PS01 by Emily Han

2025-01-14

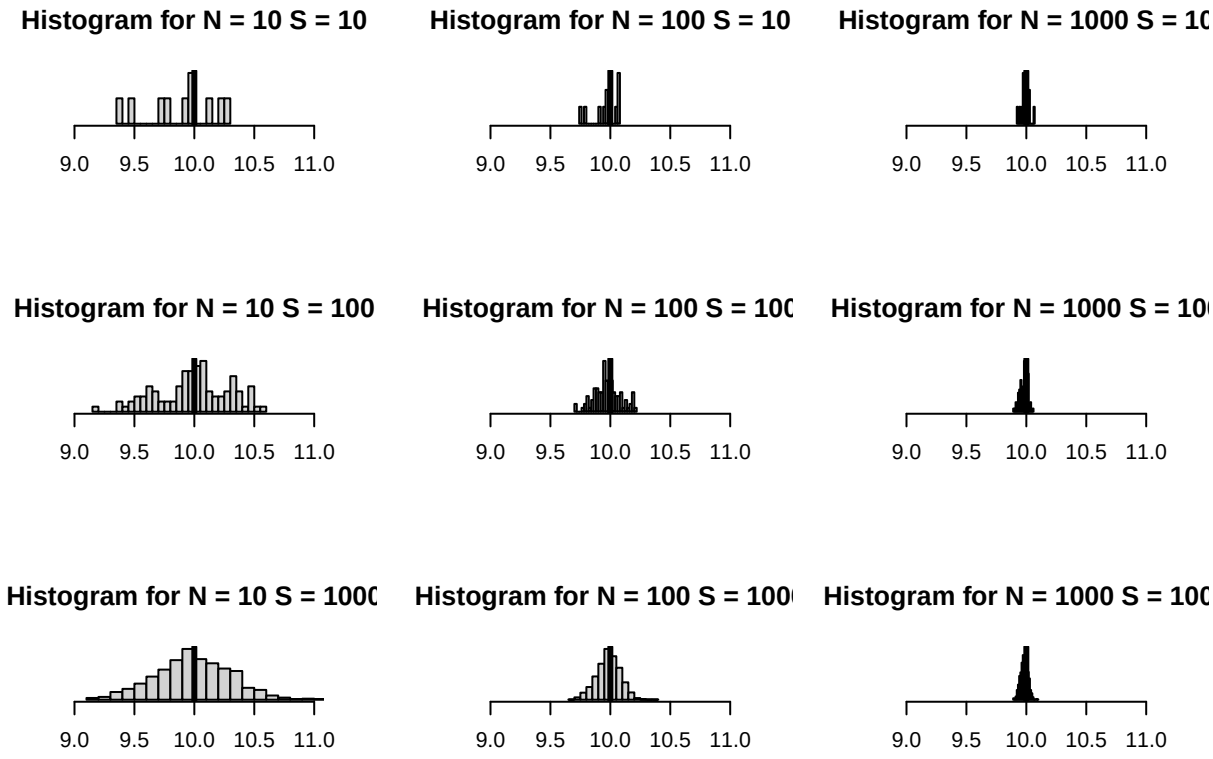
1. Univariate displays & sampling distributions

Let n denote the size of a simple random sample taken from a large population and let s be the number of times samples of size n are drawn.

1a: Plotting the histograms

Write an R program that will replicate Figure 1.2 from the Ward & Ahlquist text. Create a matrix of nine such histograms which illustrate the effect of changing $s \in 10, 100, 1000$ and $n \in 10, 100, 1000$. Put the horizontal axis on the same scale for all plots. HINT: try constructing loops using for, replicate, and sample.

```
# Define parameters
population <- rnorm(10000, mean = 10, sd =1)
n <- c(10, 100, 1000)
iteration <- c(10, 100, 1000)


#For creating matrix of graphs
par(mfrow = c(3,3))

for (i in iteration) {
  for (j in n) {
    sam_mean <- replicate(i, mean(sample(population, j)))
    title = paste("Histogram for N =", j, "S =", i)
    plt = hist(sam_mean, breaks = 20, xlim = c(9, 11),
               yaxt = "n", xlab = NULL, ylab = NULL, main = title)
    abline(v = 10, lwd = 3)
    plt
  }
}
```



1b: Interpretation

What does this matrix of histograms say about the efficiency of sampling when we consider the number of groups versus the number of units within a group? What is the key assumption or attribute of the situation here that underpins this lesson?

The matrix of histogram suggests that sampling with higher amount more units (size) within a group is more efficient than sampling large number of groups. The key attribute of this situation suggest having more units within a group bring the mean of the samples closer to the true population more efficiently due to less variability in the data.

---

2. Monte Carlo integration

Approximating the integral using simulation & checking the answering

$E[f(X)] = \frac{1}{b-a}\int_a^b f(x)dx$

$\int_a^b f(x)dx = E[f(X)] * (b-a)$

```
# Define the function
f <- function(x) {exp(-x) * sin(x)}


# Taking large sample of values between 2 to 5
sam_5_2 <- runif(100000,2,5)
```

```r
# Save the results after plugging in the values
result <- f(sam_5_2)

# Multiple with 3(mean of 2-5)
approximate_answer <- 3 * mean(result)

print(approximate_answer)
```

```
## [1] 0.03534424
```

```r
integrate(f, 2, 5)
```

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

---

3. Systematic and stochastic components

3a. Rephrasing the model in terms of its systematic and stochastic components.

Rephrase this model in terms of its systematic and stochastic components.

$Y_i = \beta_0 + \beta_i X_i + \epsilon_i$

Systematic Component
$\theta_i = 1 + 0.5x_{i1} - 2.2x_{i2} + x_{i3}$

Stochastic Component

$Y_i \sim f_N(1 + 0.5x_{i1} - 2.2x_{i2} + x_{i3}, \epsilon \sim \mu = 0, \sigma^2 = 1.5)$

b. Download `xmat.csv` from the course canvas site and load it into your `R` session.

i. What are the dimensions of the **X** matrix you just downloaded? Call this number `n`.

n = 1000 x 3

ii. Type `set.seed(10825)`. Then combine the **X** matrix you just downloaded with the linear model you described in part (a) to generate `n` simulated y values. Becareful with the constant! Then regress these simulated y values on **X**. Display the results in a regression table.

```r
set.seed(10825)
```

```r
dat <- read.csv("./data/xmat.csv")

Xi <- data.matrix(dat)

e <- rnorm(1000, mean = 0, sd = sqrt(1.5))

yi <- 0.5*Xi[,1] - 2.2*Xi[,2] + Xi[,3] + e + 1


regress <- lm(yi ~ Xi[,1] + Xi[,2] + Xi[,3])
summary(regress)
```

```
##
## Call:
## lm(formula = yi ~ Xi[, 1] + Xi[, 2] + Xi[, 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 ***
## Xi[, 1]      0.48024    0.01925   24.95   <2e-16 ***
## Xi[, 2]     -2.26451    0.07852  -28.84   <2e-16 ***
## Xi[, 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
```

---

4. OLS in matrix form

a & b

a. Write an R function to perform an OLS regression using matrices and vectors.

Your function should take take two inputs: a vector of the dependent variable and a matrix of explanatory variables. Do not use any pre-programmed functions. You will regress the effective number of legislative parties (enps) on the number of effective ethnic groups (eneth), the log of median district magnitude electoral district magnitude (ml), and the multiplicative interaction between the two. Report a table of the regression parameter estimates and their standard errors. Don't forget the constant!

b. Make a Normal quantile comparison plot of the residuals from your regression and describe what it says in fewer than three sentences.

Calculating $\hat{\beta}$

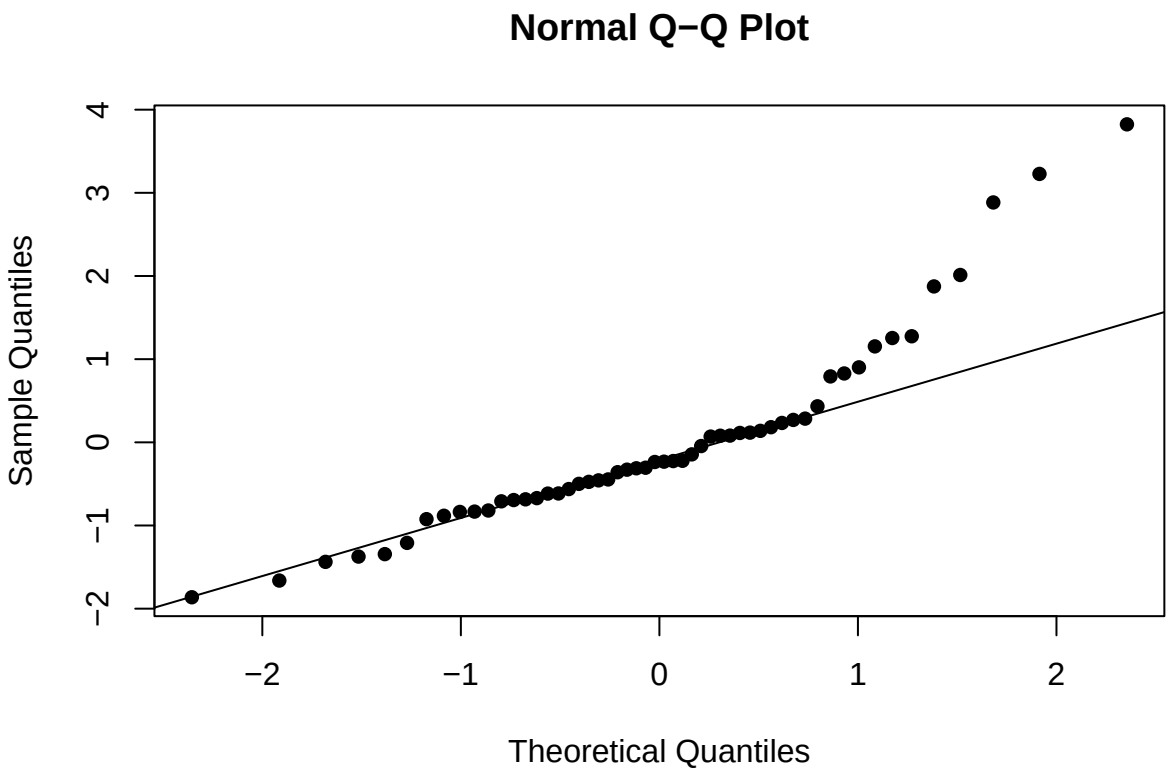
- $\hat{\beta} = (X'X)^{-1}X'Y$
- $Y =$ `enps`
- $X =$ `constant`, `eneth`, `ml`, `eneth*ml`

Calculating Standard Error

- $\hat{\epsilon} = y - X\hat{\beta}$
- $RSS = \hat{\epsilon}'\hat{\epsilon}$
- $Var(\hat{\beta}) = \frac{1}{(n-k)} * RSS * (X'X)^{-1}$

```r
library(foreign)

dat4 <- read.dta("./data/coxappend.dta")
dat4$log_ml <- log(dat4$ml)


#Create X and Y matrices
X4 <- as.matrix(cbind(1, dat4$eneth, dat4$log_ml, dat4$eneth * dat4$log_ml))
Y4 <- as.matrix((dat4$enps))
```

```r
ols <- function (X,Y) {

  # Calculate components to calculate beta_hat
  X_t <- t(X)
  X_inv <- solve(X_t %*% X)
  X_t_Y <- X_t %*% Y

  # Calculate beta_hat
  beta_hat <- X_inv %*% X_t_Y

  # Calculate vector of residuals
  error <- Y - (X %*% beta_hat)
  res_sum_sqr <- as.matrix(t(error) %*% error)


  # Calculate Variance-Covariance Matrix
  VCV <- 1/(54-4) * as.numeric(res_sum_sqr) * X_inv

  # Standard errors of the estimated coefficients
  StdErr <- sqrt(diag(VCV))


  # Label and organize results into a data frame
  ols_table <- as.data.frame(cbind(beta_hat, StdErr))
  colnames(ols_table) <- c('Estimate','Std. Error')
  rownames(ols_table) <- c('(Intercept)','eneth','ml','eneth:ml')


  ### b. Graphing Q-Q plot
  quartile_plot <- qqnorm(error, pch = 16)
  qqline(error, datax = FALSE, distribution = qnorm,
       probs = c(0.25, 0.75))


  return(ols_table)
  return(quartile_plot)
}


ols(X4, Y4)
```




```
##                Estimate Std. Error
## (Intercept)  2.6713673  0.6072149
## eneth       -0.3619712  0.3486305
## ml          -0.1911175  0.2967357
## eneth:ml     0.4833255  0.1805094
```

The normal quantile comparison plot shows an increasing trend of the residuals which means that the model performs badly

c. Perform the same regression using the `lm()` command.

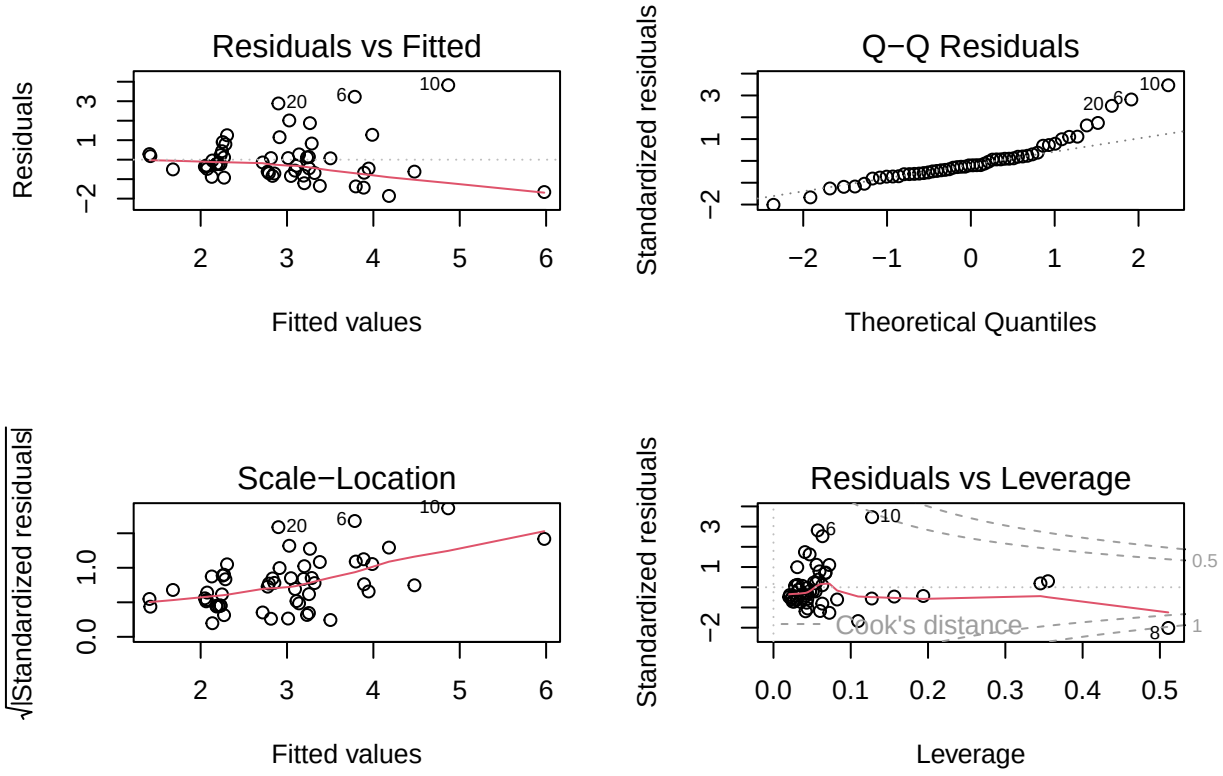
```
mod <- lm(enps ~ eneth + log_ml + eneth * log_ml, data = dat4)
summary(mod)
```

```
##
## Call:
## lm(formula = enps ~ eneth + log_ml + eneth * log_ml, data = dat4)
##
## 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
```

d. Report the Residual-v.-fitted, Scale-location, and Residual-v.-leverage diagnostic plots for this regression

```
par(mfrow = c(2,2))
plot(mod)
```



e. Based on these results, is this OLS regression an appropriate model for these data?

In the residual vs fitted graph, slight downward curve might suggest non-linearity. In addition, the upward trend in both QQ plot and Scale-Location plot suggests heteroscedasticity as the cluster of the data become more sparse as the value increases. In addition, there are some outliers with large residuals that might be problematic; for example observation 10. If the model still shows similar trends suggesting heteroscedasticity or non-linearity after addressing or removing the outliers, OLS regression might not be the appropriate model the data.

```
sessionInfo()
```

```
## R version 4.4.1 (2024-06-14)
## Platform: x86_64-apple-darwin20
```

```
## Running under: macOS 15.2
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] foreign_0.8-86
##
## loaded via a namespace (and not attached):
##  [1] compiler_4.4.1    fastmap_1.2.0     cli_3.6.3         tools_4.4.1
##  [5] htmltools_0.5.8.1 rstudioapi_0.16.0 yaml_2.3.10       rmarkdown_2.28
##  [9] highr_0.11        knitr_1.48        xfun_0.48         digest_0.6.37
## [13] rlang_1.1.4       evaluate_1.0.0
```