else {  
 print("No duplicate rows found after combining male and female life tables.")  
}

## [1] "No duplicate rows found after combining male and female life tables."

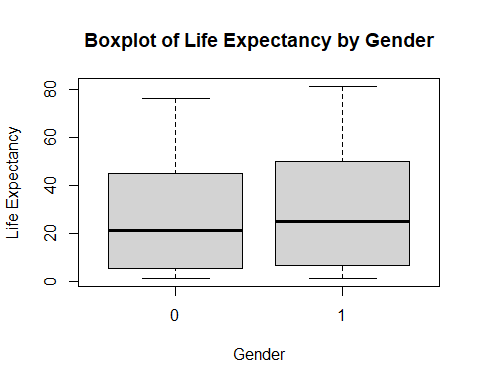
# Part C) Exploratory Data Analysis (EDA)  
# Part 1) Summary Statistics  
summary(relevant\_columns)

## Gender Age ex   
## Min. :0.0 Min. : 0 Min. : 1.32   
## 1st Qu.:0.0 1st Qu.: 27 1st Qu.: 6.08   
## Median :0.5 Median : 55 Median :23.09   
## Mean :0.5 Mean : 55 Mean :28.21   
## 3rd Qu.:1.0 3rd Qu.: 83 3rd Qu.:47.62   
## Max. :1.0 Max. :110 Max. :81.48

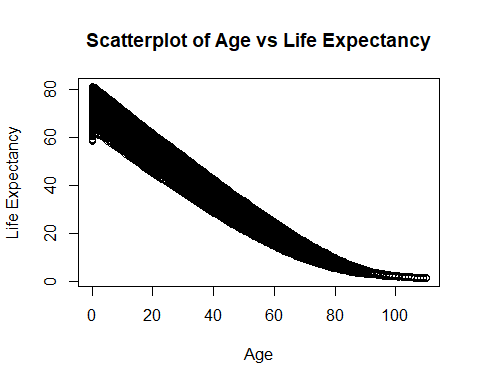
# Part 2) Visualizations  
if (!requireNamespace("ggplot2", quietly = TRUE)) {  
 install.packages("ggplot2")  
}  
library(ggplot2)  
  
ggplot\_data <- data.frame(Gender = factor(relevant\_columns$Gender),  
 LifeExpectancy = relevant\_columns$ex)  
  
# Create a side-by-side histogram using ggplot2  
ggplot(ggplot\_data, aes(x = LifeExpectancy, fill = Gender)) +  
 geom\_histogram(position = "dodge", bins = 20, color = "black", alpha = 0.7) +  
 labs(title = "Life Expectancy by Gender",  
 x = "Life Expectancy",  
 y = "Frequency") +  
 scale\_fill\_manual(values = c("blue", "pink")) +  
 theme\_minimal()



# Boxplot of Life Expectancy by Gender  
boxplot(ex ~ Gender, data = relevant\_columns, main = "Boxplot of Life Expectancy by Gender", xlab = "Gender", ylab = "Life Expectancy")



# Scatterplot of Age vs Life Expectancy  
plot(relevant\_columns$Age, relevant\_columns$ex, main = "Scatterplot of Age vs Life Expectancy", xlab = "Age", ylab = "Life Expectancy")



# Part 3) Identify Relationships or Trends  
# Correlation Matrix  
correlation\_matrix <- cor(relevant\_columns)  
print(correlation\_matrix)

## Gender Age ex  
## Gender 1.00000000 0.0000000 0.07164447  
## Age 0.00000000 1.0000000 -0.96990681  
## ex 0.07164447 -0.9699068 1.00000000

# Linear Model: Life Expectancy ~ Age + Gender  
linear\_model <- lm(ex ~ Age + Gender, data = relevant\_columns)  
summary(linear\_model)

##   
## Call:  
## lm(formula = ex ~ Age + Gender, data = relevant\_columns)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -10.1630 -4.0546 -0.9562 3.8105 13.4685   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 64.969802 0.084948 764.82 <2e-16 \*\*\*  
## Age -0.698348 0.001192 -585.84 <2e-16 \*\*\*  
## Gender 3.305743 0.076390 43.27 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.369 on 19755 degrees of freedom  
## Multiple R-squared: 0.9459, Adjusted R-squared: 0.9458   
## F-statistic: 1.725e+05 on 2 and 19755 DF, p-value: < 2.2e-16

# Extract R-squared and MSE for Linear Model  
r\_squared <- summary(linear\_model)$r.squared  
mse <- mean(linear\_model$residuals^2)  
print(paste("Linear Regression R-squared:", r\_squared))

## [1] "Linear Regression R-squared: 0.945852156317003"

print(paste("Linear Regression MSE:", mse))

## [1] "Linear Regression MSE: 28.820008903982"

# Part D) Statistical Analysis  
# Part 1) Hypothesis Testing  
t\_test\_result <- t.test(ex ~ Gender, data = relevant\_columns)  
print(t\_test\_result)

##   
## Welch Two Sample t-test  
##   
## data: ex by Gender  
## t = -10.096, df = 19614, p-value < 2.2e-16  
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0  
## 95 percent confidence interval:  
## -3.947536 -2.663951  
## sample estimates:  
## mean in group 0 mean in group 1   
## 26.56067 29.86642

# Part 2) Correlation Analysis  
correlation\_age\_ex <- cor(relevant\_columns$Age, relevant\_columns$ex)  
print(paste("Correlation between Age and Life Expectancy:", correlation\_age\_ex))

## [1] "Correlation between Age and Life Expectancy: -0.969906813047034"

# Part 3) Linear Regression  
linear\_model <- lm(ex ~ Age + Gender, data = relevant\_columns)  
summary(linear\_model)

##   
## Call:  
## lm(formula = ex ~ Age + Gender, data = relevant\_columns)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -10.1630 -4.0546 -0.9562 3.8105 13.4685   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 64.969802 0.084948 764.82 <2e-16 \*\*\*  
## Age -0.698348 0.001192 -585.84 <2e-16 \*\*\*  
## Gender 3.305743 0.076390 43.27 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.369 on 19755 degrees of freedom  
## Multiple R-squared: 0.9459, Adjusted R-squared: 0.9458   
## F-statistic: 1.725e+05 on 2 and 19755 DF, p-value: < 2.2e-16

# Extract R-squared and MSE for Linear Model  
r\_squared <- summary(linear\_model)$r.squared  
mse <- mean(linear\_model$residuals^2)  
print(paste("Linear Regression R-squared:", r\_squared))

## [1] "Linear Regression R-squared: 0.945852156317003"

print(paste("Linear Regression MSE:", mse))

## [1] "Linear Regression MSE: 28.820008903982"

# Part 4) Advanced Technique: Ridge Regression  
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-8

X <- model.matrix(ex ~ Age + Gender, data = relevant\_columns)[, -1]  
y <- relevant\_columns$ex  
  
ridge\_model <- cv.glmnet(X, y, alpha = 0) # alpha = 0 for ridge regression  
print(ridge\_model)

##   
## Call: cv.glmnet(x = X, y = y, alpha = 0)   
##   
## Measure: Mean-Squared Error   
##   
## Lambda Index Measure SE Nonzero  
## min 2.238 100 32.77 0.3012 2  
## 1se 2.238 100 32.77 0.3012 2

ridge\_predictions <- predict(ridge\_model, s = "lambda.min", newx = X)  
ridge\_residuals <- y - ridge\_predictions  
ridge\_rmse <- sqrt(mean(ridge\_residuals^2))  
ridge\_r\_squared <- cor(y, ridge\_predictions)^2  
print(paste("Ridge Regression RMSE:", ridge\_rmse))

## [1] "Ridge Regression RMSE: 5.72323528909707"

print(paste("Ridge Regression R-squared:", ridge\_r\_squared))

## [1] "Ridge Regression R-squared: 0.945852156317003"

# Install and load the caret package for data partitioning  
if (!requireNamespace("caret", quietly = TRUE)) {  
 install.packages("caret")  
}  
library(caret)

## Loading required package: lattice

# Split the data into training and testing sets  
set.seed(123)  
index <- createDataPartition(relevant\_columns$ex, p = 0.8, list = FALSE)  
train\_data <- relevant\_columns[index, ]  
test\_data <- relevant\_columns[-index, ]  
  
# Fit the linear regression model on the training data  
linear\_model <- lm(ex ~ Age + Gender, data = train\_data)  
linear\_predictions <- predict(linear\_model, newdata = test\_data)  
  
# Evaluate the performance of the linear regression model on the testing data  
linear\_rmse <- sqrt(mean((test\_data$ex - linear\_predictions)^2))  
test\_r\_squared <- 1 - (sum((test\_data$ex - linear\_predictions)^2) / sum((test\_data$ex - mean(test\_data$ex))^2))  
test\_mse <- mean((test\_data$ex - linear\_predictions)^2)  
print(paste("Linear Regression RMSE on Testing Data:", linear\_rmse))

## [1] "Linear Regression RMSE on Testing Data: 5.44076952018271"

print(paste("Linear Regression R-squared on Testing Data:", test\_r\_squared))

## [1] "Linear Regression R-squared on Testing Data: 0.943964951833563"

print(paste("Linear Regression MSE on Testing Data:", test\_mse))

## [1] "Linear Regression MSE on Testing Data: 29.6019729717492"

# Load required libraries  
library(rpart)  
library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(e1071)  
library(gbm)

## Loaded gbm 2.1.9

## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.com/gbm-developers/gbm3

# Decision Trees  
tree\_model <- rpart(ex ~ Age + Gender, data = train\_data)  
tree\_predictions <- predict(tree\_model, newdata = test\_data)  
tree\_rmse <- sqrt(mean((test\_data$ex - tree\_predictions)^2))  
tree\_r\_squared <- cor(test\_data$ex, tree\_predictions)^2  
print(paste("Decision Tree RMSE on Testing Data:", tree\_rmse))

## [1] "Decision Tree RMSE on Testing Data: 5.07050790592342"

print(paste("Decision Tree R-squared on Testing Data:", tree\_r\_squared))

## [1] "Decision Tree R-squared on Testing Data: 0.95133445873889"

# Random Forest  
rf\_model <- randomForest(ex ~ Age + Gender, data = train\_data)  
rf\_predictions <- predict(rf\_model, newdata = test\_data)  
rf\_rmse <- sqrt(mean((test\_data$ex - rf\_predictions)^2))  
rf\_r\_squared <- cor(test\_data$ex, rf\_predictions)^2  
print(paste("Random Forest RMSE on Testing Data:", rf\_rmse))

## [1] "Random Forest RMSE on Testing Data: 8.20690478673503"

print(paste("Random Forest R-squared on Testing Data:", rf\_r\_squared))

## [1] "Random Forest R-squared on Testing Data: 0.951841178176324"

# Support Vector Machine  
svm\_model <- svm(ex ~ Age + Gender, data = train\_data)  
svm\_predictions <- predict(svm\_model, newdata = test\_data)  
svm\_rmse <- sqrt(mean((test\_data$ex - svm\_predictions)^2))  
svm\_r\_squared <- cor(test\_data$ex, svm\_predictions)^2  
print(paste("SVM RMSE on Testing Data:", svm\_rmse))

## [1] "SVM RMSE on Testing Data: 2.71996183597951"

print(paste("SVM R-squared on Testing Data:", svm\_r\_squared))

## [1] "SVM R-squared on Testing Data: 0.986162460997585"

# Gradient Boosting  
gb\_model <- gbm(ex ~ Age + Gender, data = train\_data, distribution = "gaussian")  
gb\_predictions <- predict(gb\_model, newdata = test\_data, n.trees = 100) # Adjust the number of trees as needed  
gb\_rmse <- sqrt(mean((test\_data$ex - gb\_predictions)^2))  
gb\_r\_squared <- cor(test\_data$ex, gb\_predictions)^2  
print(paste("Gradient Boosting RMSE on Testing Data:", gb\_rmse))

## [1] "Gradient Boosting RMSE on Testing Data: 2.95589142241367"

print(paste("Gradient Boosting R-squared on Testing Data:", gb\_r\_squared))

## [1] "Gradient Boosting R-squared on Testing Data: 0.983638203065615"