

Homework-3

Group 12

Introduction:

This report explores two distinct statistical problems:

1. **Bivariate Distribution Analysis:** We generate and analyze unimodal and bimodal bivariate normal distributions using kernel density estimation. The contours of these density estimates help in understanding the distribution of data.
2. **Monte Carlo Approximation of Pi:** We approximate the value of π using Monte Carlo integration and study the variability in integral estimation.

Bivariate Distribution Analysis

Unimodal Bivariate Normal Distribution

A bivariate normal (BVN) distribution with mean (0,0) and identity covariance matrix is generated. We use kernel density estimation (KDE) to estimate its density and visualize it using a contour plot.

```
library(MASS)
```

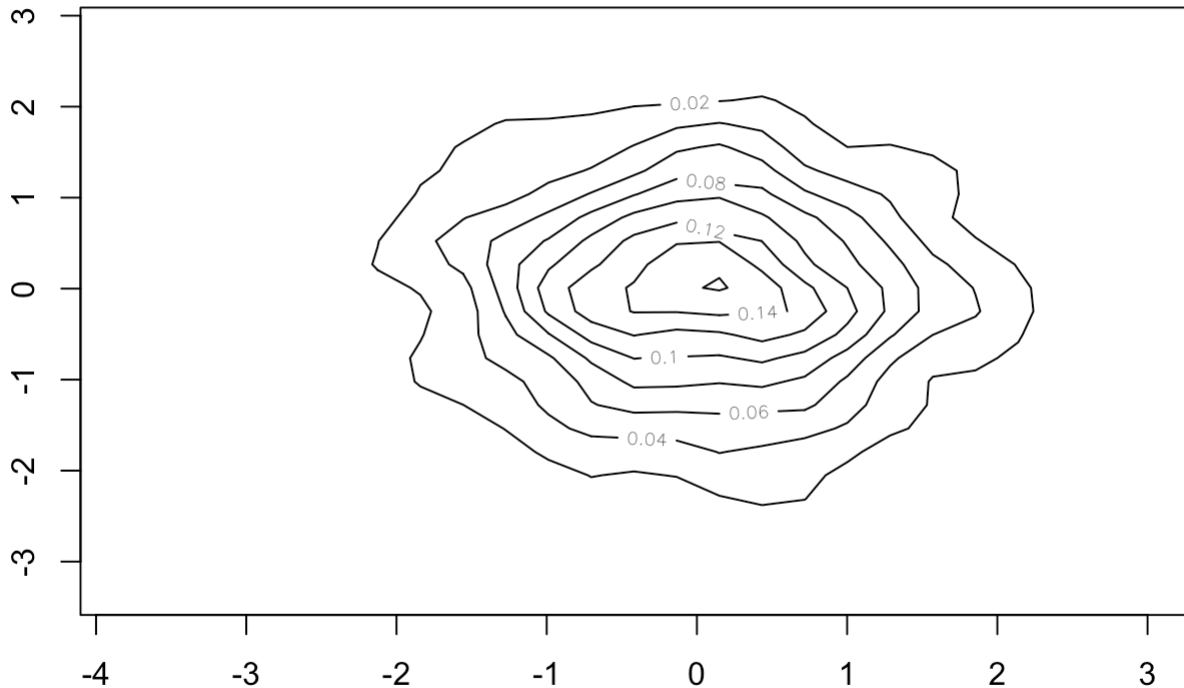
Generating data from a bivariate unimodal distribution

```
data1 <- mvrnorm(n = 1000, mu = c(0,0), Sigma = matrix(c(1,0,0,1), nrow = 2))
```

Kernel density estimation and contour plot

```
kde.data1 <- kde2d(data1[,1], data1[,2])  
contour(kde.data1, main = "Density Contour Plot - Unimodal")
```

Density Contour Plot - Unimodal



The contour plot consists of concentric curves, indicating that the data is unimodal.

Bimodal Bivariate Normal Distribution

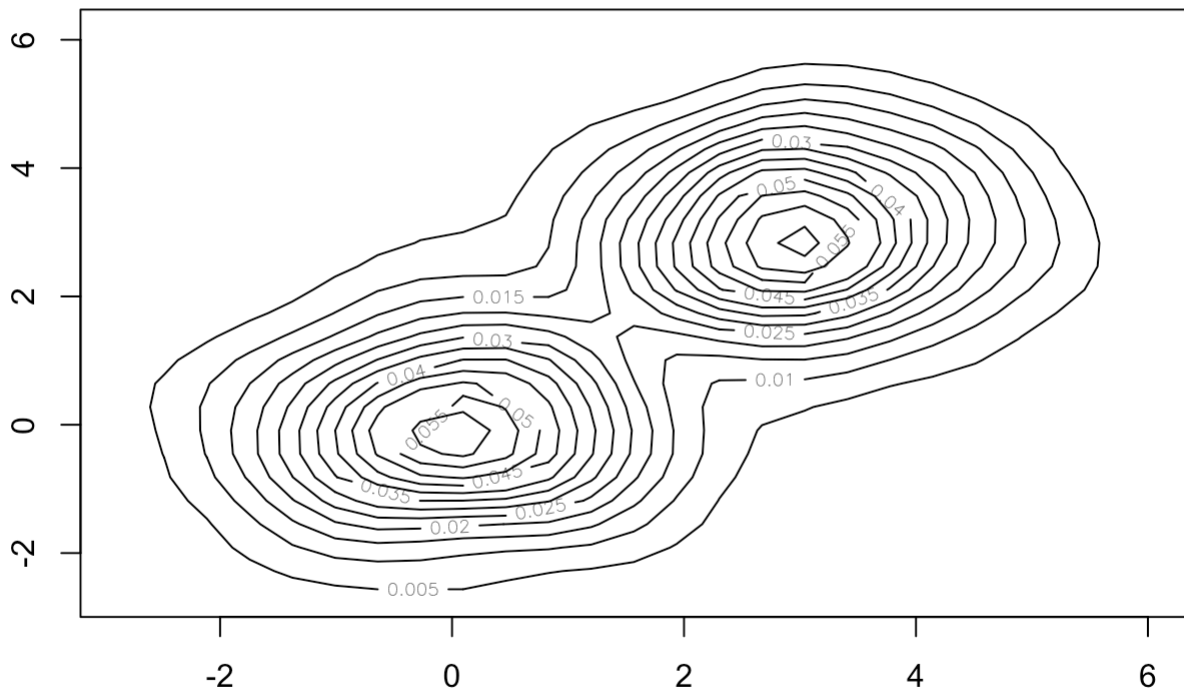
A mixture of two BVN distributions is created: one centered at (0,0) and another at (3,3), each with an identity covariance matrix. KDE is used to estimate the density, followed by contour visualization.

```
# Generating data from a bivariate bimodal distribution
mvrnorm.mixture <- function(n) {
  data <- matrix(0, nrow = n, ncol = 2)
  for(i in 1:n) {
    u <- runif(1)
    if(u < 0.5) {
      data[i, ] <- mvrnorm(1, mu = c(0,0),
                           Sigma = matrix(c(1,0,0,1), nrow = 2))
    } else {
      data[i, ] <- mvrnorm(1, mu = c(3,3),
                           Sigma = matrix(c(1,0,0,1), nrow = 2))
    }
  }
  return(data)
}

data2 <- mvrnorm.mixture(n = 1000)
```

```
# Kernel density estimation and contour plot
kde.data2 <- kde2d(data2[,1], data2[,2])
contour(kde.data2, main = "Density Contour Plot - Bimodal")
```

Density Contour Plot - Bimodal



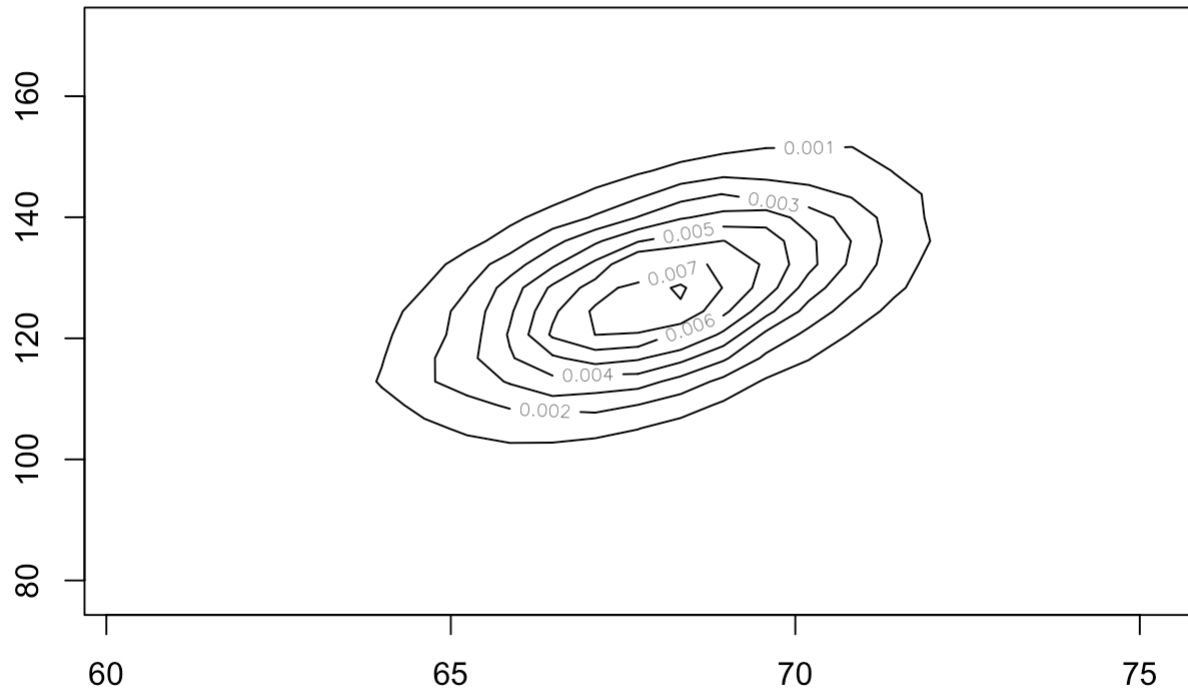
The contour plot exhibits two separate sets of concentric curves, confirming the bimodal nature of the dataset.

Real-World Data Analysis

We analyze a real-world dataset (SOCR-HeightWeight.csv) to estimate its density using KDE.

```
# Mode analysis of real-world data
df <- read.csv("SOCR-HeightWeight.csv")
data <- df[,2:3]
kde.data <- kde2d(data[,1], data[,2])
contour(kde.data, main = "Density Contour Plot - Real Data")
```

Density Contour Plot - Real Data



Conclusion:

The contour plot provides insights into the distribution of real-world height-weight data, helping identify the presence of single or multiple modes.

Monte Carlo Estimation of Pi

Approximation of π using Monte Carlo Integration

The integral formulation provides an estimate of the error in approximating π by 355/113. We estimate the integral using Monte Carlo sampling.

```
# Monte Carlo estimation function
est.int <- function(n) {
  u <- runif(n)
  rtn <- mean(u^8 * (1-u)^8 * (25+816*u^2) / 3164 / (1+u^2))
  return(rtn)
}
```

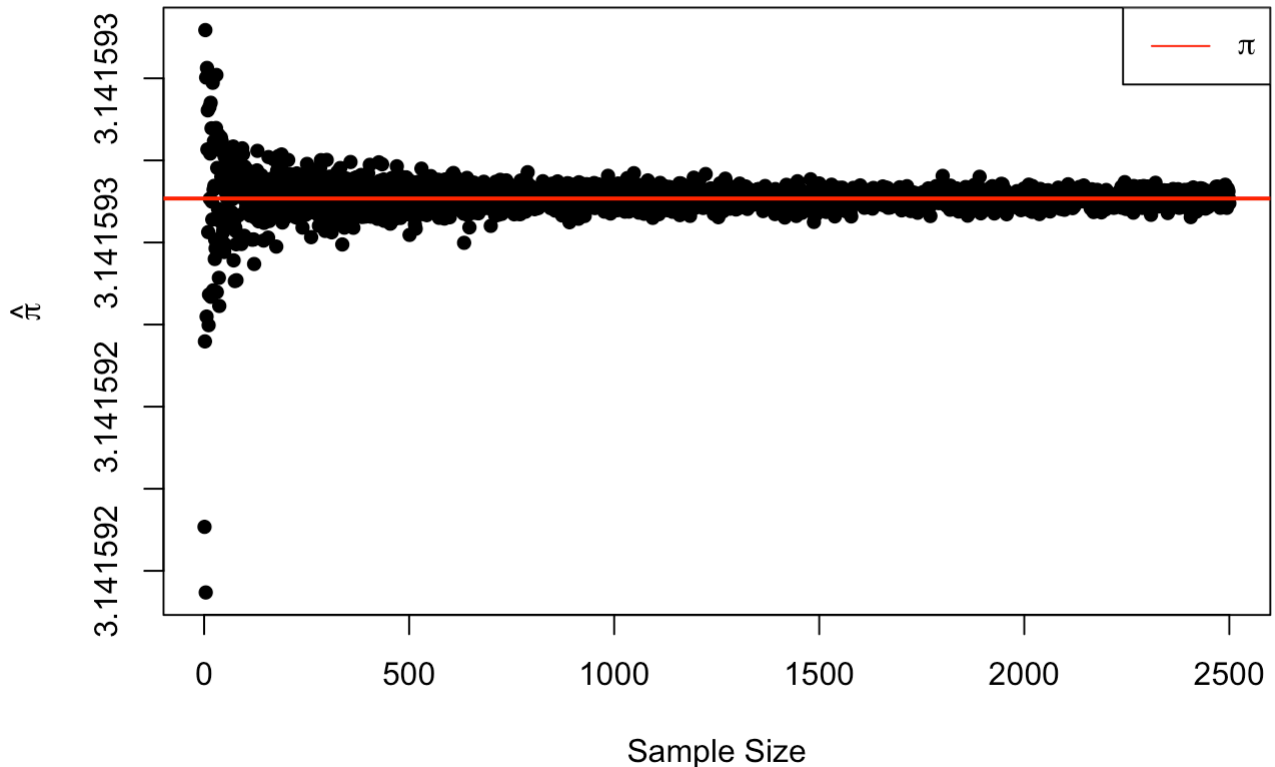
Computing estimated values of pi

```

pi.hat <- numeric(2500)
for(i in 1:2500){
  pi.hat[i] <- 355/113 - est.int(i)
}

plot(pi.hat, pch = 16, xlab = "Sample Size", ylab = expression(hat(pi)))
abline(h = pi, col = "red", lwd = 2)
legend("topright", col = "red", lty = 1, legend = expression(pi))

```



As the sample size increases, the Monte Carlo estimate converges to the true value of π , demonstrating the effectiveness of this method.

Variability in Monte Carlo Estimates

We analyze how variability in the integral estimate changes with different scaling factors.

```

# Variability of the integral function
f <- function(n, delta)
{
  u <- runif(n)
  integrand <- u^8 * (1-u)^8 * (25+816*u^2) / 3164 / (1+u^2)
  int.hat <- mean(integrand)
  if (n > 1) {
    rtn <- n^delta * int.hat / sqrt(var(integrand))
  } else {
    rtn <- int.hat # For n = 1, we just return the mean of the integrand
  }
}

```

```

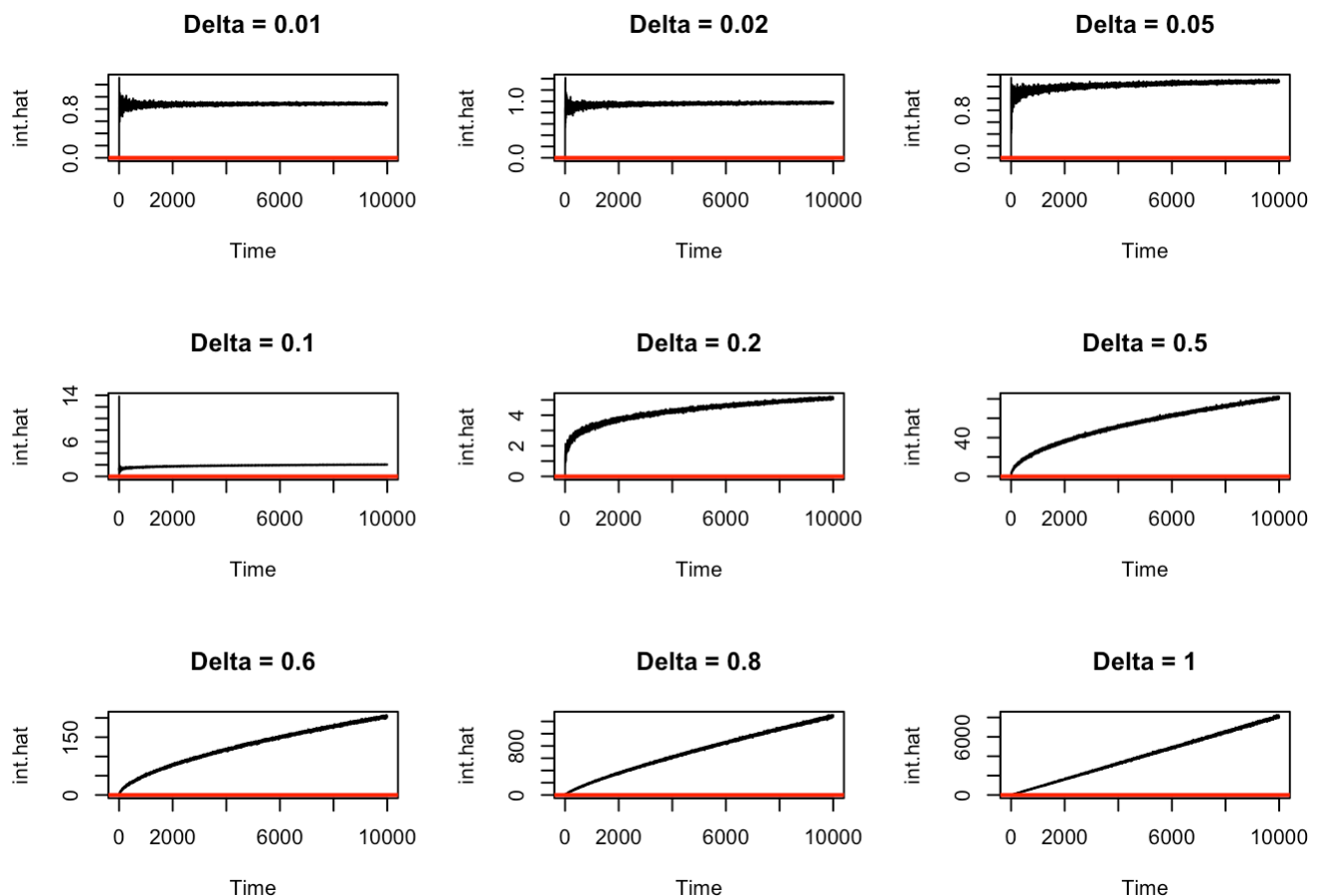
}
return(rtn)
}

par(mfrow = c(3,3))

delta <- c(0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 0.6, 0.8, 1 )
for(i in 1:length(delta))
{
  int.hat <- numeric(10000)
  for(n in 1:10000)
    int.hat[n] <- f(n, delta[i])

  plot.ts(int.hat, pch = 16, main = paste("Delta =", delta[i]))
  abline(h = 0, col = "red", lwd = 2)
}

```



Conclusion:

The results indicate that the scaling factor significantly influences the variability of the integral estimate. While the rate of convergence is subtle and challenging to observe, even with relatively large sample sizes (e.g., 10,000), the curve still converges for delta values between 0 and 0.5. However, for delta values greater than 0.5, the curve exhibits clear divergence, as evidenced by increased fluctuations in the estimates. This observation aligns with the central limit theorem, which suggests that the integral estimate should converge for delta values less than 0.5.

Summary

- The bivariate normal density estimation successfully identified unimodal and bimodal distributions.
- Real-world data contour plots help visualize multimodality.
- Monte Carlo estimation of π demonstrates convergence to the true value as sample size increases.
- Scaling factors impact the variability of Monte Carlo integral estimates.