Homework 3

Tiffany Tu

```r
# load packages
library(MASS)       # for mvrnorm
library(ggplot2)    # for plots
library(sandwich)   # for robust vcov estimators
library(lmtest)     # for coeftest, waldtest
library(ISLR)       # for Hitters data

# helper: nice printing
round_mat <- function(m, d=2) {
  as.data.frame(round(m, d))
}

round_mat4 <- function(m, d=4) {
  as.data.frame(round(m, d))
}

# simulation helper for Question 1 / 2
simdat <- function(mu, sd, rho, n, beta, sigma) {
  # construct covariance matrix from sd and rho (2x2 case)
  cv <- rho*sd[1]*sd[2]
  Sigma <- matrix(c(sd[1]^2, cv,
                    cv,         sd[2]^2),
                  2, 2)
  x <- MASS::mvrnorm(n, mu, Sigma)
  y <- cbind(1, x) %*% beta + rnorm(n, 0, sigma)
  return(data.frame(y, x1 = x[,1], x2 = x[,2]))
}
```

Question 1: Variance-Covariance Matrix of the OLS estimator vector

We have $X = [1, X_1, X_2]'$.

Assumptions:

- $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ with $\mu_1 = 3$ and $\sigma_1^2 = 4$
- $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ with $\mu_2 = 2$ and $\sigma_2^2 = 6$
- $X_1 \perp X_2$ (independent)
- $Y = X'\beta + e$ with $\beta = [5, 0.4, 0.2]'$
- $e \sim \mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 10$ and $\mathbb{E}[e|X] = 0$

Recall: $\text{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 \quad \Rightarrow \quad \mathbb{E}[X^2] = \text{Var}(X) + (\mathbb{E}[X])^2$.

For X_1: - $\mathbb{E}[X_1] = 3$ - $\text{Var}(X_1) = 4$ - $\mathbb{E}[X_1^2] = 4 + 3^2 = 13$

For X_2: - $\mathbb{E}[X_2] = 2$ - $\text{Var}(X_2) = 6$ - $\mathbb{E}[X_2^2] = 6 + 2^2 = 10$

Independence gives $\mathbb{E}[X_1 X_2] = \mathbb{E}[X_1]\mathbb{E}[X_2] = 3 \cdot 2 = 6$.

So

$$Q_{XX} = \mathbb{E}[XX'] = \begin{bmatrix} 1 & \mathbb{E}[X_1] & \mathbb{E}[X_2] \\ \mathbb{E}[X_1] & \mathbb{E}[X_1^2] & \mathbb{E}[X_1 X_2] \\ \mathbb{E}[X_2] & \mathbb{E}[X_1 X_2] & \mathbb{E}[X_2^2] \end{bmatrix} = \begin{bmatrix} 1 & 3 & 2 \\ 3 & 13 & 6 \\ 2 & 6 & 10 \end{bmatrix}.$$

1.a) Answer

$$Q_{XX} = \begin{bmatrix} 1 & 3 & 2 \\ 3 & 13 & 6 \\ 2 & 6 & 10 \end{bmatrix}.$$

1.b)

Task: - Set `set.seed(1234)` - Simulate $n = 100$ obs using the given DGP
- where `mu = c(3,2)`, `sd = c(2, sqrt(6))`, `rho = 0`, `beta = c(5,0.4,0.2)`, `sigma = sqrt(10)` - Compute $\hat{Q}_{XX} = \frac{1}{n} X'X$ for $X = [1, x1, x2]$ - Round to 2 decimals - Compare to Q_{XX} above

```
set.seed(1234)
mu_vec    <- c(3, 2)
sd_vec    <- c(2, sqrt(6))   # sd = sqrt(var)
rho_val   <- 0               # independence
n_100     <- 100
beta_vec  <- c(5, 0.4, 0.2)
sigma_val <- sqrt(10)

dat_100 <- simdat(mu = mu_vec,
                  sd = sd_vec,
                  rho = rho_val,
                  n = n_100,
                  beta = beta_vec,
                  sigma = sigma_val)

X_100 <- cbind(1, dat_100$x1, dat_100$x2)
Qhat_100 <- (t(X_100) %*% X_100) / n_100

Q_xx_true <- matrix(c(1,3,2,
                      3,13,6,
                      2,6,10),
                    nrow=3, byrow=TRUE)

round_Qhat_100   <- round_mat(Qhat_100, 2)
round_Q_xx_true  <- round_mat(Q_xx_true, 2)

round_Qhat_100
```

```
    V1    V2   V3
1 1.00  2.92 1.62
2 2.92 12.73 4.84
3 1.62  4.84 8.60
```

```
round_Q_xx_true
```

```
  V1 V2 V3
1  1  3  2
2  3 13  6
3  2  6 10
```

We expect $\hat{Q}_{XX}$ to be reasonably close but not perfect with $n = 100$.

1.c)

Now: - `set.seed(2345)` - `n = 1000` (everything else same) - Compute and compare again

```
set.seed(2345)
n_1000 <- 1000

dat_1000 <- simdat(mu = mu_vec,
                   sd = sd_vec,
                   rho = rho_val,
                   n = n_1000,
                   beta = beta_vec,
                   sigma = sigma_val)

X_1000 <- cbind(1, dat_1000$x1, dat_1000$x2)
Qhat_1000 <- (t(X_1000) %*% X_1000) / n_1000

round_Qhat_1000 <- round_mat(Qhat_1000, 2)
round_Qhat_1000
```

```
   V1    V2   V3
1 1.0  3.00 1.90
2 3.0 12.73 5.73
3 1.9  5.73 9.47
```

```
round_Q_xx_true
```

```
  V1 V2 V3
1  1  3  2
2  3 13  6
3  2  6 10
```

With $n = 1000$, $\hat{Q}_{XX}$ should be even closer to Q_{XX} by the Law of Large Numbers.

1.d)

We want

$$V_\beta = \sigma^2 Q_{XX}^{-1}$$

using the *true* Q_{XX} from 1(a), and $\sigma^2 = 10$.

We'll: 1. Create `Q_xx_true` 2. Invert it 3. Multiply by σ^2 4. Round to 2 decimals

```
sigma2_true <- 10
Vbeta_theory <- sigma2_true * solve(Q_xx_true)
round_mat(Vbeta_theory, 2)
```

```
     V1   V2    V3
1 39.17 -7.5 -3.33
2 -7.50  2.5  0.00
3 -3.33  0.0  1.67
```

This matrix is the theoretical finite-sample variance of OLS under our assumptions.

1.e)

Now we use the simulated data from **1(c)** ($n = 1000$):

For homoskedastic OLS,

$$\widehat{V}_\beta = s^2 (X'X)^{-1}$$

where

$$s^2 = \frac{1}{n-k} \sum_i \hat{e}_i^2$$

and $k = 3$ parameters (intercept + 2 regressors).

Steps: 1. Regress `y` on `x1` and `x2` using `dat_1000` 2. Grab `(X'X)^{-1}` from that regression 3. Compute s^2 4. Form $\widehat{V}_\beta$ 5. Round to 4 decimals

```
fit_1000 <- lm(y ~ x1 + x2, data = dat_1000)

# design matrix and inverse XtX
Xmat <- model.matrix(fit_1000)
XtX_inv <- solve(t(Xmat) %*% Xmat)
```

```r
n <- nrow(Xmat)
k <- ncol(Xmat)

resid_vec <- resid(fit_1000)
s2_hat <- sum(resid_vec^2)/(n - k)

Vbeta_hat <- as.numeric(s2_hat) * XtX_inv

round_mat4(Vbeta_hat, 4)
```

```
            (Intercept)      x1      x2
(Intercept)      0.0402 -0.0080 -0.0032
x1              -0.0080  0.0027  0.0000
x2              -0.0032  0.0000  0.0017
```

1.f)

Now multiply $\widehat{V}_\beta$ from 1(e) by n:

$$n \cdot \widehat{V}_\beta \approx V_\beta$$

This should be close to the result in 1(d).
Round to 2 decimals.

```r
Vbeta_hat_scaled <- n * Vbeta_hat
round_mat(Vbeta_hat_scaled, 2)
```

```
            (Intercept)    x1    x2
(Intercept)       40.23 -8.04 -3.22
x1                -8.04  2.68 -0.01
x2                -3.22 -0.01  1.70
```

```r
round_mat(Vbeta_theory, 2)  # from 1(d), for comparison
```

```
     V1   V2    V3
1 39.17 -7.5 -3.33
2 -7.50  2.5  0.00
3 -3.33  0.0  1.67
```

We compare `n * Vbeta_hat` to $\sigma^2 Q_{XX}^{-1}$ from 1(d). They should be quite similar because $X'X/n \to Q_{XX}$ and $s^2 \to \sigma^2$ as n grows.

Question 2: Confidence Intervals

We are told:

- Use
  `mu = c(1,4)`
  `sd = c(1,2)`
  `rho = 0.3`
  `n = 100`
  `beta = c(0.5, 1.5, 3.0)` so $\beta = [\beta_0, \beta_1, \beta_2]'$
  `sigma = 2`
- Regress `y ~ x1 + x2`
- Assume homoskedasticity
- $\alpha = 0.20 \Rightarrow 80\%$ CI
- For each repetition $r = 1, \ldots, 100$ we compute
  1. the 80% CI for β_1
  2. $T^{(r)}(1.5) = (\hat{\beta}_1 - 1.5)/SE(\hat{\beta}_1)$

Use `set.seed(3456)` **once** at the start, then repeat simulation 100 times.

We'll store: - lower and upper CI bounds for β_1 - indicator for whether CI covers true $\beta_1 = 1.5$ - test statistic $T(1.5)$ - indicator for reject $H_0 : \beta_1 = 1.5$ at $\alpha = 0.20$
(two-sided test $\to$ critical value $= t_{0.9,df=97}$ because n=100, k=3 $\to$ df=97, and tail prob=0.1 each side for 80% CI)

```
set.seed(3456)

n_sim    <- 100
n_obs    <- 100
mu_q2    <- c(1,4)
sd_q2    <- c(1,2)
rho_q2   <- 0.3
beta_q2  <- c(0.5, 1.5, 3.0)
sigma_q2 <- 2
```

```
results <- data.frame(iter = 1:n_sim,
                      beta1_hat = NA,
                      se_beta1  = NA,
                      ci_low    = NA,
                      ci_high   = NA,
                      covers    = NA,
                      Tstat     = NA,
                      reject    = NA)

for (r in 1:n_sim) {
  d <- simdat(mu = mu_q2,
              sd = sd_q2,
              rho = rho_q2,
              n = n_obs,
              beta = beta_q2,
              sigma = sigma_q2)

  fit <- lm(y ~ x1 + x2, data = d)
  co  <- summary(fit)$coefficients

  bhat1 <- co["x1","Estimate"]
  se1   <- co["x1","Std. Error"]

  df    <- n_obs - 3  # 100 - (intercept,x1,x2)
  alpha <- 0.20
  tcrit <- qt(1 - alpha/2, df = df)  # qt(0.9, 97)

  ci_low  <- bhat1 - tcrit * se1
  ci_high <- bhat1 + tcrit * se1

  # T(1.5)
  Tstat <- (bhat1 - 1.5)/se1

  # reject H0: beta1=1.5 at alpha=0.20 two-sided?
  # reject if |T| > tcrit
  reject_null <- (abs(Tstat) > tcrit)

  results$beta1_hat[r] <- bhat1
  results$se_beta1[r]  <- se1
  results$ci_low[r]    <- ci_low
  results$ci_high[r]   <- ci_high
  results$covers[r]    <- (ci_low <= 1.5 & ci_high >= 1.5)
```

```r
  results$Tstat[r]      <- Tstat
  results$reject[r]     <- reject_null
}

head(results)
```

```
  iter beta1_hat  se_beta1    ci_low  ci_high covers      Tstat reject
1    1  1.431959 0.1721402 1.2098393 1.654078   TRUE -0.3952670  FALSE
2    2  1.144476 0.2381678 0.8371584 1.451793  FALSE -1.4927459   TRUE
3    3  1.473111 0.1860596 1.2330312 1.713192   TRUE -0.1445162  FALSE
4    4  1.574693 0.2070059 1.3075845 1.841801   TRUE  0.3608234  FALSE
5    5  1.547270 0.2147985 1.2701070 1.824433   TRUE  0.2200676  FALSE
6    6  1.892938 0.2420612 1.5805966 2.205279  FALSE  1.6232999   TRUE
```

```r
summary(results)
```

```
      iter          beta1_hat         se_beta1          ci_low
 Min.   :  1.00   Min.   :0.8028   Min.   :0.1721   Min.   :0.5019
 1st Qu.: 25.75   1st Qu.:1.3684   1st Qu.:0.1980   1st Qu.:1.0934
 Median : 50.50   Median :1.4816   Median :0.2136   Median :1.2266
 Mean   : 50.50   Mean   :1.4821   Mean   :0.2119   Mean   :1.2086
 3rd Qu.: 75.25   3rd Qu.:1.5877   3rd Qu.:0.2269   3rd Qu.:1.3286
 Max.   :100.00   Max.   :1.9381   Max.   :0.2579   Max.   :1.6925
    ci_high        covers             Tstat             reject
 Min.   :1.104   Mode :logical   Min.   :-2.99043   Mode :logical
 1st Qu.:1.626   FALSE:18        1st Qu.:-0.71713   FALSE:82
 Median :1.750   TRUE :82        Median :-0.09439   TRUE :18
 Mean   :1.755                   Mean   :-0.08490
 3rd Qu.:1.859                   3rd Qu.: 0.43062
 Max.   :2.223                   Max.   : 2.30113
```

```r
tcrit
```

```
[1] 1.29034
```

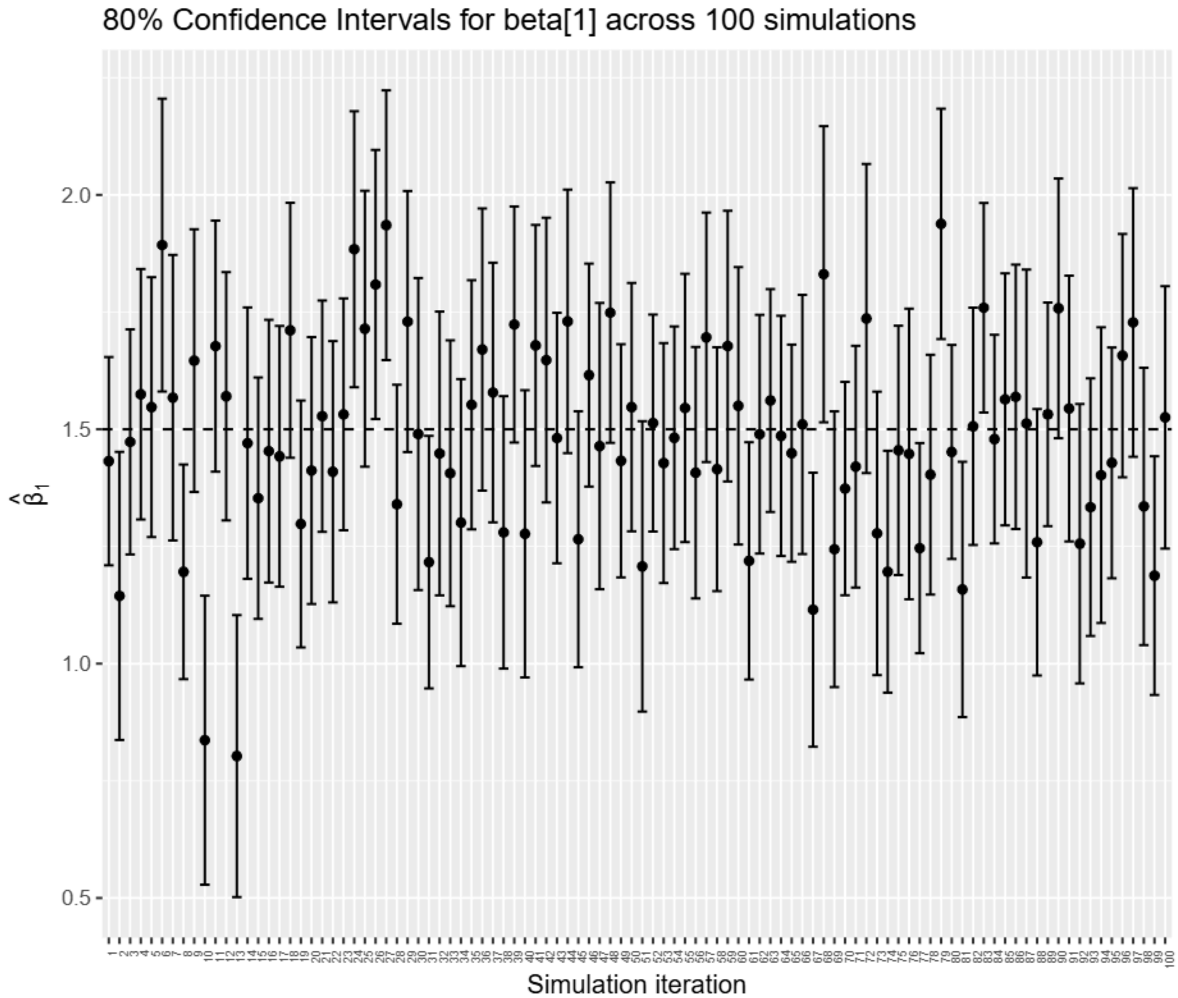
We now have: - All 100 CIs - All 100 T statistics - Coverage indicators - Rejection indicators

2.a) Plot the 100 confidence intervals

We'll make a dataframe suitable for plotting with `geom_errorbar`.
We'll also mark which ones cover the true $\beta_1 = 1.5$.

```
plot_df <- results
plot_df$iter <- factor(plot_df$iter)

ggplot(plot_df, aes(x = iter, y = beta1_hat)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low, ymax = ci_high)) +
  geom_hline(yintercept = 1.5, linetype = "dashed") +
  labs(x = "Simulation iteration",
       y = expression(hat(beta)[1]),
       title = "80% Confidence Intervals for beta[1] across 100 simulations") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size=5))
```



Answer for 2.a: The plot above shows each simulation's $\hat{\beta}_1$ with its 80% CI, and the dashed line is the true $\beta_1 = 1.5$.

2.b)

What proportion of the intervals contain the true $\beta_1 = 1.5$?

```r
coverage_rate <- mean(results$covers)
coverage_rate
```

```
[1] 0.82
```

Answer for 2.b:
The proportion of CIs that contain 1.5 is 0.82.
For an 80% CI, we expect about 0.80 in large samples.

2.c)

What proportion of tests reject $H_0 : \beta_1 = 1.5$ at $\alpha = 0.20$?

```
rejection_rate <- mean(results$reject)
rejection_rate
```

```
[1] 0.18
```

Answer for 2.c:
The rejection rate is 0.18 using a two-sided t-test at $\alpha = 0.20$.

Note the logical link: - If the 80% CI contains 1.5, we *do not* reject. - If it does not contain 1.5, we *do* reject. So rejection rate $\approx 1-$ coverage rate.

Question 3: Hypothesis Tests

Model:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + e$$

where
- Y = Salary
- X_1 = Hits
- X_2 = Years

We'll load the Hitters data (from `ISLR`), drop rows with `NA` Salary, then run `lm(Salary ~ Hits + Years)`.

We'll then compute:

- 3.a) t-stat and p-value for $H_0 : \beta_0 = 0$ under *Normal errors*
- 3.b) z-stat and p-value for $H_0 : \beta_0 = 0$ using asymptotic Normal (homoskedastic)
- 3.c) z-stat and p-value for $H_0 : \beta_0 = 0$ using asymptotic Normal with *heteroskedasticity-robust* (HC1) SEs
- 3.d) F-test (and p-value) for joint $H_0 : \beta_1 = 0$ and $\beta_2 = 0$
- 3.e) Test $H_0 : 6\beta_1 = \beta_2$ at $\alpha = 0.05$, Normal errors

- 3.f) Test joint $H_0 : \beta_1 = 6$ and $\beta_2 = 30$ at $\alpha = 0.05$, Normal errors

```
data(Hitters, package="ISLR")
Hitters <- Hitters[!is.na(Hitters$Salary), ]

fit_hit <- lm(Salary ~ Hits + Years, data = Hitters)
summ_hit <- summary(fit_hit)
summ_hit
```

```
Call:
lm(formula = Salary ~ Hits + Years, data = Hitters)

Residuals:
    Min      1Q  Median      3Q     Max 
-838.06 -211.89  -43.40   76.07 2248.37 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -199.2510    67.4690  -2.953  0.00343 ** 
Hits           4.3124     0.5013   8.603 7.46e-16 ***
Years         36.9501     4.7187   7.831 1.24e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 366.1 on 260 degrees of freedom
Multiple R-squared:  0.3465,    Adjusted R-squared:  0.3415 
F-statistic: 68.94 on 2 and 260 DF,  p-value: < 2.2e-16
```

We'll extract coefficient estimates, standard errors, df, etc.

```
coefs <- coef(summ_hit)
coefs
```

```
               Estimate Std. Error   t value     Pr(>|t|)
(Intercept) -199.250976 67.4689750 -2.953224 3.432509e-03
Hits           4.312438  0.5012647  8.603116 7.461315e-16
Years         36.950116  4.7187203  7.830537 1.236069e-13
```

```
df_resid <- df.residual(fit_hit)
df_resid
```

```
[1] 260
```

We'll also get heteroskedasticity-robust (HC1) SEs:

```
vcov_hc1 <- vcovHC(fit_hit, type="HC1")
robust_se <- sqrt(diag(vcov_hc1))
robust_se
```

```
(Intercept)        Hits       Years
 96.7030610   0.7548996   4.9022273
```

3.a)

Hypothesis:
$H_0 : \beta_0 = 0$ vs $H_1 : \beta_0 \neq 0$

Under *classical linear model with Normal errors*,
the test statistic is the usual t-statistic:

$$t = \frac{\hat{\beta}_0 - 0}{SE(\hat{\beta}_0)}$$

with df = residual df.

Two-sided p-value uses the t distribution.

```
b0_hat    <- coefs["(Intercept)","Estimate"]
se_b0     <- coefs["(Intercept)","Std. Error"]
tstat_b0  <- (b0_hat - 0)/se_b0
pval_b0_t <- 2 * (1 - pt(abs(tstat_b0), df = df_resid))

tstat_b0
```

```
[1] -2.953224
```

```
pval_b0_t
```

```
[1] 0.003432509
```

Answer 3.a:
- t-statistic for $H_0 : \beta_0 = 0$ is -2.9532
- two-sided p-value (t dist, df = 260) is 0.003433

3.b)

Now treat the estimator as asymptotically Normal under homoskedasticity (no Normality of e assumption).
Then we form a z-statistic:

$$z = \frac{\hat{\beta}_0 - 0}{SE(\hat{\beta}_0)}$$

and p-value from standard Normal.

This is numerically the same numerator/denominator as in 3.a, but we reference $\mathcal{N}(0, 1)$ instead of t_{df}.

```
zstat_b0_homosked <- tstat_b0  # same ratio
pval_b0_z <- 2 * (1 - pnorm(abs(zstat_b0_homosked)))

zstat_b0_homosked
```

```
[1] -2.953224
```

```
pval_b0_z
```

```
[1] 0.00314474
```

Answer 3.b:
- z-statistic: -2.9532
- two-sided p-value (Normal reference): 0.003145

3.c)

Now allow heteroskedasticity.
Use HC1 robust SE for the intercept.

$$z = \frac{\hat{\beta}_0 - 0}{SE_{HC1}(\hat{\beta}_0)}.$$

p-value from standard Normal.

```
se_b0_hc1 <- robust_se[["(Intercept)"]]
zstat_b0_hc1 <- (b0_hat - 0)/se_b0_hc1
pval_b0_hc1  <- 2 * (1 - pnorm(abs(zstat_b0_hc1)))

se_b0_hc1
```

```
[1] 96.70306
```

```
zstat_b0_hc1
```

```
[1] -2.060441
```

```
pval_b0_hc1
```

```
[1] 0.03935638
```

Answer 3.c:
- Robust (HC1) SE for $\hat{\beta}_0$: 96.7031
- z-statistic (HC1): -2.0604
- two-sided p-value (Normal ref): 0.03936

3.d)

We now test:

$$H_0 : \beta_1 = 0 \quad \text{and} \quad \beta_2 = 0$$

This is a standard joint F-test comparing: - unrestricted model: `Salary ~ Hits + Years` - restricted model: `Salary ~ 1` (intercept only)

We can use `anova(restricted, unrestricted)`.

```
fit_restrict <- lm(Salary ~ 1, data = Hitters)

anova_res <- anova(fit_restrict, fit_hit)
anova_res
```

```
Analysis of Variance Table

Model 1: Salary ~ 1
Model 2: Salary ~ Hits + Years
  Res.Df      RSS Df Sum of Sq      F    Pr(>F)
1    262 53319113
2    260 34841700  2  18477413 68.942 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We'll pull the F-statistic and p-value.

```
F_stat <- anova_res$F[2]
p_val_F <- anova_res$`Pr(>F)`[2]

F_stat
```

```
[1] 68.94221
```

```
p_val_F
```

```
[1] 9.509004e-25
```

Answer 3.d:
- F-statistic for $H_0 : \beta_1 = \beta_2 = 0$ is 68.9422
- p-value is 9.509×10^{-25}

3.e)

Test linear hypothesis:

$$H_0 : 6\beta_1 = \beta_2 \quad \Leftrightarrow \quad 6\beta_1 - \beta_2 = 0.$$

Assume Normal errors → use classic linear restriction test (1 restriction → equivalent to a t^2 test).

We'll build matrix R and vector r for $R\beta = r$ with
$R = [0, 6, -1]$ (corresponding to $(\beta_0, \beta_1, \beta_2)$)
and $r = 0$.

The Wald statistic for 1 linear restriction is:

$$W = \frac{(R\hat{\beta} - r)^2}{R\widehat{\text{Var}}(\hat{\beta})R'}.$$

Under Normal errors, W is $F_{1,df}$ (or t^2). We'll compute and get p-value.

```r
bhat <- coef(fit_hit)                    # (Intercept), Hits, Years
vcov_homo <- vcov(fit_hit)               # homoskedastic OLS Var

R <- matrix(c(0, 6, -1), nrow=1)         # 1 x 3
r_vec <- 0

num <- R %*% bhat - r_vec                # scalar
den <- R %*% vcov_homo %*% t(R)          # 1x1 matrix
W <- as.numeric( (num)^2 / den )         # F-stat with 1 df in numerator
p_val_W <- 1 - pf(W, df1=1, df2=df_resid)

W
```

```
[1] 3.852612
```

```r
p_val_W
```

```
[1] 0.05073473
```

```r
# Also t-stat
t_linear <- as.numeric(num / sqrt(den))
p_val_t_two_sided <- 2*(1-pt(abs(t_linear), df=df_resid))

t_linear
```

```
[1] -1.962807
```

```
p_val_t_two_sided
```

```
[1] 0.05073473
```

Answer 3.e:
- The test statistic (as t) is -1.9628
- Two-sided p-value using $t_{df}‘ is 0.05073
- At $\alpha = 0.05$, we reject if p-value < 0.05.

3.f)

Now test the joint hypotheses:

$$H_0 : \begin{cases} \beta_1 = 6 \\ \beta_2 = 30 \end{cases}$$

That's 2 linear restrictions: - $\beta_1 - 6 = 0$ - $\beta_2 - 30 = 0$

We set

$$R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad r = \begin{bmatrix} 6 \\ 30 \end{bmatrix}.$$

Wald statistic:

$$W = (R\hat{\beta} - r)'[R\widehat{\text{Var}}(\hat{\beta})R']^{-1}(R\hat{\beta} - r),$$

with df1 $= 2$, df2 $=$ residual df. We'll compute W and its p-value.

```
R2 <- rbind(c(0,1,0),
            c(0,0,1))
r2 <- c(6,30)

diff_vec <- R2 %*% bhat - r2
V_restr  <- R2 %*% vcov_homo %*% t(R2)
W_joint  <- t(diff_vec) %*% solve(V_restr) %*% diff_vec
W_joint  <- as.numeric(W_joint)

df1 <- nrow(R2)        # 2 restrictions
df2 <- df_resid
```

```r
p_val_joint <- 1 - pf(W_joint, df1=df1, df2=df2)

W_joint
```

```
[1] 13.32359
```

```r
p_val_joint
```

```
[1] 3.099817e-06
```

Answer 3.f:
- Joint Wald F-statistic for $H_0 : \beta_1 = 6,\ \beta_2 = 30$ is 13.3236 with df1 = 2 and df2 = 260.
- p-value is 3.1×10^{-6}.
- At $\alpha = 0.05$, reject if p-value < 0.05.

Wrap-up

- **Q1**: We derived Q_{XX} analytically, confirmed sample $\hat{Q}_{XX}$ approaches it, computed theoretical $V_\beta = \sigma^2 Q_{XX}^{-1}$, estimated $\widehat{V}_\beta$ from simulated data, and showed $n \cdot \widehat{V}_\beta$ lines up with theory.
- **Q2**: We simulated 100 datasets, built 80% CIs for β_1, plotted them, computed empirical coverage and rejection rates for $H_0 : \beta_1 = 1.5$ at $\alpha = 0.20$.
- **Q3**: We ran OLS on `Hitters`, then computed t/z tests for $\beta_0 = 0$ under different assumptions, plus F/Wald tests for joint linear hypotheses on slope coefficients.