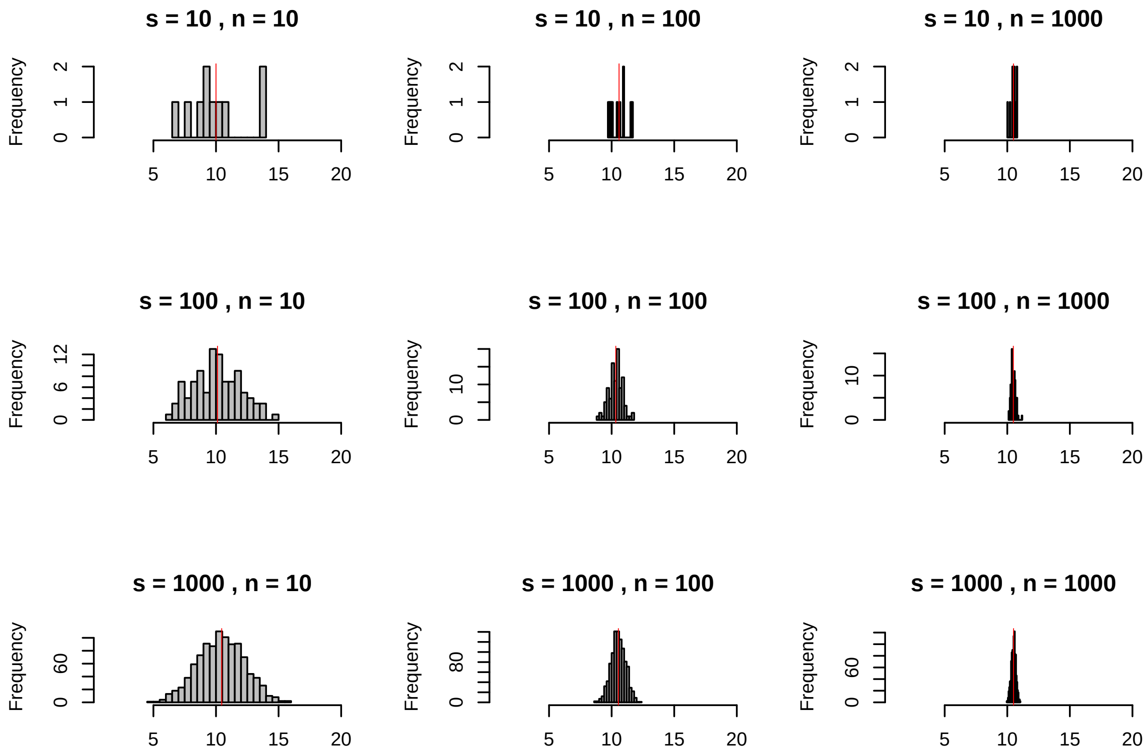
Problem-Set-1 for POLI 271

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

```r
library(haven)
data_2 = read_dta("coxappend.dta")
attributes(data_2)
```

```
$class
[1] "tbl_df"     "tbl"        "data.frame"
```

```
$row.names
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
[51] 51 52 53 54

$names
 [1] "var12"    "drop"     "year"     "enpv"     "enps"     "eneth"
 [7] "ml"       "upper"    "enpres"   "proximit" "lnml"     "lmleneth"
[13] "smdp"     "smdpeth"  "multi"    "enpvlml"  "enpvUpp"  "multiV"
[19] "enpvQ"    "enpvmult" "enpvsmdp" "proxpres" "drop2"
```

```r
head(data_2, 10)
```

```
# A tibble: 10 x 23
   var12      drop  year  enpv  enps eneth    ml upper enpres proximit  lnml
   <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>    <dbl> <dbl>
 1 ARGENTINA     0  1985  3.37  2.37  1.34   9   0       2.51    0.550  2.20
 2 AUSTRALIA     0  1984  2.79  2.38  1.11   1   0       0       0      0
 3 AUSTRIA       0  1986  2.72  2.63  1.01  30   0.115   2.27    0.800  3.40
 4 BAHAMAS       0  1987  2.11  1.96  1.34   1   0       0       0      0
 5 BARBADOS      0  1986  1.93  1.25  1.50   1   0       0       0      0
 6 BELGIUM       0  1985  8.13  7.01  2.35   8   0.401   0       0      2.08
 7 BELIZE        0  1984  2.06  1.60  3.46   1   0       0       0      0
 8 BOLIVIA       1  1985  4.58  4.32  3.77  17.5 0       4.58    1      2.86
 9 BOTSWANA      0  1984  1.96  1.35  1.11   1   0       0       0      0
10 BRAZIL        0  1990  9.68  8.69  2.22  30   0       5.69    0.630  3.40
# i 12 more variables: lmleneth <dbl>, smdp <dbl>, smdpeth <dbl>, multi <dbl>,
#   enpvlml <dbl>, enpvUpp <dbl>, multiV <dbl>, enpvQ <dbl>, enpvmult <dbl>,
#   enpvsmdp <dbl>, proxpres <dbl>, drop2 <dbl>
```

Problem a

```r
ols_regression = function(y, X) {
  X = cbind(1, X)

  # beta_hat = (X'X)^(-1) X'y
  beta_hat = solve(t(X) %*% X) %*% t(X) %*% y

  # residuals = y - X * beta_hat
```

```r
  residuals = y - X %*% beta_hat

  # sigma^2 = RSS / (n - p)
  n = nrow(X)
  p = ncol(X)
  sigma2 = sum(residuals^2) / (n - p)

  # SE(beta) = sqrt(diag(sigma^2 * (X'X)^(-1)))
  se_beta = sqrt(diag(sigma2 * solve(t(X) %*% X)))

  return(list(
    coefficients = beta_hat,
    standard_errors = se_beta,
    residuals = residuals
  ))
}

y = data_2$enps
X = data.frame(
  eneth = data_2$eneth,
  log_ml = log(data_2$ml),
  interaction = data_2$eneth * log(data_2$ml)
)

results = ols_regression(y, as.matrix(X))

print("OLS Results:")
```

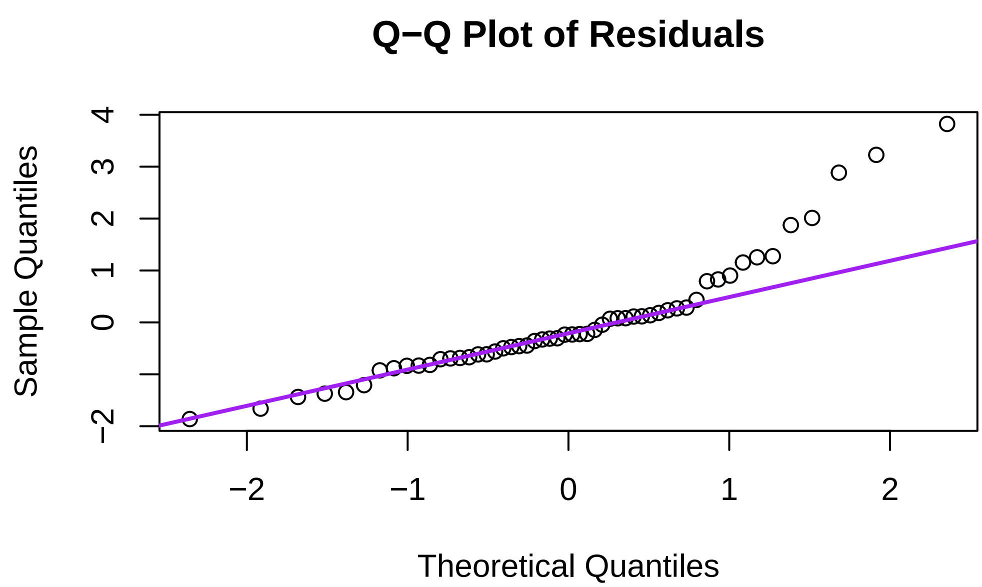
```
[1] "OLS Results:"
```

```r
coefficients_table = data.frame(
  Coefficient = results$coefficients,
  Std_Error = results$standard_errors
)
rownames(coefficients_table) = c("Intercept", "eneth", "log_ml", "interaction")
print(coefficients_table)
```

```
            Coefficient Std_Error
Intercept     2.6713673 0.6072149
eneth        -0.3619712 0.3486305
log_ml       -0.1911175 0.2967357
interaction   0.4833255 0.1805094
```

Problem b

```
qq_plot = function(residuals) {
  qqnorm(residuals, main = "Q-Q Plot of Residuals")
  qqline(residuals, col = "purple", lwd = 2)
}

residuals = results$residuals
qq_plot(residuals)
```



- The Q-Q plot shows that the residuals mostly align with the reference line.
- The deviations at the tails ndicates outliers in the residual distribution.
- Overall, the residuals are close to a normal distribution.

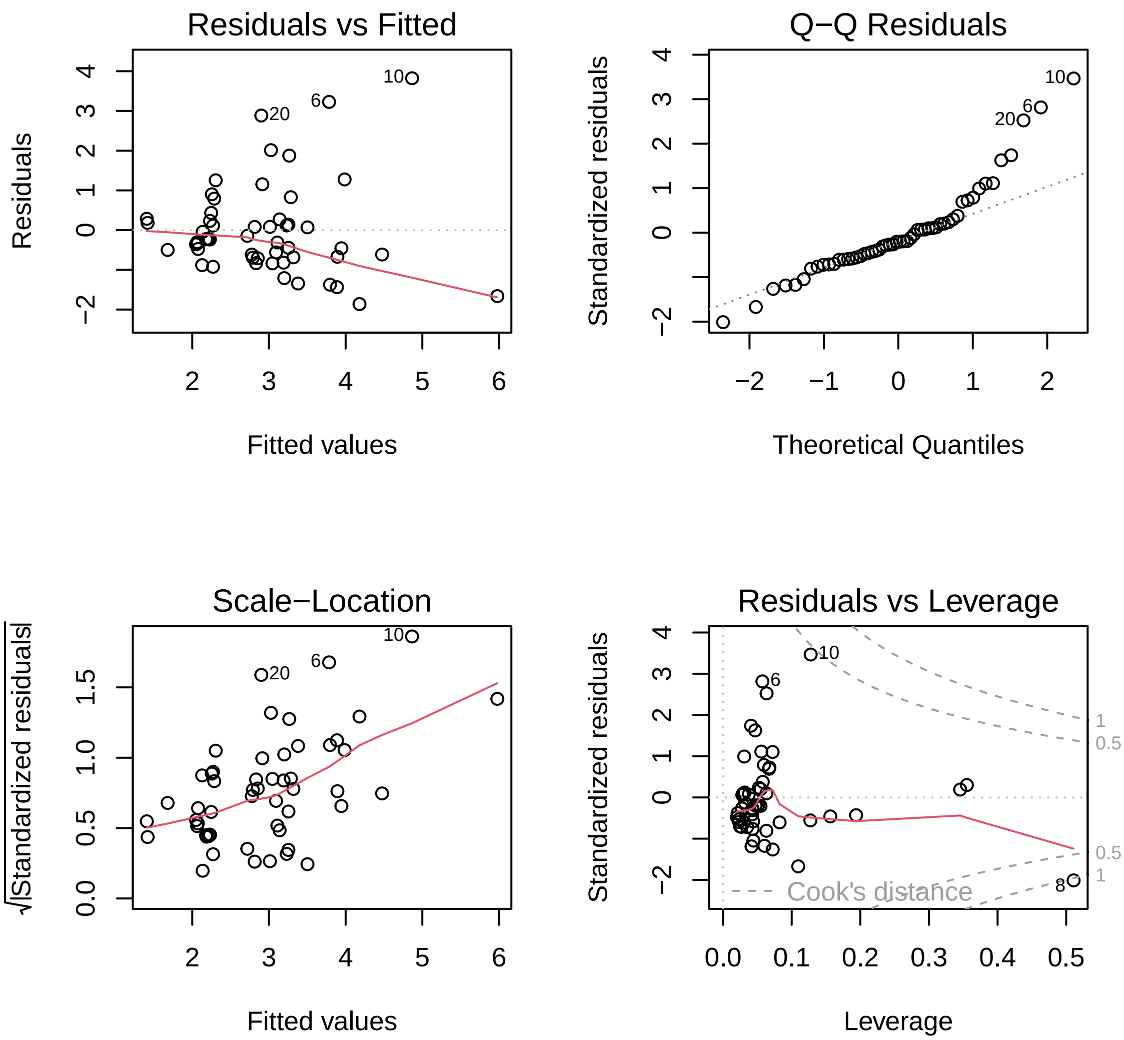
Problem c

```
lm_model = lm(enps ~ eneth * log(ml), data = data_2)
summary(lm_model)
```

```
Call:
lm(formula = enps ~ eneth * log(ml), data = data_2)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.8627 -0.6818 -0.2346  0.2605  3.8235 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)     2.6714     0.6072   4.399 5.69e-05 ***
eneth          -0.3620     0.3486  -1.038    0.304    
log(ml)        -0.1911     0.2967  -0.644    0.522    
eneth:log(ml)   0.4833     0.1805   2.678    0.010 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.181 on 50 degrees of freedom
Multiple R-squared:  0.3629,    Adjusted R-squared:  0.3247 
F-statistic: 9.493 on 3 and 50 DF,  p-value: 4.541e-05
```

Problem d

```r
par(mfrow = c(2, 2))
plot(lm_model)
```



The OLS model gives us some useful insights, but it's not a perfect fit for the data. The patterns in the plots suggest the relationships might not be completely linear, the spread of the errors isn't consistent, and a few points have a big influence on the results.

Appendix

I certify that we did not use any LLM or generative AI tool in this assignment.