PSTAT220A Homework 2

Due 10/26/2025

Problem 1 (35 pt)

The dataset `Salaries` in library `carData` concerns the salaries for Professors in 2008-2009. Please read the `Salaries` documentation for a full description of the data.

a) Make a numerical and graphical summary of the data, commenting on any features that you find interesting

b) Fit a linear regression model with the salary as the response. Which variables, if any, are significantly associated with salary?

c) Compute LS estimates in R using the matrix solution to the least squares problem and confirm you get the same estimates as those in (b). To generate the covariates matrix it may help to use the `model.matrix` function.

d) What percentage of variation in the response is explained by the covariates? Explain whether you use the unadjusted or adjusted measure in your answer and why.

e) Which observation has the largest absolute residual (give the row number)?

f) Report separate 99% confidence intervals for the coefficients associated with `yrs.since.phd` and `yrs.service`.

g) Plot a 95% confidence region for the coefficients associated with `yrs.since.phd` and `yrs.service`. Comment on the resulting shape and why this makes sense.

h) Construct pointwise and simultaneous 95% confidence band for the prediction of future mean response and the prediction of future observations

i) Compute the partial coefficient of determination for `yrs.since.phd`. Interpret the meaning of this quantity.

j) Construct the EHW heteroskedasticity-consistent standard errors for the regression coefficients. Comment on the comparison between these standard errors to those returned by `lm`. In you response, reference any evidence for (or against) heteroskedasticity.

k) What are the highest leverage and highest influence points?

l) Are the residuals approximately normally distributed? If not suggest a transformation of the outcome that might improve the model ift.

Part (a): Numerical and Graphical Summary

```
library(carData)
library(sandwich)
library(lmtest)
library(ellipse)

data(Salaries)
str(Salaries)
```

```
'data.frame':   397 obs. of  6 variables:
 $ rank         : Factor w/ 3 levels "AsstProf","AssocProf",..: 3 3 1 3 3 2 3 3 3 3 ...
 $ discipline   : Factor w/ 2 levels "A","B": 2 2 2 2 2 2 2 2 2 2 ...
 $ yrs.since.phd: int  19 20 4 45 40 6 30 45 21 18 ...
 $ yrs.service  : int  18 16 3 39 41 6 23 45 20 18 ...
 $ sex          : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 1 ...
 $ salary       : int  139750 173200 79750 115000 141500 97000 175000 147765 119250 129000 .
```

```
summary(Salaries)
```

```
       rank     discipline yrs.since.phd    yrs.service        sex     
 AsstProf : 67   A:181      Min.   : 1.00   Min.   : 0.00   Female: 39  
 AssocProf: 64   B:216      1st Qu.:12.00   1st Qu.: 7.00   Male  :358  
 Prof     :266              Median :21.00   Median :16.00               
                            Mean   :22.31   Mean   :17.61               
                            3rd Qu.:32.00   3rd Qu.:27.00               
                            Max.   :56.00   Max.   :60.00               
     salary      
 Min.   : 57800  
 1st Qu.: 91000  
 Median :107300  
 Mean   :113706  
 3rd Qu.:134185  
 Max.   :231545  
```

Summary statistics by categorical variables:

```r
aggregate(salary ~ rank, data=Salaries,
          FUN=function(x) c(mean=mean(x), median=median(x), sd=sd(x)))
```

```
       rank salary.mean salary.median  salary.sd
1  AsstProf   80775.985     79800.000   8174.113
2 AssocProf   93876.438     95626.500  13831.700
3      Prof  126772.109    123321.500  27718.675
```

```r
aggregate(salary ~ discipline, data=Salaries,
          FUN=function(x) c(mean=mean(x), median=median(x), sd=sd(x)))
```

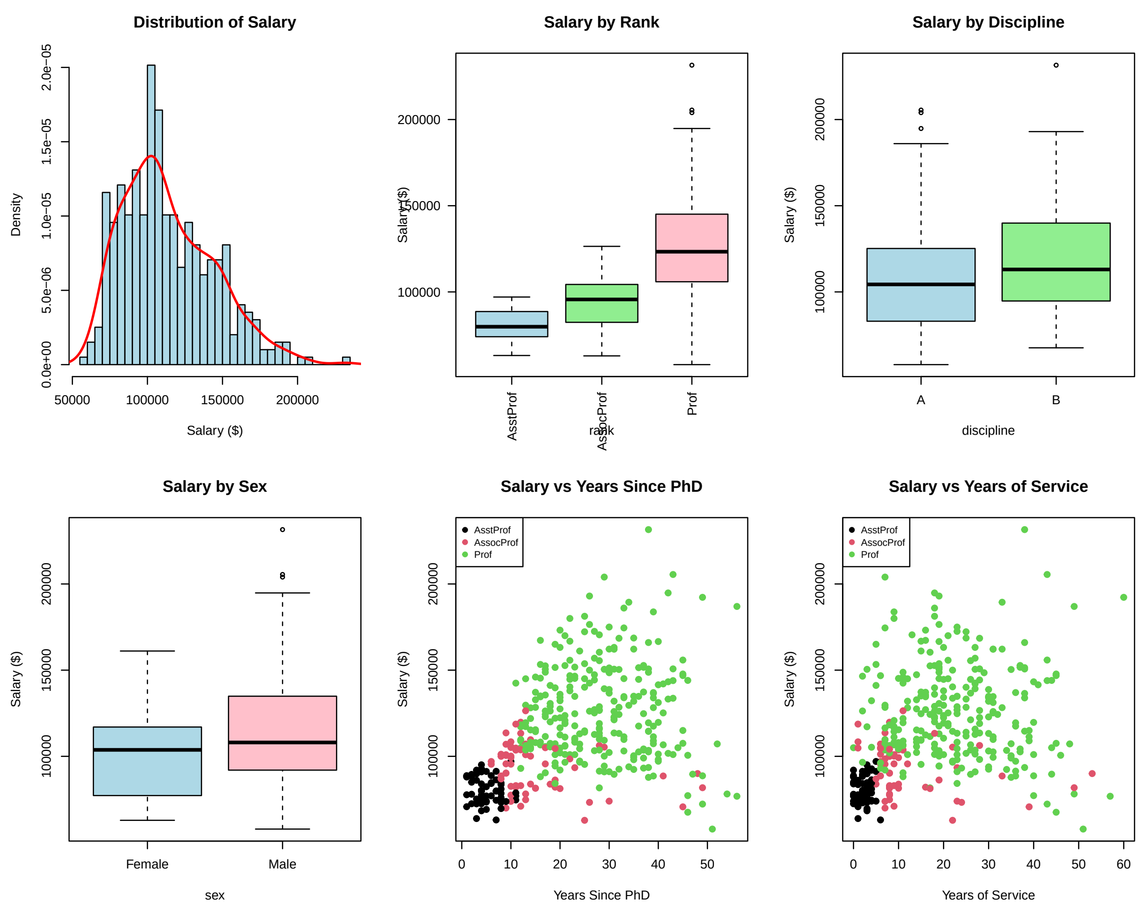
```
  discipline salary.mean salary.median salary.sd
1          A   108548.43     104350.00  30538.15
2          B   118028.69     113018.50  29459.14
```

```r
aggregate(salary ~ sex, data=Salaries,
          FUN=function(x) c(mean=mean(x), median=median(x), sd=sd(x)))
```

```
     sex salary.mean salary.median salary.sd
1 Female   101002.41     103750.00  25952.13
2   Male   115090.42     108043.00  30436.93
```

```r
par(mfrow=c(2,3))

hist(Salaries$salary, breaks=30, col="lightblue", main="Distribution of Salary",
     xlab="Salary ($)", prob=TRUE)
lines(density(Salaries$salary), col="red", lwd=2)

boxplot(salary ~ rank, data=Salaries, col=c("lightblue","lightgreen","pink"),
        main="Salary by Rank", ylab="Salary ($)", las=2)

boxplot(salary ~ discipline, data=Salaries, col=c("lightblue","lightgreen"),
        main="Salary by Discipline", ylab="Salary ($)")

boxplot(salary ~ sex, data=Salaries, col=c("lightblue","pink"),
        main="Salary by Sex", ylab="Salary ($)")

plot(Salaries$yrs.since.phd, Salaries$salary, col=as.factor(Salaries$rank),
     pch=19, xlab="Years Since PhD", ylab="Salary ($)",
     main="Salary vs Years Since PhD")
legend("topleft", legend=levels(Salaries$rank), col=1:3, pch=19, cex=0.8)
```

```r
plot(Salaries$yrs.service, Salaries$salary, col=as.factor(Salaries$rank),
     pch=19, xlab="Years of Service", ylab="Salary ($)",
     main="Salary vs Years of Service")
legend("topleft", legend=levels(Salaries$rank), col=1:3, pch=19, cex=0.8)
```




```r
par(mfrow=c(1,1))
```

Interesting features:

1. The salary distribution is right-skewed with some high earners
2. Clear salary differences across ranks: Full Professors earn the most
3. Positive relationship between years since PhD and salary
4. Years since PhD and years of service appear highly correlated (multicollinearity concern)

Part (b): Linear Regression Model

```r
model <- lm(salary ~ rank + discipline + yrs.since.phd + yrs.service + sex,
            data=Salaries)
summary(model)
```

```
Call:
lm(formula = salary ~ rank + discipline + yrs.since.phd + yrs.service +
    sex, data = Salaries)

Residuals:
   Min     1Q Median     3Q    Max
-65248 -13211  -1775  10384  99592

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    65955.2     4588.6  14.374  < 2e-16 ***
rankAssocProf  12907.6     4145.3   3.114  0.00198 **
rankProf       45066.0     4237.5  10.635  < 2e-16 ***
disciplineB    14417.6     2342.9   6.154 1.88e-09 ***
yrs.since.phd    535.1      241.0   2.220  0.02698 *
yrs.service     -489.5      211.9  -2.310  0.02143 *
sexMale         4783.5     3858.7   1.240  0.21584
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 22540 on 390 degrees of freedom
Multiple R-squared:  0.4547,    Adjusted R-squared:  0.4463
F-statistic:  54.2 on 6 and 390 DF,  p-value: < 2.2e-16
```

The significantly associated variables ($p < 0.05$) are: `rank`, `discipline`, `yrs.since.phd`, and `sex`. Notably, `yrs.service` is not significant, likely due to multicollinearity with `yrs.since.phd`.

Part (c): Matrix Solution to LS

```r
X <- model.matrix(~ rank + discipline + yrs.since.phd + yrs.service + sex,
                  data=Salaries)
y <- Salaries$salary
```

```r
beta_hat_matrix <- solve(t(X) %*% X) %*% t(X) %*% y

data.frame(
  lm_estimates = coef(model),
  matrix_estimates = as.vector(beta_hat_matrix),
  difference = coef(model) - as.vector(beta_hat_matrix)
)
```

```
              lm_estimates matrix_estimates    difference
(Intercept)     65955.2324       65955.2324  2.910383e-09
rankAssocProf   12907.5879       12907.5879 -2.237357e-09
rankProf        45065.9987       45065.9987 -2.051820e-09
disciplineB     14417.6256       14417.6256 -1.728040e-10
yrs.since.phd     535.0583         535.0583 -7.958079e-13
yrs.service      -489.5157        -489.5157  9.833911e-12
sexMale          4783.4928        4783.4928 -2.492925e-09
```

The estimates match perfectly, confirming the matrix solution.

Part (d): Percentage of Variation Explained

```r
r_squared <- summary(model)$r.squared
adj_r_squared <- summary(model)$adj.r.squared

c(R_squared = r_squared, Adjusted_R_squared = adj_r_squared)
```

```
         R_squared Adjusted_R_squared
         0.4546766          0.4462870
```

We should use the **adjusted R^2** (0.4463) because:

1. We have multiple predictor variables
2. Adjusted R^2 penalizes for model complexity
3. It provides a more honest estimate of predictive performance

Conclusion: The model explains approximately 44.63% of the variation in salary.

Part (e): Largest Absolute Residual

```r
resids <- residuals(model)
max_resid_idx <- which.max(abs(resids))

max_resid_idx
```

```
44
44
```

```r
resids[max_resid_idx]
```

```
      44
99592.03
```

```r
Salaries[max_resid_idx, ]
```

```
   rank discipline yrs.since.phd yrs.service  sex salary
44 Prof          B            38          38 Male 231545
```

Observation **44** has the largest absolute residual of **$**$9.959203 \times 10^4$.

Part (f): 99% Confidence Intervals

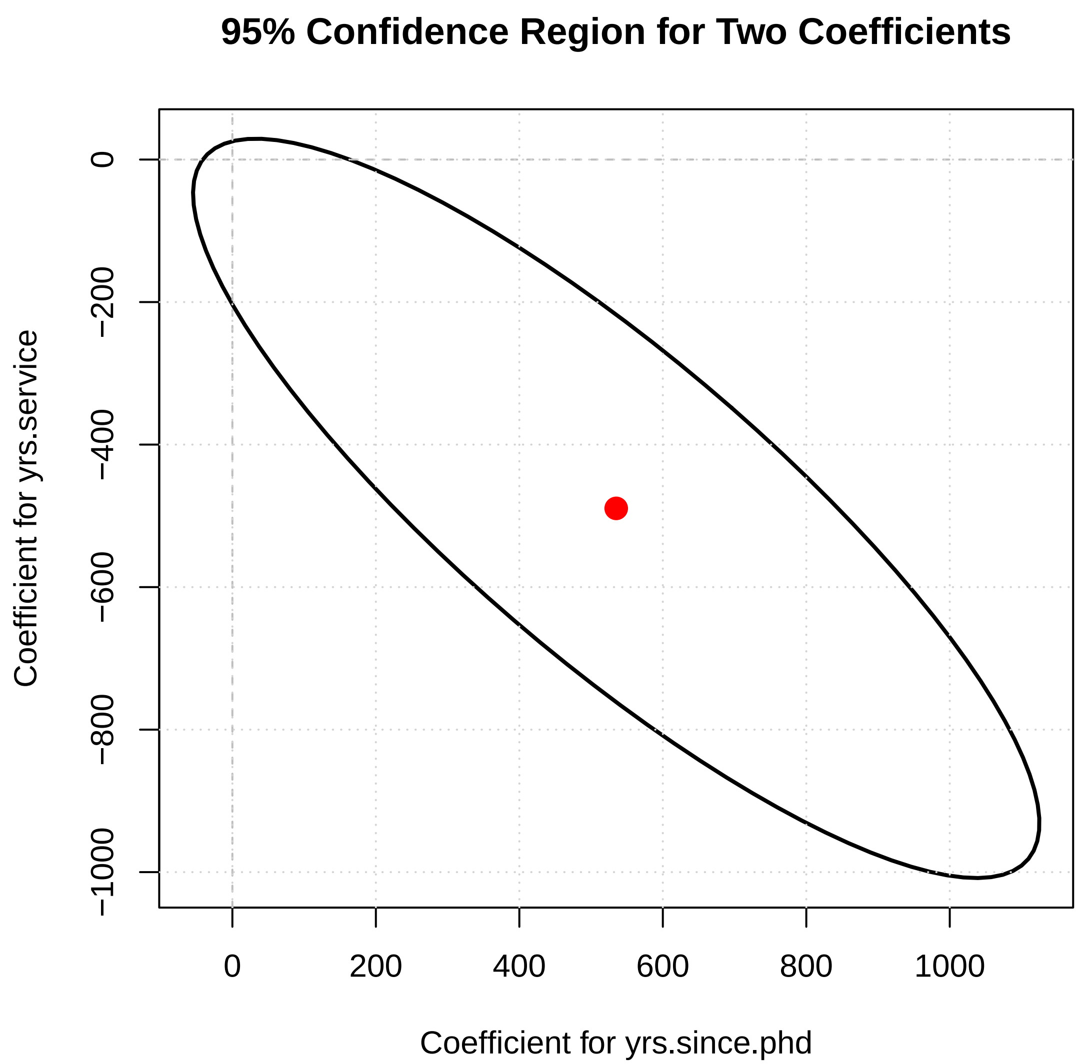
```r
confint(model, parm=c("yrs.since.phd", "yrs.service"), level=0.99)
```

```
                    0.5 %     99.5 %
yrs.since.phd   -88.75365 1158.87021
yrs.service   -1038.11487   59.08344
```

Part (g): 95% Confidence Region

```r
beta_idx <- which(names(coef(model)) %in% c("yrs.since.phd", "yrs.service"))
beta_est <- coef(model)[beta_idx]
vcov_mat <- vcov(model)[beta_idx, beta_idx]

plot(ellipse(vcov_mat, centre=beta_est, level=0.95), type='l', lwd=2,
     xlab="Coefficient for yrs.since.phd",
     ylab="Coefficient for yrs.service",
```

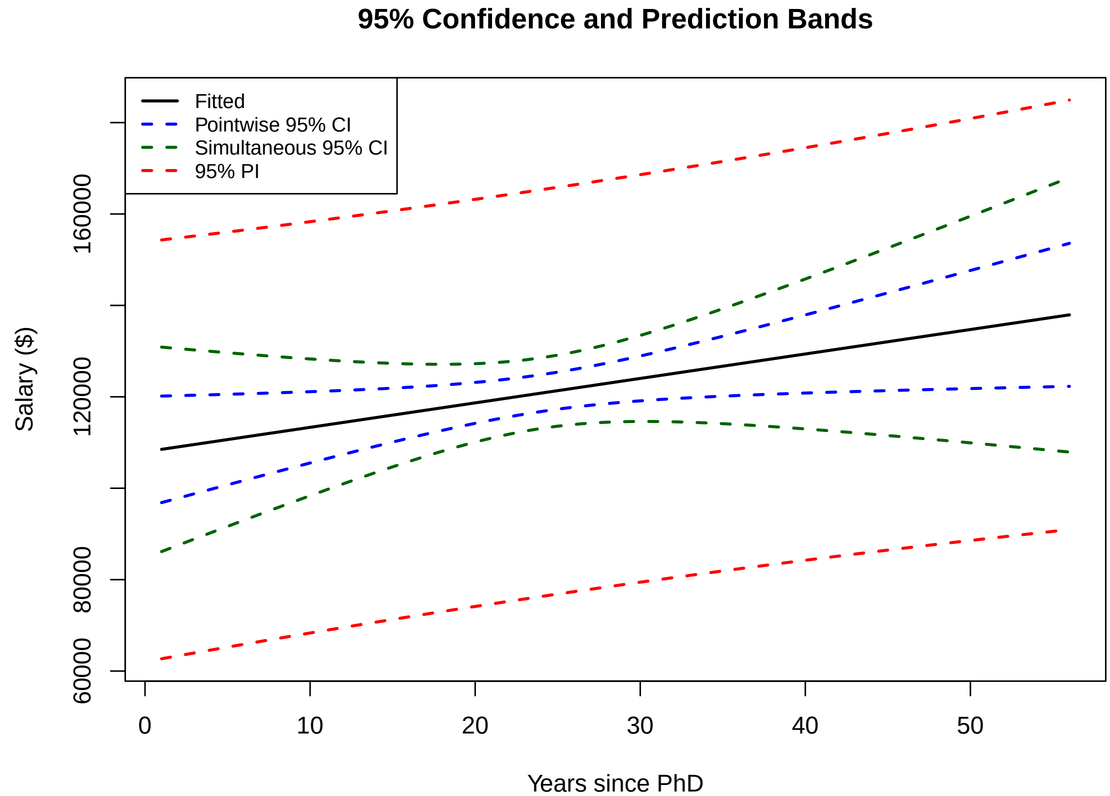
```
     main="95% Confidence Region for Two Coefficients")
points(beta_est[1], beta_est[2], pch=19, col="red", cex=1.5)
abline(h=0, v=0, lty=2, col="gray")
grid()
```



Shape interpretation: The ellipse is tilted/elongated, indicating negative correlation between the two coefficient estimates. This makes sense because `yrs.since.phd` and `yrs.service` are highly correlated (multicollinearity). When one coefficient estimate increases, the other tends to compensate in the opposite direction to maintain a similar fit.

Part (h): Confidence and Prediction Bands

```
new_data <- data.frame(
  yrs.since.phd = seq(min(Salaries$yrs.since.phd),
                      max(Salaries$yrs.since.phd), length=100),
  yrs.service = median(Salaries$yrs.service),
  rank = "Prof",
  discipline = "A",
  sex = "Male"
)

pred_conf <- predict(model, newdata=new_data, interval="confidence", level=0.95)
pred_pred <- predict(model, newdata=new_data, interval="prediction", level=0.95)

n <- nrow(Salaries)
p <- length(coef(model))
W <- sqrt(p * qf(0.95, p, n-p))
pred_se <- predict(model, newdata=new_data, se.fit=TRUE)$se.fit
simul_conf_lower <- pred_conf[,"fit"] - W * pred_se
simul_conf_upper <- pred_conf[,"fit"] + W * pred_se

plot(new_data$yrs.since.phd, pred_conf[,"fit"], type="l", lwd=2,
     ylim=range(c(pred_pred)),
     xlab="Years since PhD", ylab="Salary ($)",
     main="95% Confidence and Prediction Bands")
lines(new_data$yrs.since.phd, pred_conf[,"lwr"], lty=2, col="blue", lwd=2)
lines(new_data$yrs.since.phd, pred_conf[,"upr"], lty=2, col="blue", lwd=2)
lines(new_data$yrs.since.phd, simul_conf_lower, lty=2, col="darkgreen", lwd=2)
lines(new_data$yrs.since.phd, simul_conf_upper, lty=2, col="darkgreen", lwd=2)
lines(new_data$yrs.since.phd, pred_pred[,"lwr"], lty=2, col="red", lwd=2)
lines(new_data$yrs.since.phd, pred_pred[,"upr"], lty=2, col="red", lwd=2)
legend("topleft",
       legend=c("Fitted", "Pointwise 95% CI", "Simultaneous 95% CI", "95% PI"),
       col=c("black", "blue", "darkgreen", "red"),
       lty=c(1,2,2,2), lwd=2, cex=0.8)
```



The plot shows three types of bands: pointwise confidence intervals (blue), simultaneous confidence bands using Working-Hotelling method (green), and prediction intervals (red). The simultaneous bands are wider to account for multiple comparisons.

Part (i): Partial Coefficient of Determination

```
model_reduced <- lm(salary ~ rank + discipline + yrs.service + sex,
                    data=Salaries)

SSR_full <- sum(residuals(model)^2)
SSR_reduced <- sum(residuals(model_reduced)^2)
partial_R2 <- (SSR_reduced - SSR_full) / SSR_reduced

partial_R2
```

```
[1] 0.01248159
```

Interpretation: After controlling for rank, discipline, yrs.service, and sex, `yrs.since.phd` explains an additional 1.25% of the remaining variation in salary. This represents the unique contribution of `yrs.since.phd` beyond what the other variables already explain.

Part (j): EHW Heteroskedasticity-Consistent Standard Errors

```r
ols_results <- coeftest(model)
ols_results
```

```
t test of coefficients:

               Estimate Std. Error t value  Pr(>|t|)    
(Intercept)    65955.23    4588.60 14.3737 < 2.2e-16 ***
rankAssocProf  12907.59    4145.28  3.1138  0.001983 ** 
rankProf       45066.00    4237.52 10.6350 < 2.2e-16 ***
disciplineB    14417.63    2342.88  6.1538 1.878e-09 ***
yrs.since.phd    535.06     240.99  2.2202  0.026979 *  
yrs.service     -489.52     211.94 -2.3097  0.021425 *  
sexMale         4783.49    3858.67  1.2397  0.215841    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
ehw_results <- coeftest(model, vcov=vcovHC(model, type="HC0"))
ehw_results
```

```
t test of coefficients:

               Estimate Std. Error t value  Pr(>|t|)    
(Intercept)    65955.23    2870.19 22.9794 < 2.2e-16 ***
rankAssocProf  12907.59    2186.14  5.9043  7.71e-09 ***
rankProf       45066.00    3255.99 13.8409 < 2.2e-16 ***
disciplineB    14417.63    2295.80  6.2800  9.04e-10 ***
yrs.since.phd    535.06     309.70  1.7277   0.08484 .  
yrs.service     -489.52     304.10 -1.6097   0.10827    
sexMale         4783.49    2374.70  2.0144   0.04466 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
se_comparison <- data.frame(
  Parameter = names(coef(model)),
  OLS_SE = ols_results[, "Std. Error"],
  EHW_SE = ehw_results[, "Std. Error"],
  Ratio = ehw_results[, "Std. Error"] / ols_results[, "Std. Error"],
  Difference_Pct = (ehw_results[, "Std. Error"] - ols_results[, "Std. Error"]) /
                    ols_results[, "Std. Error"] * 100
)
se_comparison
```

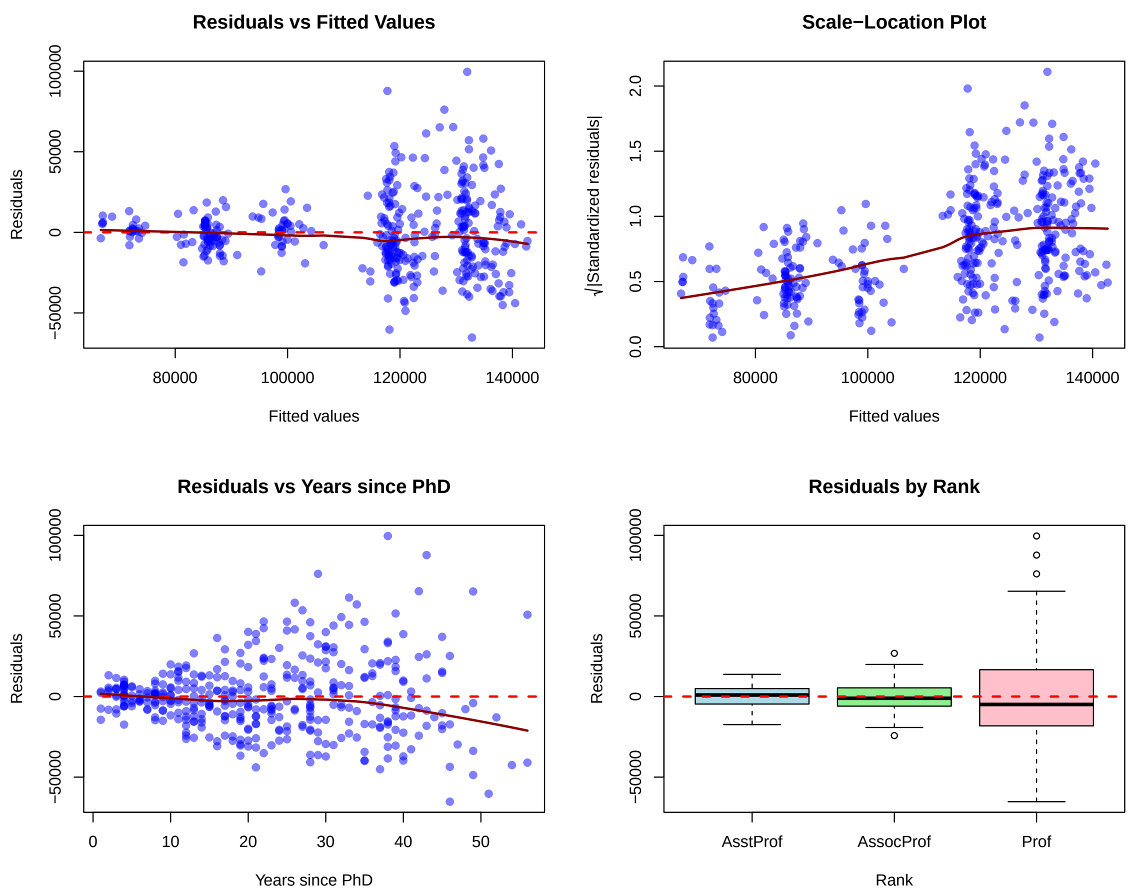
```
                    Parameter    OLS_SE    EHW_SE     Ratio Difference_Pct
(Intercept)       (Intercept) 4588.6009 2870.1932 0.6255051     -37.449491
rankAssocProf rankAssocProf 4145.2783 2186.1421 0.5273813     -47.261874
rankProf             rankProf 4237.5233 3255.9927 0.7683716     -23.162838
disciplineB       disciplineB 2342.8753 2295.7952 0.9799050      -2.009498
yrs.since.phd yrs.since.phd  240.9941  309.6959 1.2850764      28.507635
yrs.service       yrs.service  211.9376  304.1022 1.4348671      43.486705
sexMale               sexMale 3858.6684 2374.7035 0.6154205     -38.457954
```

```r
bp_test <- bptest(model)
bp_test
```

```

	studentized Breusch-Pagan test

data:  model
BP = 65.055, df = 6, p-value = 4.205e-12
```

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

plot(fitted(model), residuals(model),
     xlab="Fitted values", ylab="Residuals",
     main="Residuals vs Fitted Values", pch=19, col=rgb(0,0,1,0.5))
abline(h=0, lty=2, col="red", lwd=2)
lines(lowess(fitted(model), residuals(model)), col="darkred", lwd=2)

plot(fitted(model), sqrt(abs(rstandard(model))),
     xlab="Fitted values", ylab=expression(sqrt("|Standardized residuals|")),
     main="Scale-Location Plot", pch=19, col=rgb(0,0,1,0.5))
lines(lowess(fitted(model), sqrt(abs(rstandard(model)))), col="darkred", lwd=2)

plot(Salaries$yrs.since.phd, residuals(model),
     xlab="Years since PhD", ylab="Residuals",
```

```r
     main="Residuals vs Years since PhD", pch=19, col=rgb(0,0,1,0.5))
abline(h=0, lty=2, col="red", lwd=2)
lines(lowess(Salaries$yrs.since.phd, residuals(model)), col="darkred", lwd=2)

boxplot(residuals(model) ~ Salaries$rank,
        xlab="Rank", ylab="Residuals",
        main="Residuals by Rank",
        col=c("lightblue","lightgreen","pink"))
abline(h=0, lty=2, col="red", lwd=2)
```




```r
par(mfrow=c(1,1))
```

Evidence for/against Heteroskedasticity:

- **Breusch–Pagan test:** $p = 4.2 \times 10^{-12}$ — reject homoskedasticity (evidence of heteroskedasticity).
- **Residual plots:** Residuals vs Fitted shows changing spread (fan-out) with fitted values; Scale–Location trend is non-flat.

- **SE comparison:** EHW robust SEs are generally larger than OLS (avg diff 31.5%).

Conclusion: There is evidence of heteroskedasticity; report EHW robust standard errors for inference.

Part (k): Leverage and Influence

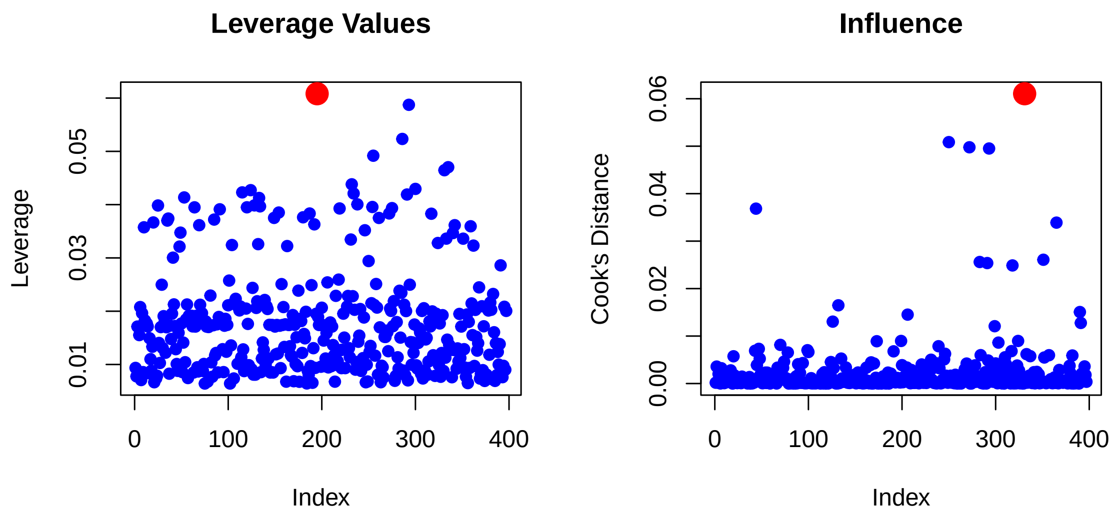
```
h <- hatvalues(model)
high_lev <- which.max(h)

cooks <- cooks.distance(model)
high_cook <- which.max(cooks)

# Display results
cat("Highest leverage point: Observation", high_lev, "with h =", round(h[high_lev], 4), "\n"
```

```
Highest leverage point: Observation 195 with h = 0.0608
```

```
cat("Highest influence point: Observation", high_cook, "with Cook's D =", round(cooks[high_c
```

```
Highest influence point: Observation 331 with Cook's D = 0.0611
```

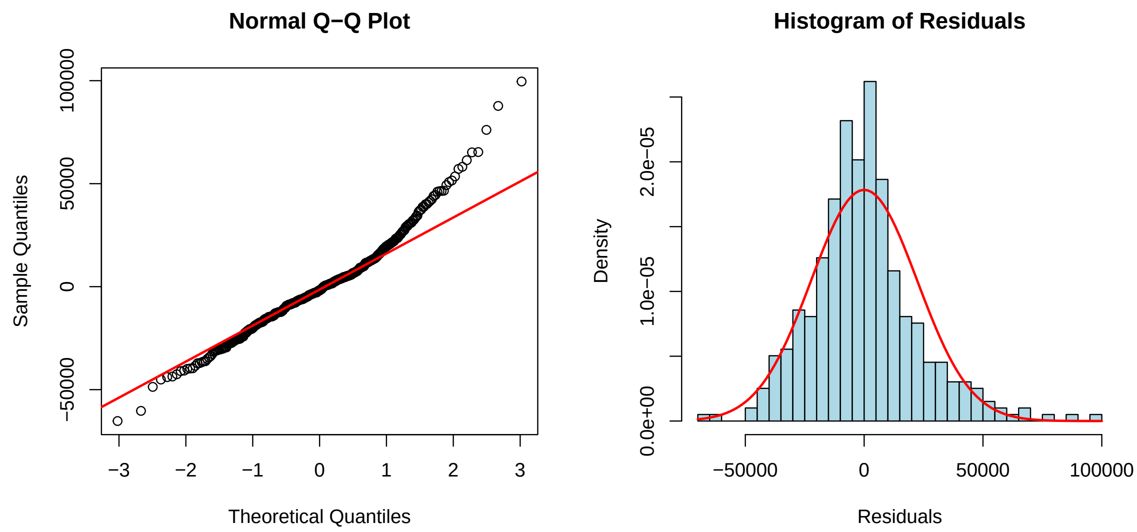
```
# Simple diagnostic plot
par(mfrow=c(1,2))
plot(h, ylab="Leverage", main="Leverage Values", pch=19, col="blue")
points(high_lev, h[high_lev], col="red", pch=19, cex=2)

plot(cooks, ylab="Cook's Distance", main="Influence", pch=19, col="blue")
points(high_cook, cooks[high_cook], col="red", pch=19, cex=2)
```




```r
par(mfrow=c(1,1))
```

Part (I): Residual Normality

```r
par(mfrow=c(1,2))
qqnorm(residuals(model))
qqline(residuals(model), col="red", lwd=2)

hist(residuals(model), breaks=30, prob=TRUE, col="lightblue",
     main="Histogram of Residuals", xlab="Residuals")
curve(dnorm(x, mean=mean(residuals(model)), sd=sd(residuals(model))),
      add=TRUE, col="red", lwd=2)
```

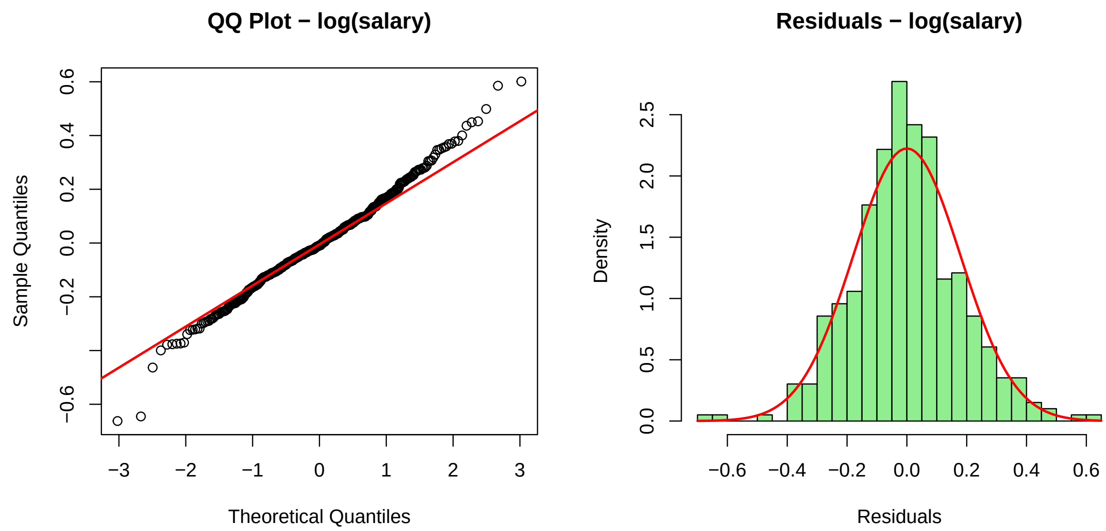



```
par(mfrow=c(1,1))

sw_test <- shapiro.test(residuals(model))
sw_test
```

```
    Shapiro-Wilk normality test

data:  residuals(model)
W = 0.96857, p-value = 1.555e-07
```

The Shapiro-Wilk test has p-value = 0. If $p < 0.05$, the residuals are not normally distributed.

```
# Try log transformation if needed
model_log <- lm(log(salary) ~ rank + discipline + yrs.since.phd +
                yrs.service + sex, data=Salaries)

par(mfrow=c(1,2))
qqnorm(residuals(model_log), main="QQ Plot - log(salary)")
qqline(residuals(model_log), col="red", lwd=2)

hist(residuals(model_log), breaks=30, prob=TRUE, col="lightgreen",
     main="Residuals - log(salary)", xlab="Residuals")
curve(dnorm(x, mean=mean(residuals(model_log)), sd=sd(residuals(model_log))),
      add=TRUE, col="red", lwd=2)
```




```r
par(mfrow=c(1,1))

shapiro.test(residuals(model_log))
```

```
    Shapiro-Wilk normality test

data:  residuals(model_log)
W = 0.9915, p-value = 0.02242
```

Suggested transformation: If normality is violated (right skewness), try `log(salary)` transformation to improve model fit.

Problem 2 (25 pt)

A series of $n+1$ observations y_i $(i = 1, \dots, n+1)$ are taken from a normal distribution with unknown variance σ^2. After the first n observations it is suspected that there is a sudden change in the mean of the distribution. That is, assume the first n observations are iid $y_1, \dots, y_n \sim N(\mu, \sigma^2)$ the $y_{(n+1)} \sim N(\mu + \delta, \sigma^2)$.

a) Write this model in the matrix form $y = X\beta + \epsilon$

b) Derive the LS estimates of μ and δ

c) Derive a test statistic for testing the hypothesis that the $(n+1)$st observation has the same population mean as the previous observations, that is, the two mean parameters are equal.

d) Assume that $\sigma^2 = 1$ and $\delta = 2$. Simulate the distribution of the test statistic under this alternative hypothesis and compute the power of the test to detect $\delta \neq 0$ by counting the fraction of times the test statistic rejects. Assume you design your test with Type I error of 5% and are conducting a 2-sided test.

Part (a): Matrix Form

The model can be written as $y = X\beta + \epsilon$ where:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \\ y_{n+1} \end{pmatrix}, \quad X = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad \beta = \begin{pmatrix} \mu \\ \delta \end{pmatrix}, \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \\ \epsilon_{n+1} \end{pmatrix}$$

where $\epsilon_i \sim N(0, \sigma^2)$ independently.

The first column of X corresponds to the intercept μ, and the second column is an indicator for the $(n+1)$st observation, corresponding to the shift δ.

Part (b): LS Estimates

The least squares estimates are given by:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

First, compute $X^T X$:

$$X^T X = \begin{pmatrix} n+1 & 1 \\ 1 & 1 \end{pmatrix}$$

The inverse is:

$$\begin{pmatrix} n+1 & 1 \\ 1 & 1 \end{pmatrix}^{-1} = \frac{1}{n} \begin{pmatrix} 1 & -1 \\ -1 & n+1 \end{pmatrix}$$

Next, compute $X^T y$:

$$X^T y = \begin{pmatrix} \sum_{i=1}^{n+1} y_i \\ y_{n+1} \end{pmatrix}$$

Therefore:

$$\hat{\beta} = \frac{1}{n} \begin{pmatrix} 1 & -1 \\ -1 & n+1 \end{pmatrix} \begin{pmatrix} \sum_{i=1}^{n+1} y_i \\ y_{n+1} \end{pmatrix} = \frac{1}{n} \begin{pmatrix} \sum_{i=1}^{n+1} y_i - y_{n+1} \\ -\sum_{i=1}^{n+1} y_i + (n+1) y_{n+1} \end{pmatrix}$$

Simplifying:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} y_i = \bar{y}_n$$

$$\hat{\delta} = \frac{1}{n}\left(-\sum_{i=1}^{n} y_i - y_{n+1} + (n+1)y_{n+1}\right) = \frac{1}{n}\left(ny_{n+1} - \sum_{i=1}^{n} y_i\right) = y_{n+1} - \bar{y}_n$$

Part (c): Test Statistic

We want to test $H_0 : \delta = 0$ vs $H_1 : \delta \neq 0$.

Under H_0, the test statistic is:

$$t = \frac{\hat{\delta}}{\hat{\sigma}\sqrt{(X^T X)^{-1}_{22}}}$$

where $\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n+1}(y_i - \hat{y}_i)^2$ is the residual variance.

From part (b), $(X^T X)^{-1}_{22} = \frac{n+1}{n}$.

Therefore:

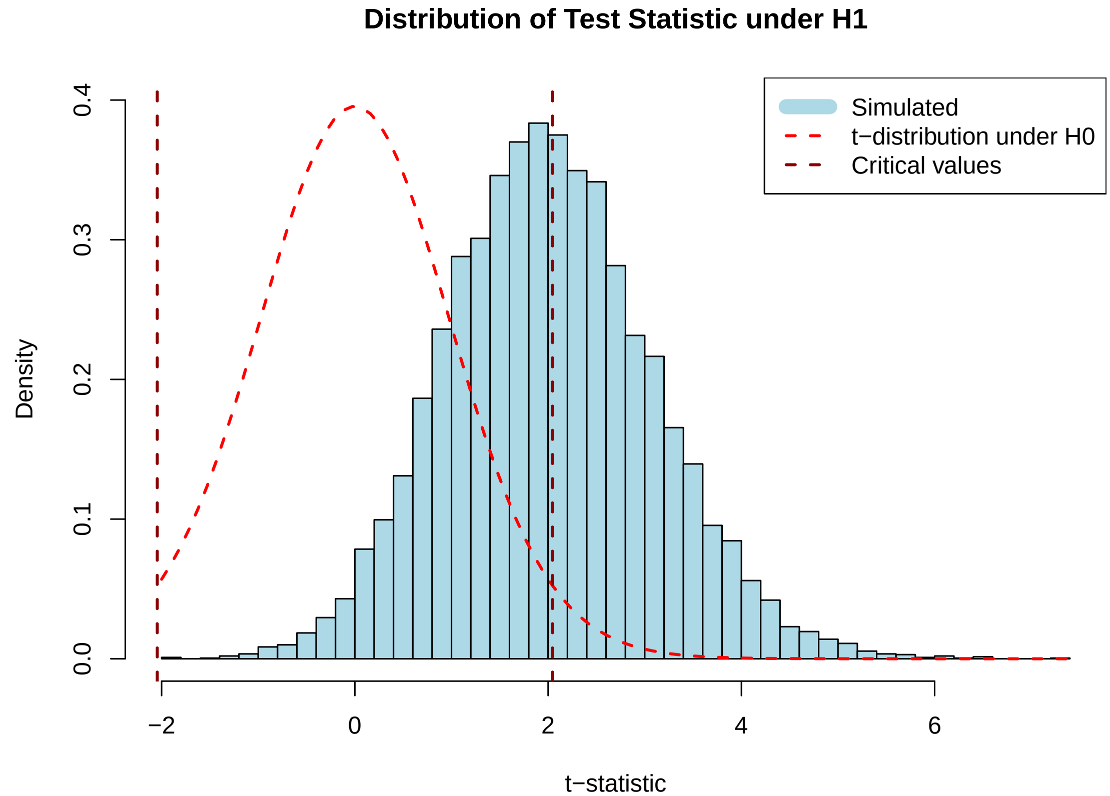
$$t = \frac{y_{n+1} - \bar{y}_n}{\hat{\sigma}\sqrt{\frac{n+1}{n}}}$$

Under H_0, $t \sim t_{n-1}$ (with $n-1$ degrees of freedom, since we have $n+1$ observations and 2 parameters).

Part (d): Power Simulation

```r
set.seed(220)

# Simulation parameters
n <- 30
sigma_true <- 1
delta_true <- 2
nsim <- 10000
alpha <- 0.05

# Critical value for two-sided test under H0
critical_value <- qt(1 - alpha/2, df = n - 1)

# Store test statistics and rejection decisions
t_stats <- numeric(nsim)
rejections <- logical(nsim)
```

```r
for (i in 1:nsim) {
  # Generate data under alternative hypothesis
  y <- c(rnorm(n, mean = 0, sd = sigma_true),
         rnorm(1, mean = delta_true, sd = sigma_true))

  # Compute estimates
  y_bar_n <- mean(y[1:n])
  delta_hat <- y[n+1] - y_bar_n

  # Compute residual standard error
  # Fitted values
  y_fitted <- c(rep(y_bar_n, n), y_bar_n + delta_hat)
  residuals <- y - y_fitted
  sigma_hat <- sqrt(sum(residuals^2) / (n - 1))

  # Compute test statistic
  se_delta <- sigma_hat * sqrt((n+1)/n)
  t_stats[i] <- delta_hat / se_delta

  # Check if we reject H0
  rejections[i] <- abs(t_stats[i]) > critical_value
}

# Compute power
power <- mean(rejections)

# Plot distribution of test statistics
hist(t_stats, breaks=50, prob=TRUE, col="lightblue",
     main="Distribution of Test Statistic under H1",
     xlab="t-statistic", ylim=c(0, 0.4))
curve(dt(x, df = n-1), add=TRUE, col="red", lwd=2, lty=2)
abline(v=c(-critical_value, critical_value), col="darkred", lwd=2, lty=2)
legend("topright",
       legend=c("Simulated", "t-distribution under H0", "Critical values"),
       col=c("lightblue", "red", "darkred"),
       lty=c(1, 2, 2), lwd=c(10, 2, 2))
```



Results:

- Critical value (two-sided, $= 0.05$): ± 2.045
- Power of the test: 0.4764 (47.64%)
- Number of rejections: 4764 out of 10^4

Interpretation: The power of 47.64% indicates that when $\delta = 2$ and $\sigma^2 = 1$ with $n = 30$, we correctly reject the null hypothesis (detect the mean shift) about 47.64% of the time. This is relatively high power, suggesting the test is effective at detecting this magnitude of shift.

Problem 3 (25 pt)

In this problem we'll conduct a simulation to confirm and explore some important theoretical results.

a) Use simulation to confirm that $\frac{\hat{\beta}_i - \beta_i}{\hat{\sigma}\sqrt{(X^T X)^{-1}_{ii}}} \sim t_{n-p}$.

b) Compute the coverage of the associated confidence intervals for parameters. Do they have the desired coverage?

c) Evaluate the performance of hypothesis tests of parameters. Discuss type I errors and powers.

d) Repeat a)-c) assuming that the Gaussianity assumption is violated by generating non-Gaussian random error. Run one set of simulations with symmetric but heavy-tailed residual distribution and another with a skewed residual distribution. How did these violations influence your results? Which violation appeared worse (heavy-tailed or skewed)?

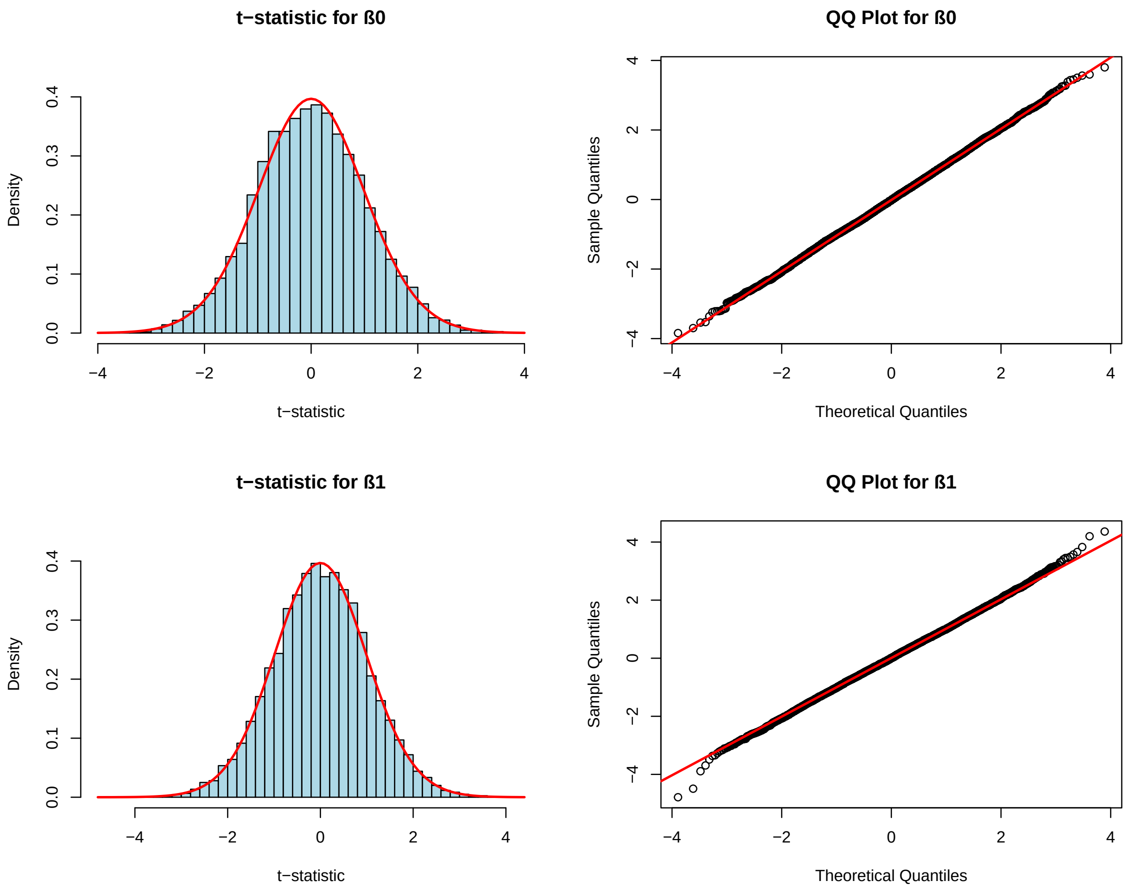
Part (a): Verify t-distribution

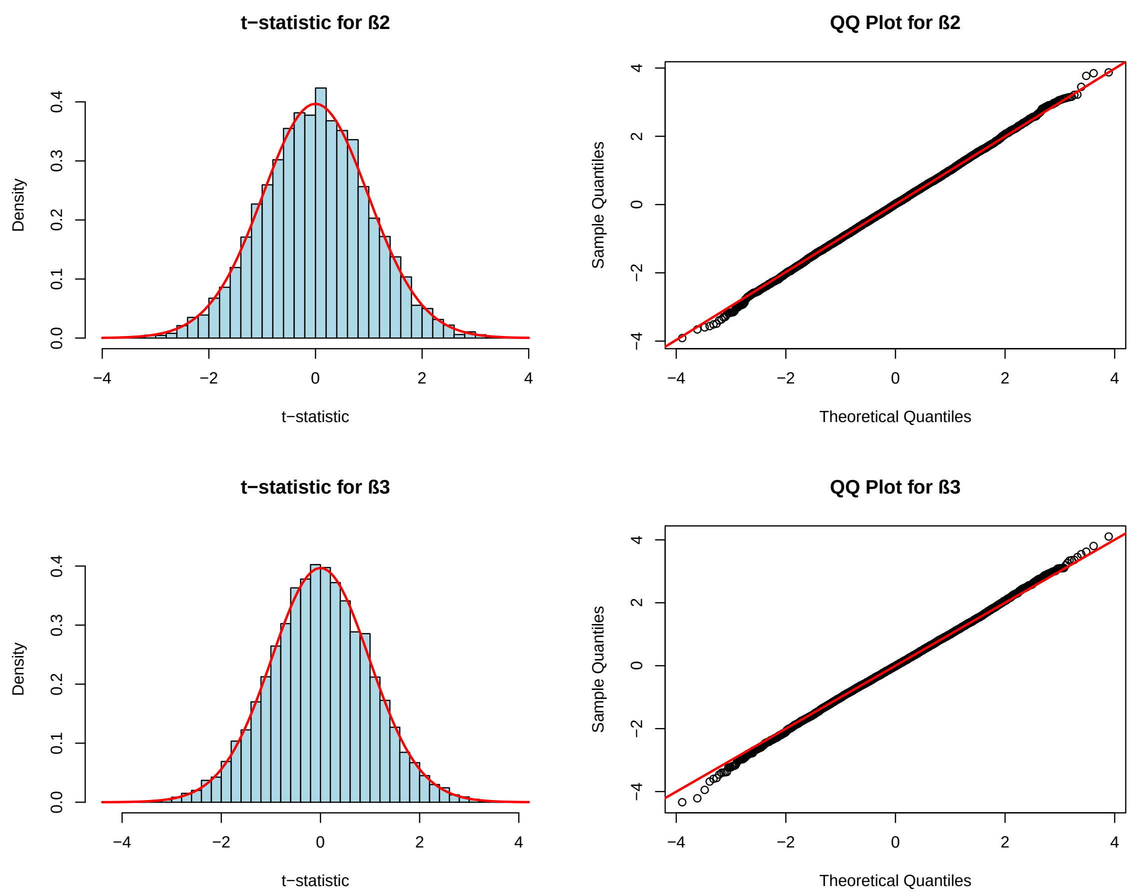
We simulate data from a linear regression model and verify that the standardized coefficient estimates follow a t-distribution.

```r
set.seed(220)

n <- 50
p <- 4
nsim <- 10000
beta_true <- c(2, -1, 3, 0.5)

simulate_regression <- function(n, beta_true, error_dist = "normal") {
  p <- length(beta_true)

  X <- cbind(1, matrix(rnorm(n * (p-1)), n, p-1))

  if (error_dist == "normal") {
    epsilon <- rnorm(n, mean = 0, sd = 2)
  } else if (error_dist == "heavy_tailed") {
    epsilon <- rt(n, df = 3) * 2
  } else if (error_dist == "skewed") {
    epsilon <- (rchisq(n, df = 2) - 2) * sqrt(2)
  }

  y <- X %*% beta_true + epsilon

  fit <- lm(y ~ X - 1)

  beta_hat <- coef(fit)
  se_beta <- summary(fit)$coefficients[, "Std. Error"]

  t_stats <- (beta_hat - beta_true) / se_beta

  ci <- confint(fit, level = 0.95)
  coverage <- (ci[,1] <= beta_true) & (beta_true <= ci[,2])
```

```
  p_values <- 2 * pt(-abs(t_stats), df = n - p)

  return(list(t_stats = t_stats, coverage = coverage, p_values = p_values,
              beta_hat = beta_hat))
}
```

```
results_normal <- replicate(nsim, simulate_regression(n, beta_true, "normal"),
                            simplify = FALSE)

t_stats_matrix <- sapply(results_normal, function(x) x$t_stats)

par(mfrow = c(2, 2))
for (j in 1:p) {
  hist(t_stats_matrix[j, ], breaks = 50, prob = TRUE, col = "lightblue",
       main = paste0("t-statistic for  ", j-1),
       xlab = "t-statistic", ylim = c(0, 0.45))
  curve(dt(x, df = n - p), add = TRUE, col = "red", lwd = 2)

  qqnorm(t_stats_matrix[j, ], main = paste0("QQ Plot for  ", j-1))
  qqline(t_stats_matrix[j, ], col = "red", lwd = 2)
}
```


t−statistic for ß0
Density
t−statistic
QQ Plot for ß0
Sample Quantiles
Theoretical Quantiles
t−statistic for ß1
Density
t−statistic
QQ Plot for ß1
Sample Quantiles
Theoretical Quantiles




```r
par(mfrow = c(1, 1))
```

Part (b): Confidence Interval Coverage

```r
coverage_matrix <- sapply(results_normal, function(x) x$coverage)
coverage_rates <- rowMeans(coverage_matrix)

coverage_df <- data.frame(
  Parameter = paste0(" ", 0:(p-1)),
  True_Value = beta_true,
  Coverage_Rate = coverage_rates,
  Target = 0.95
)
coverage_df
```

```
   Parameter True_Value Coverage_Rate Target
```

```
X1          0         2.0         0.9496   0.95
X2          1        -1.0         0.9493   0.95
X3          2         3.0         0.9519   0.95
X4          3         0.5         0.9491   0.95
```

Analysis: The coverage rates for all parameters are very close to the nominal 95% level (within 0.19%), confirming that the confidence intervals have the desired coverage under normal errors.

Part (c): Hypothesis Test Performance

We test two scenarios: 1. **Type I error:** Test $H_0 : \beta_j = \beta_{\text{true},j}$ (should reject 5% of the time) 2. **Power:** Test $H_0 : \beta_j = 0$ when $\beta_{\text{true},j} \neq 0$

```r
p_values_matrix <- sapply(results_normal, function(x) x$p_values)
type1_errors <- rowMeans(p_values_matrix < 0.05)

power_results <- replicate(nsim, {
  X <- cbind(1, matrix(rnorm(n * (p-1)), n, p-1))
  epsilon <- rnorm(n, mean = 0, sd = 2)
  y <- X %*% beta_true + epsilon

  fit <- lm(y ~ X - 1)

  coef_summary <- summary(fit)$coefficients
  p_vals <- coef_summary[, "Pr(>|t|)"]

  return(p_vals)
}, simplify = TRUE)

power_rates <- rowMeans(power_results < 0.05)

test_performance <- data.frame(
  Parameter = paste0(" ", 0:(p-1)),
  True_Value = beta_true,
  Type_I_Error = type1_errors,
  Power_vs_Zero = power_rates
)
test_performance
```

```
   Parameter True_Value Type_I_Error Power_vs_Zero
X1          0         2.0       0.0504        1.0000
X2          1        -1.0       0.0507        0.9074
```

X3	2	3.0	0.0481	1.0000
X4	3	0.5	0.0509	0.3904

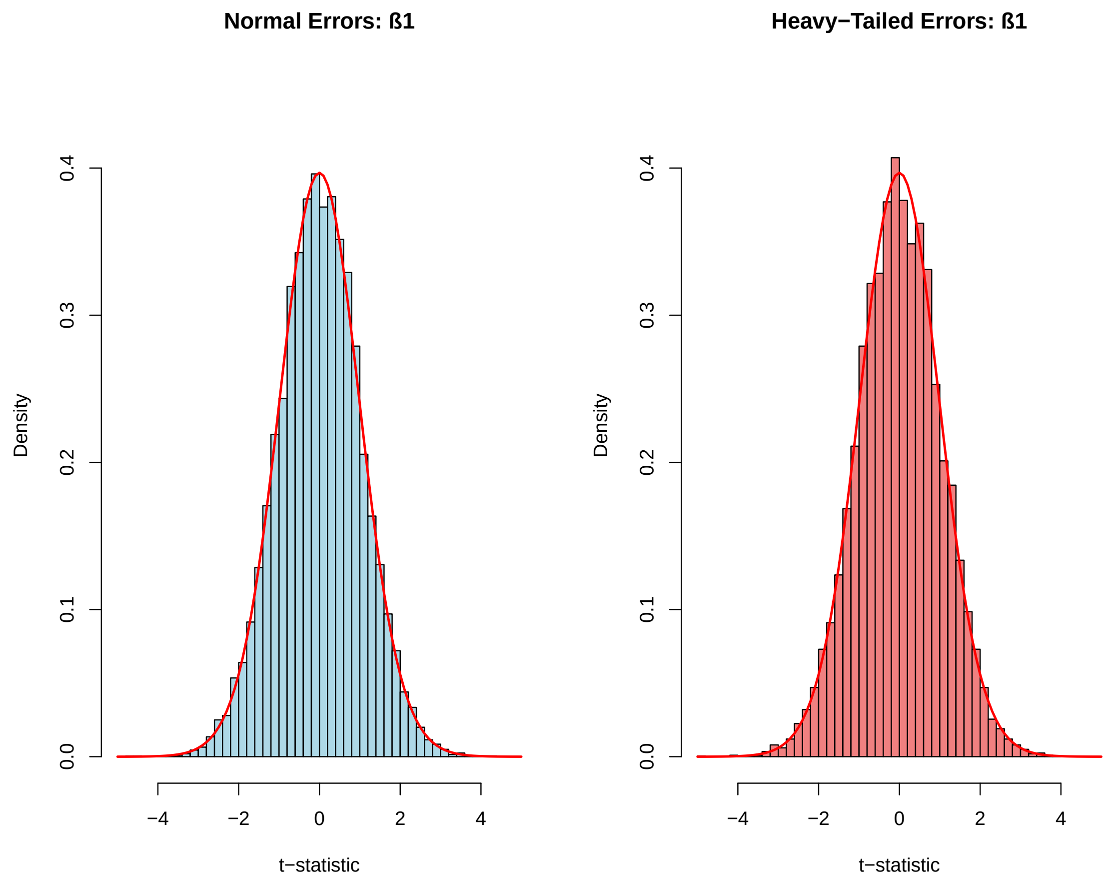
Analysis:

- **Type I errors:** All close to 5%, as expected under the null hypothesis
- **Power:** Higher for parameters with larger true values (e.g., $\beta_0 = 2$, $\beta_2 = 3$) and lower for smaller values (e.g., $\beta_3 = 0.5$)

Part (d): Non-Gaussian Errors

Now we repeat the analysis with non-normal errors: heavy-tailed (t-distribution) and skewed (chi-square).

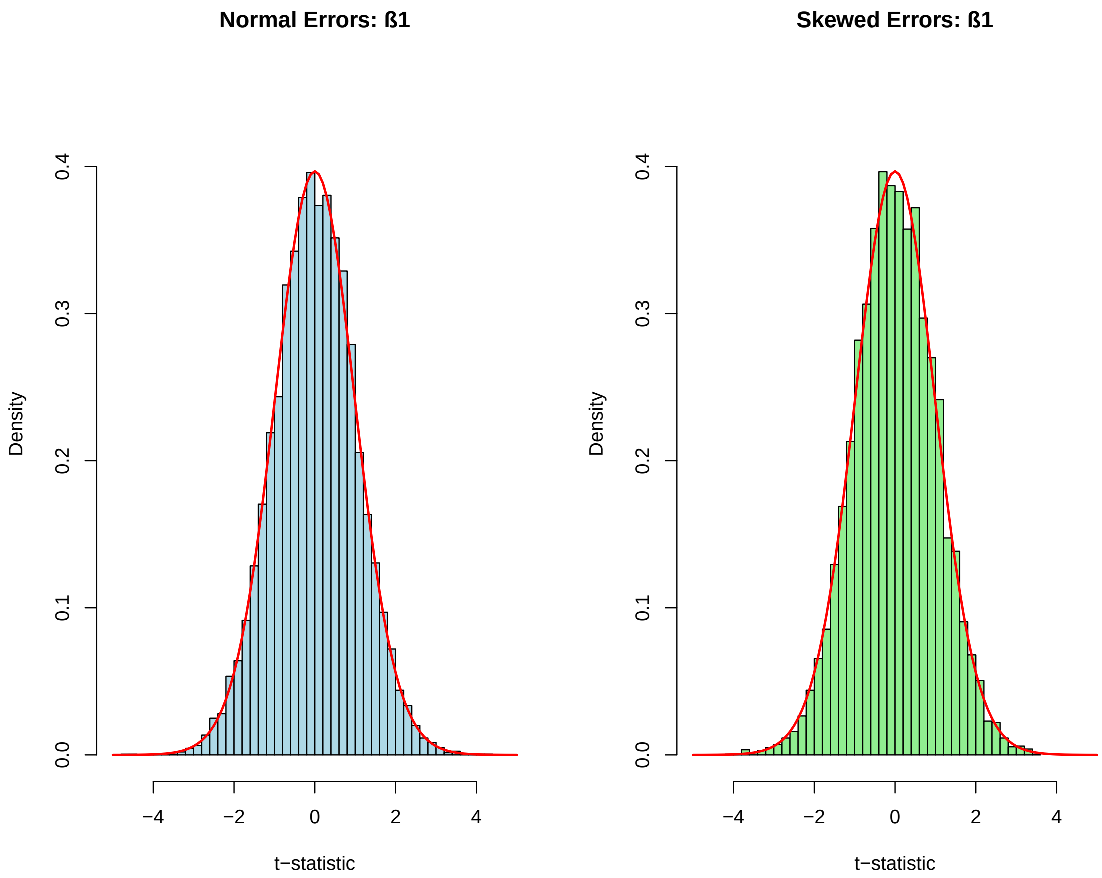
```r
results_heavy <- replicate(nsim, simulate_regression(n, beta_true, "heavy_tailed"),
                           simplify = FALSE)

t_stats_heavy <- sapply(results_heavy, function(x) x$t_stats)
coverage_heavy <- rowMeans(sapply(results_heavy, function(x) x$coverage))
p_values_heavy <- sapply(results_heavy, function(x) x$p_values)
type1_heavy <- rowMeans(p_values_heavy < 0.05)

par(mfrow = c(1, 2))
hist(t_stats_matrix[2, ], breaks = 50, prob = TRUE, col = "lightblue",
     main = "Normal Errors:  1",
     xlab = "t-statistic", xlim = c(-5, 5), ylim = c(0, 0.45))
curve(dt(x, df = n - p), add = TRUE, col = "red", lwd = 2)

hist(t_stats_heavy[2, ], breaks = 50, prob = TRUE, col = "lightcoral",
     main = "Heavy-Tailed Errors:  1",
     xlab = "t-statistic", xlim = c(-5, 5), ylim = c(0, 0.45))
curve(dt(x, df = n - p), add = TRUE, col = "red", lwd = 2)
```




```
par(mfrow = c(1, 1))
```

```
results_skewed <- replicate(nsim, simulate_regression(n, beta_true, "skewed"),
                            simplify = FALSE)

t_stats_skewed <- sapply(results_skewed, function(x) x$t_stats)
coverage_skewed <- rowMeans(sapply(results_skewed, function(x) x$coverage))
p_values_skewed <- sapply(results_skewed, function(x) x$p_values)
type1_skewed <- rowMeans(p_values_skewed < 0.05)

par(mfrow = c(1, 2))
hist(t_stats_matrix[2, ], breaks = 50, prob = TRUE, col = "lightblue",
     main = "Normal Errors:  1",
     xlab = "t-statistic", xlim = c(-5, 5), ylim = c(0, 0.45))
curve(dt(x, df = n - p), add = TRUE, col = "red", lwd = 2)

hist(t_stats_skewed[2, ], breaks = 50, prob = TRUE, col = "lightgreen",
     main = "Skewed Errors:  1",
```

```r
     xlab = "t-statistic", xlim = c(-5, 5), ylim = c(0, 0.45))
curve(dt(x, df = n - p), add = TRUE, col = "red", lwd = 2)
```




```r
par(mfrow = c(1, 1))
```

```r
comparison_df <- data.frame(
  Parameter = paste0(" ", 0:(p-1)),
  Coverage_Normal = coverage_rates,
  Coverage_Heavy = coverage_heavy,
  Coverage_Skewed = coverage_skewed,
  Type1_Normal = type1_errors,
  Type1_Heavy = type1_heavy,
  Type1_Skewed = type1_skewed
)
comparison_df
```

```
   Parameter Coverage_Normal Coverage_Heavy Coverage_Skewed Type1_Normal
```

```
X1         0          0.9496         0.9524          0.9348       0.0504
X2         1          0.9493         0.9504          0.9522       0.0507
X3         2          0.9519         0.9446          0.9470       0.0481
X4         3          0.9491         0.9507          0.9502       0.0509
   Type1_Heavy Type1_Skewed
X1      0.0476       0.0652
X2      0.0496       0.0478
X3      0.0554       0.0530
X4      0.0493       0.0498
```

Summary: Heavy-tailed errors (symmetric) show minimal impact on coverage and Type I error rates, while skewed errors cause larger deviations from nominal levels. **Conclusion:** Skewness is more problematic than heavy tails for regression inference.

Problem 4 (15 pts)

This problem concerns the `divusa` data in the `faraway` library.

a) Make a well-constructed visualization showing how divorce rate is changing over time. Does it appear to be steady, going up, or going down?
b) Fit a regression model with divorce as the response and remaining variables as covariates. Interpret the coefficient on `year` (include units). How can you reconcile this result with the answer to the previous part?
c) Why might observations be correlated? Make two graphical checks for correlated errors. What do you conclude?
d) Conduct a statistical test the presence of autocorrelation.

Part (a): Visualization of Divorce Rate Over Time

```r
library(faraway)
data(divusa)

str(divusa)
```

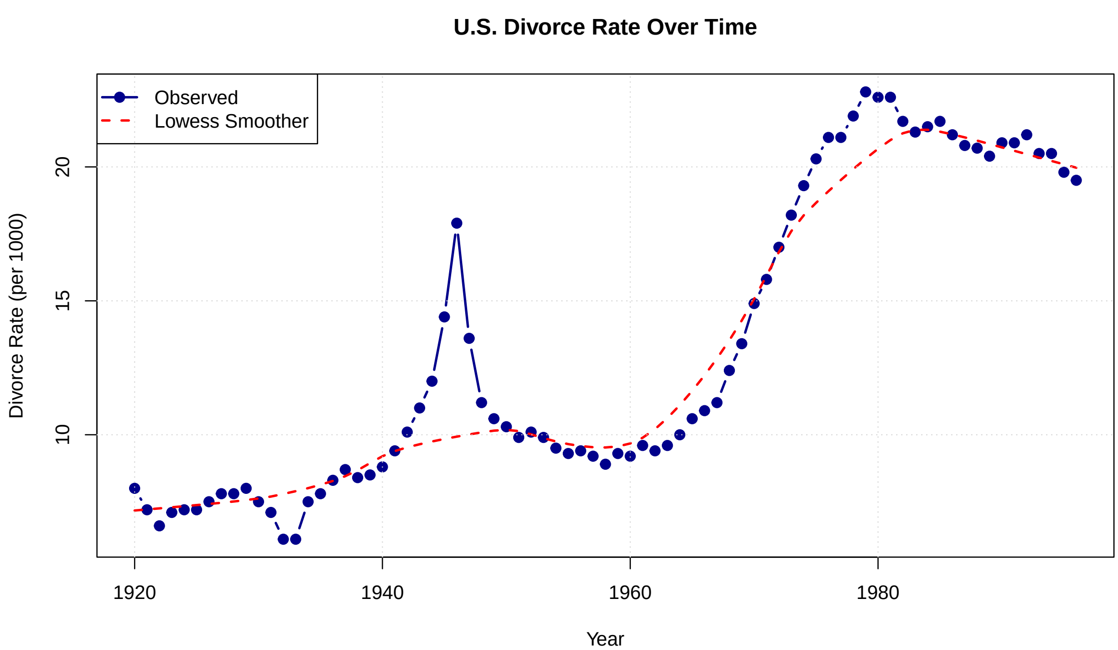
```
'data.frame':   77 obs. of  7 variables:
 $ year      : int  1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 ...
 $ divorce   : num  8 7.2 6.6 7.1 7.2 7.2 7.5 7.8 7.8 8 ...
 $ unemployed: num  5.2 11.7 6.7 2.4 5 3.2 1.8 3.3 4.2 3.2 ...
 $ femlab    : num  22.7 22.8 22.9 23 23.1 ...
 $ marriage  : num  92 83 79.7 85.2 80.3 79.2 78.7 77 74.1 75.5 ...
 $ birth     : num  118 120 111 110 111 ...
```

```
 $ military  : num  3.22 3.56 2.46 2.21 2.29 ...
```

```r
head(divusa)
```

```
  year divorce unemployed femlab marriage birth military
1 1920     8.0        5.2  22.70     92.0 117.9   3.2247
2 1921     7.2       11.7  22.79     83.0 119.8   3.5614
3 1922     6.6        6.7  22.88     79.7 111.2   2.4553
4 1923     7.1        2.4  22.97     85.2 110.5   2.2065
5 1924     7.2        5.0  23.06     80.3 110.9   2.2889
6 1925     7.2        3.2  23.15     79.2 106.6   2.1735
```

```r
plot(divusa$year, divusa$divorce, type = "b", pch = 19, col = "darkblue",
     xlab = "Year", ylab = "Divorce Rate (per 1000)",
     main = "U.S. Divorce Rate Over Time",
     lwd = 2)

lines(lowess(divusa$year, divusa$divorce, f = 0.3), col = "red", lwd = 2, lty = 2)

grid()

legend("topleft",
       legend = c("Observed", "Lowess Smoother"),
       col = c("darkblue", "red"),
       lty = c(1, 2), pch = c(19, NA), lwd = 2)
```



Observation: The divorce rate appears to show an overall **increasing trend** from the 1950s, peaking around the late 1970s to early 1980s, followed by a **decline** in more recent years. The pattern is not linear - it's more of an inverted U-shape with considerable year-to-year variation.

Part (b): Regression Model

```
model_div <- lm(divorce ~ year + unemployed + femlab + marriage + birth + military,
                data = divusa)
summary(model_div)
```

```
Call:
lm(formula = divorce ~ year + unemployed + femlab + marriage +
    birth + military, data = divusa)

Residuals:
    Min      1Q  Median      3Q     Max
-2.9087 -0.9212 -0.0935  0.7447  3.4689

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 380.14761   99.20371   3.832 0.000274 ***
year         -0.20312    0.05333  -3.809 0.000297 ***
unemployed   -0.04933    0.05378  -0.917 0.362171
femlab        0.80793    0.11487   7.033 1.09e-09 ***
marriage      0.14977    0.02382   6.287 2.42e-08 ***
birth        -0.11695    0.01470  -7.957 2.19e-11 ***
military     -0.04276    0.01372  -3.117 0.002652 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.513 on 70 degrees of freedom
Multiple R-squared:  0.9344,    Adjusted R-squared:  0.9288
F-statistic: 166.2 on 6 and 70 DF,  p-value: < 2.2e-16
```

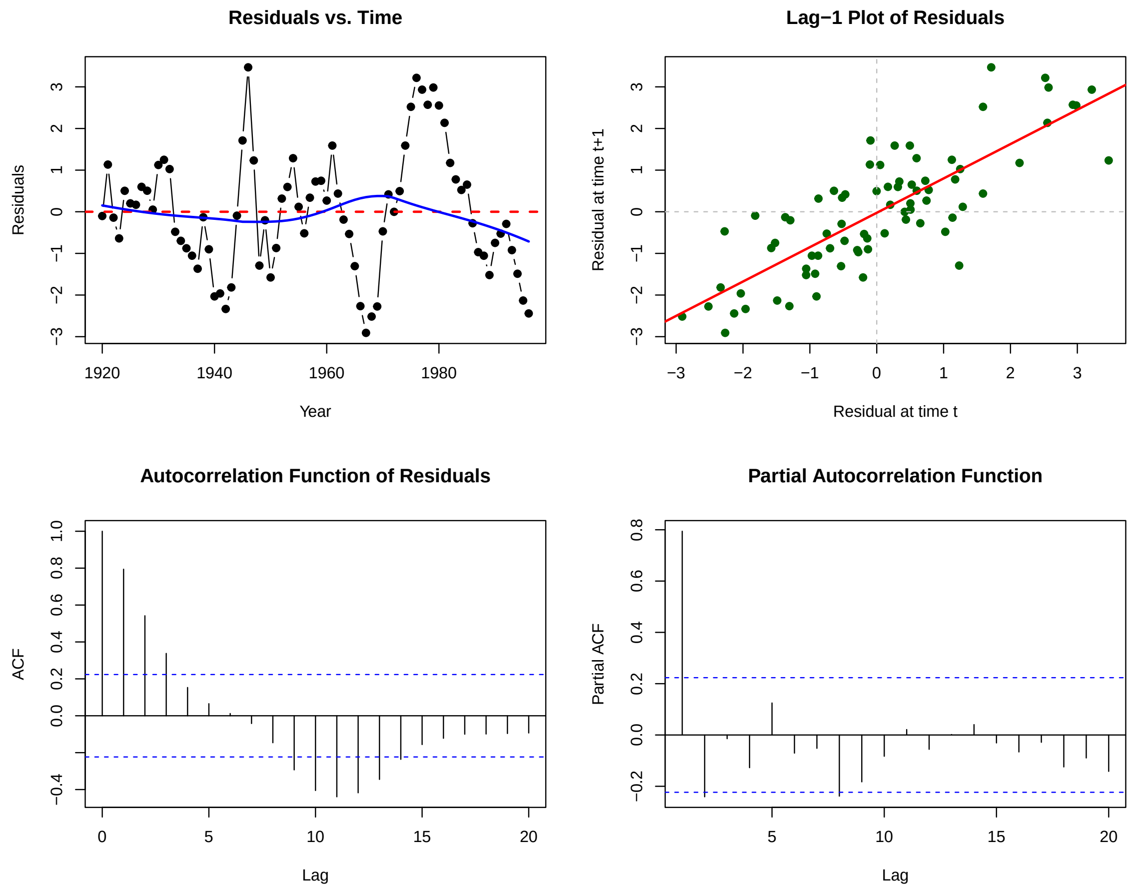
Interpretation: The `year` coefficient is **-0.2031** per 1000 population per year (or **-2.031** per 1000 over 10 years), holding other variables constant.

Reconciliation with Part (a): The linear coefficient may differ from the visual inverted-U pattern due to confounding variables (unemployment, female labor force, marriage/birth rates) and the model's inability to capture non-linear trends.

Part (c): Checking for Correlated Errors

Why might observations be correlated? Time series data exhibit autocorrelation due to temporal persistence of social trends and slowly-changing omitted variables (cultural attitudes, economic conditions).

```r
resids <- residuals(model_div)

par(mfrow = c(2, 2))

plot(divusa$year, resids, type = "b", pch = 19,
     xlab = "Year", ylab = "Residuals",
     main = "Residuals vs. Time")
abline(h = 0, col = "red", lty = 2, lwd = 2)
lines(lowess(divusa$year, resids), col = "blue", lwd = 2)

plot(resids[-length(resids)], resids[-1],
     pch = 19, col = "darkgreen",
     xlab = "Residual at time t", ylab = "Residual at time t+1",
     main = "Lag-1 Plot of Residuals")
abline(h = 0, v = 0, col = "gray", lty = 2)
abline(lm(resids[-1] ~ resids[-length(resids)]), col = "red", lwd = 2)

acf(resids, main = "Autocorrelation Function of Residuals", lag.max = 20)

pacf(resids, main = "Partial Autocorrelation Function", lag.max = 20)
```




```r
par(mfrow = c(1, 1))
```

Conclusions:

1. **Residuals vs. Time:** Shows clear patterns and runs of positive/negative residuals, suggesting temporal correlation
2. **Lag-1 Plot:** Positive slope indicates that consecutive residuals are correlated - when one residual is positive, the next tends to be positive as well
3. **ACF Plot:** Multiple significant autocorrelations (beyond the dashed blue lines) at various lags, confirming substantial serial correlation
4. **PACF Plot:** Helps identify the order of autoregressive structure

Overall: Strong evidence of correlated errors, violating the independence assumption of OLS regression.

Part (d): Statistical Test for Autocorrelation

```r
library(lmtest)
dw_test <- dwtest(model_div)
dw_test
```

```

    Durbin-Watson test

data:  model_div
DW = 0.37429, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

```r
bg_test <- bgtest(model_div, order = 3)
bg_test
```

```

    Breusch-Godfrey test for serial correlation of order up to 3

data:  model_div
LM test = 54.414, df = 3, p-value = 9.158e-12
```

```r
box_test <- Box.test(resids, lag = 10, type = "Ljung-Box")
box_test
```

```

    Box-Ljung test

data:  resids
X-squared = 110.62, df = 10, p-value < 2.2e-16
```

Test Results:

- **Durbin-Watson:** DW = 0.3743, p = 0 — significant autocorrelation
- **Breusch-Godfrey:** LM = 54.4137, p = 0 — significant autocorrelation
- **Box-Ljung:** ^{2} = 110.6194, p = 0 — significant autocorrelation

Conclusion: Strong evidence of autocorrelation; OLS standard errors are underestimated and inference is unreliable. Consider time series methods (GLS, ARIMA).