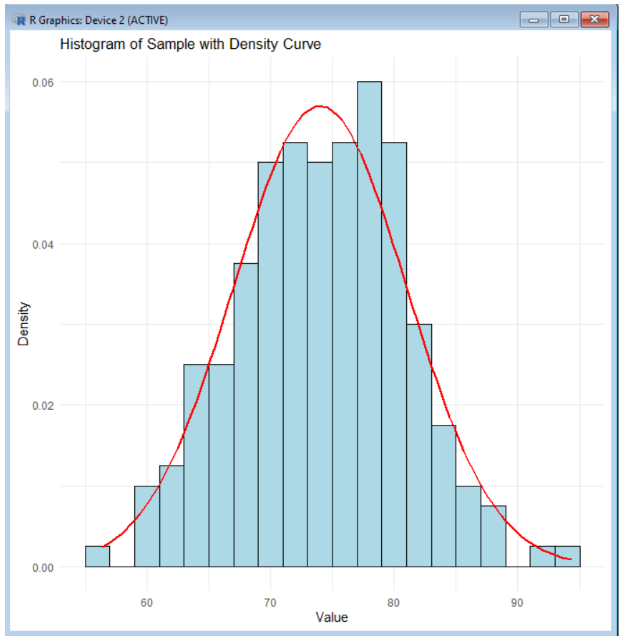
1. On a recent English test, the scores were normally distributed with a mean of 74 and a standard deviation of 7.

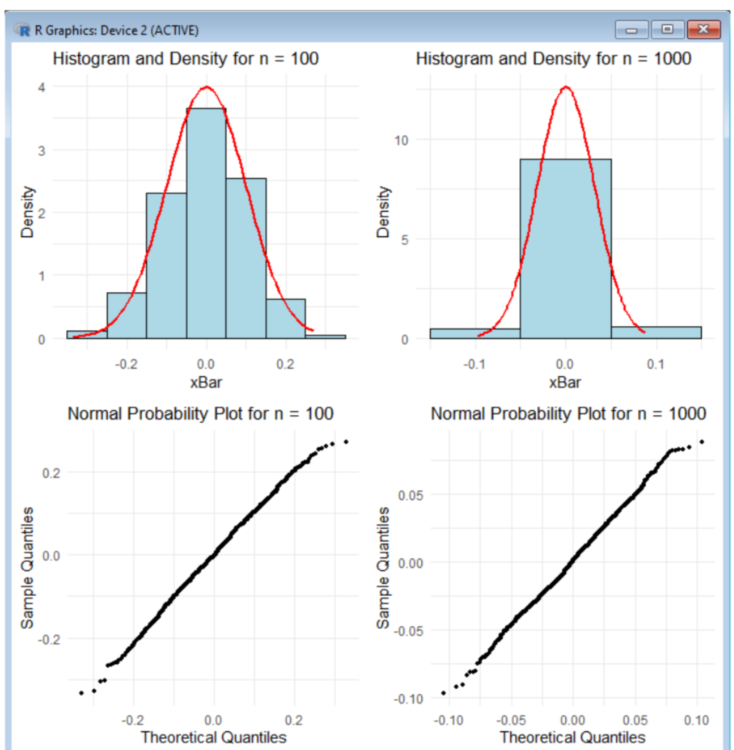
```
> # Set parameters
> mean <- 74
> sd <- 7
>
> # a) What proportion of the class would be expected to score less than 50 points?
> prop_less_than_50 <- pnorm(50, mean, sd)
> prop_less_than_50
[1] 0.0003033834
>
> # b) What proportion of the class would be expected to score more than 90 points?
> prop_more_than_90 <- 1 - pnorm(90, mean, sd)
> prop_more_than_90
[1] 0.01113549
>
> # c) What proportion of the class would be expected to score between 60 and 80 points
> prop_between_60_and_80 <- pnorm(80, mean, sd) - pnorm(60, mean, sd)
> prop_between_60_and_80
[1] 0.7815669
>
> # d) What is the third quartile of the test score?
> third_quartile <- qnorm(0.75, mean, sd)
> third_quartile
[1] 78.72143
>
> # e) What is the 90th quantile of the test score?
> quantile_90th <- qnorm(0.90, mean, sd)
> quantile_90th
[1] 82.97086
> |
```

2. Generate a sample of 200 observations from $N(\mu = 74, \sigma2 = 49)$ (do not need to show the 200 values). Draw the histogram of the sample with the density curve of $N(\mu = 74, \sigma2 = 49)$.

```
> # Load required packages
> library(ggplot2)
>
> # Set parameters
> mean <- 74
> sd <- sqrt(49) # Since variance is 49, the standard deviation is the square root of 49, which is 7.
> n <- 200
>
> # Generate a sample of 200 observations
> sample_data <- rnorm(n, mean, sd)
>
> # Create a data frame for plotting
> sample_data_df <- data.frame(value = sample_data)
>
> # Create histogram with density curve
> ggplot(sample_data_df, aes(x = value)) +
+   geom_histogram(aes(y = ..density..), binwidth = 2, fill = "lightblue", color = "black") +
+   stat_function(fun = dnorm, args = list(mean = mean, sd = sd), size = 1, color = "red") +
+   labs(title = "Histogram of Sample with Density Curve",
+        x = "Value",
+        y = "Density") +
+   theme_minimal()
> |
```



3. Assume $X1, X2, X3, \cdots, Xn$ is a random sample from the standard normal distribution. Then the sample mean $n-1 \sum Xi\ n\ i=1$ is also normal distributed with mean unchanged and standard deviation $n-1/2$. Next, you need to generate graphical evidence to support this statement. The following sample R code draws the histogram and density of the sample mean for n=10. Make sure to change the value of n for a different sample size



```r
> # Load required packages
> library(ggplot2)
> library(gridExtra)
>
> # Function to generate xBar values for given sample size
> generate_xBar <- function(n, sampleN = 1000) {
+   xBar <- array(NA, sampleN)
+   for (i in 1:sampleN) {
+     aSample <- rnorm(n, 0, 1)
+     xBar[i] <- mean(aSample)
+   }
+   return(xBar)
+ }
>
> # Function to plot histogram and density plot
> hist_density_plot <- function(xBar, n) {
+   df <- data.frame(xBar = xBar)
+   p <- ggplot(df, aes(x = xBar)) +
+     geom_histogram(aes(y = ..density..), binwidth = 0.1, fill = "lightblue", color = "black") +
+     stat_function(fun = dnorm, args = list(mean = 0, sd = 1 / sqrt(n)), size = 1, color = "red") +
+     labs(title = paste("Histogram and Density for n =", n),
+          x = "xBar",
+          y = "Density") +
+     theme_minimal()
+   return(p)
+ }
>
> # Function to plot normal probability plot
> normal_prob_plot <- function(xBar, n) {
+   df <- data.frame(xBar = xBar)
+   p <- ggplot(df, aes(sample = xBar)) +
+     stat_qq(distribution = qnorm, dparams = list(mean = 0, sd = 1 / sqrt(n)), size = 1) +
+     labs(title = paste("Normal Probability Plot for n =", n),
+          x = "Theoretical Quantiles",
+          y = "Sample Quantiles") +
+     theme_minimal()
+   return(p)
+ }
>
> # Generate xBar values for n = 100 and n = 1000
> xBar_100 <- generate_xBar(100)
> xBar_1000 <- generate_xBar(1000)
>
> # Plot histograms and density plots for n = 100 and n = 1000
> hist_density_100 <- hist_density_plot(xBar_100, 100)
> hist_density_1000 <- hist_density_plot(xBar_1000, 1000)
>
> # Plot normal probability plots for n = 100 and n = 1000
> normal_prob_100 <- normal_prob_plot(xBar_100, 100)
> normal_prob_1000 <- normal_prob_plot(xBar_1000, 1000)
>
> # Combine and display plots
> grid.arrange(hist_density_100, hist_density_1000, normal_prob_100, normal_prob_1000, ncol = 2)
>
```