# Assignment 1 - Task 1: Anscombe’s Quartet

Anscombe’s Quartet, introduced by statistician Francis Anscombe in 1973, illustrates the importance of visualizing data in statistical analysis. Although the four datasets (commonly called Dataset 1 through Dataset 4) have nearly identical statistical summaries—including means, variances, correlation coefficients, regression equations, and R² values—their scatterplots reveal very different underlying patterns.  
  
Dataset 1 shows a classic linear trend, where the fitted regression line appropriately describes the relationship between x and y. Dataset 2, while sharing the same regression line statistically, displays a curved nonlinear pattern that the simple linear model cannot capture. Dataset 3 appears mostly linear, but a single influential outlier dramatically shifts the regression line. Dataset 4 is especially striking: most of the data points are vertically aligned with little variation, and the regression line is determined almost entirely by one extreme point.   
  
This exercise demonstrates that relying only on statistical summaries or regression coefficients can be misleading. Two datasets with the same descriptive statistics may represent completely different relationships. Therefore, visualizations are essential in exploratory data analysis, as they reveal patterns, anomalies, and influential points that are invisible from numerical summaries alone. The lesson of Anscombe’s Quartet is simple yet powerful: always visualize your data before drawing conclusions.

## R Code

# Load Anscombe's dataset  
data(anscombe)  
  
# Fit regression models  
lm1 <- lm(y1 ~ x1, data = anscombe)  
lm2 <- lm(y2 ~ x2, data = anscombe)  
lm3 <- lm(y3 ~ x3, data = anscombe)  
lm4 <- lm(y4 ~ x4, data = anscombe)  
  
# Plot the four datasets with regression lines  
op <- par(mfrow = c(2,2)) # 2x2 grid  
for(i in 1:4) {  
 x <- anscombe[[paste0("x", i)]]  
 y <- anscombe[[paste0("y", i)]]  
 plot(x, y, pch = 19, col = "red",  
 main = paste("Dataset", i),  
 xlim = c(3, 19), ylim = c(3, 13))  
 abline(lm(y ~ x), col = "blue", lwd = 2)  
}  
par(op)

## Plot

