Problem-Set-1 for POLI 271

1. Univariate displays & sampling distributions

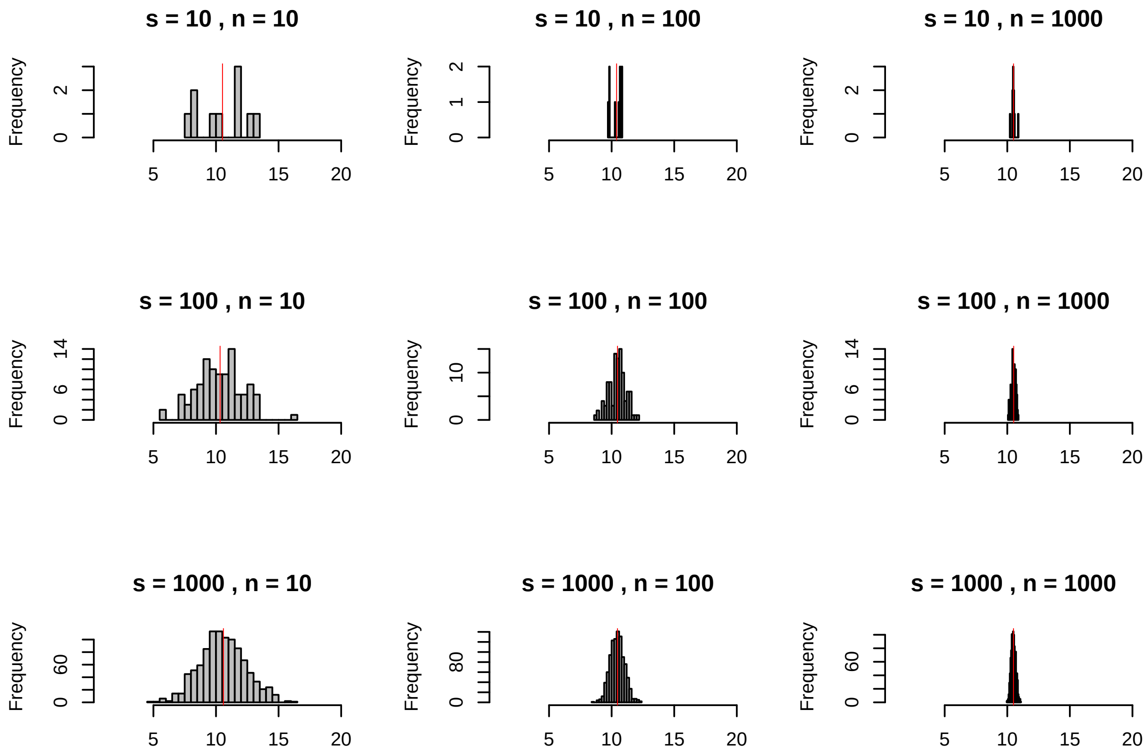
Problem a

```r
s_values <- c(10, 100, 1000)
n_values <- c(10, 100, 1000)

par(mfrow = c(3, 3))
for (s in s_values) {
  for (n in n_values) {
    sample_means <- replicate(s, mean(sample(1:20, n, replace = TRUE)))

    hist(sample_means,
         breaks = 20,
         col = "grey",
         border = "black",
         main = paste("s =", s, ", n =", n),
         xlab = "",
         xlim = c(1, 20))

    abline(v = mean(sample_means), col = "red", lwd = 0.5)
  }
}
```



Problem b

The histograms shows that when increasing the sample size reduces variability in sample means. I set it to 1:20 so it's getting closer to 10. And increasing the number of samples makes it look like normal distribution. The key assumption here is what CLT describes: the distribution of a normalized version of the sample mean converges to a standard normal distribution.

2. Monte Carlo integration

```
func = function(x){exp(-x)*sin(x)}
result = integrate(func, lower = 2, upper = 5)
print(result)
```

```
0.03564528 with absolute error < 8.3e-16
```

3. Systematic and stochastic components

Problem a

Systematic Component: $y_i = 1 + 0.5x_{i1} - 2.2x_{i2} + x_{i3}$ Stochastic Component: $\epsilon_i \sim N(\mu = 0, \sigma^2 = 1.5)$

Problem b

Part I. The dimensions of $\mathbf{X}$ is denoted as n is 2.

```
data = read.csv("xmat.csv")
print(dim(data))
```

```
[1] 1000    3
```

```
head(data, 10)
```

```
          X1 X2          X3
1  -4.8200977  1  1.54137265
2   2.5755430  0  1.25892647
3   0.3326820  1 -0.06933333
4  -1.1534374  1  0.07761559
5   2.0563184  0 -1.19921600
6   0.1335086  0  0.25054654
7   1.6025580  1 -0.41074599
8  -1.4491007  1  2.31999656
9   1.2676561  1 -0.80968744
10  1.0784026  1 -0.31089005
```

Part II.

```
set.seed(10825)
# Why not 42 and 3407
coefficient_0 = 1
coefficient_1 = 0.5
coefficient_2 = -2.2
coefficient_3 = 1
x_1 = data$X1
x_2 = data$X2
```

```r
x_3 = data$X3
e = rnorm(n = nrow(data), mean = 0, sd = sqrt(1.5))
y = coefficient_0 +
  coefficient_1*x_1 + coefficient_2*x_2 + coefficient_3*x_3 + e

linear = lm(y ~ x_1 + x_2 + x_3)
summary((linear))
```

```
Call:
lm(formula = y ~ x_1 + x_2 + x_3)

Residuals:
    Min      1Q  Median      3Q     Max
-3.5330 -0.8196  0.0124  0.8168  4.5651

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.06651    0.05084   20.98   <2e-16 ***
x_1          0.48024    0.01925   24.95   <2e-16 ***
x_2         -2.26451    0.07852  -28.84   <2e-16 ***
x_3          0.95040    0.03822   24.86   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.225 on 996 degrees of freedom
Multiple R-squared:  0.6792,    Adjusted R-squared:  0.6782
F-statistic: 702.9 on 3 and 996 DF,  p-value: < 2.2e-16
```

```r
# Beautiful p-value
```

4. OLS in matrix form