Homework 2

Tiffany Tu

2025-10-15

1 Setup

```
suppressPackageStartupMessages({
  library(tidyverse)
  library(MASS)
  library(ISLR)
  library(sandwich)
  library(lmtest)
  library(broom)
})
theme_set(theme_minimal())
```

2 Question 1: Covariance

2.1 1.a) Population identity

We want to show that

$$\operatorname{cov}(X,Y) = \mathbb{E}[(X-\mu_X)(Y-\mu_Y)] = \mathbb{E}[(X-\mu_X)Y].$$

Expanding the right-hand side,

$$\mathbb{E}[(X-\mu_X)Y] = \mathbb{E}[XY] - \mu_X\,\mathbb{E}[Y] = \mathbb{E}[XY] - \mu_X\mu_Y,$$

and

$$\operatorname{cov}(X,Y) = \mathbb{E}\big[(X-\mu_X)(Y-\mu_Y)\big] = \mathbb{E}[XY] - \mu_X\mu_Y.$$

Hence the expressions are equal.

2.2 1.b) Sample identity

Let $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ and $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$. Then

$$(n-1)\widehat{\text{cov}}(X,Y) = \sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}).$$

Using the fact that $\sum_{i=1}^{n}(Y_i - \bar{Y}) = 0$,

$$\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}) = \sum_{i=1}^{n}(X_i - \bar{X})Y_i - \bar{Y}\sum_{i=1}^{n}(X_i - \bar{X}) = \sum_{i=1}^{n}(X_i - \bar{X})Y_i.$$

Therefore the identities are equivalent.

3 Question 2: Simpson's Paradox and the FWL Theorem

We will load **multi** from either `multi.RData` (preferred) or `multi.csv`, then standardize column names and map to `sales`, `p1`, and `p2`.

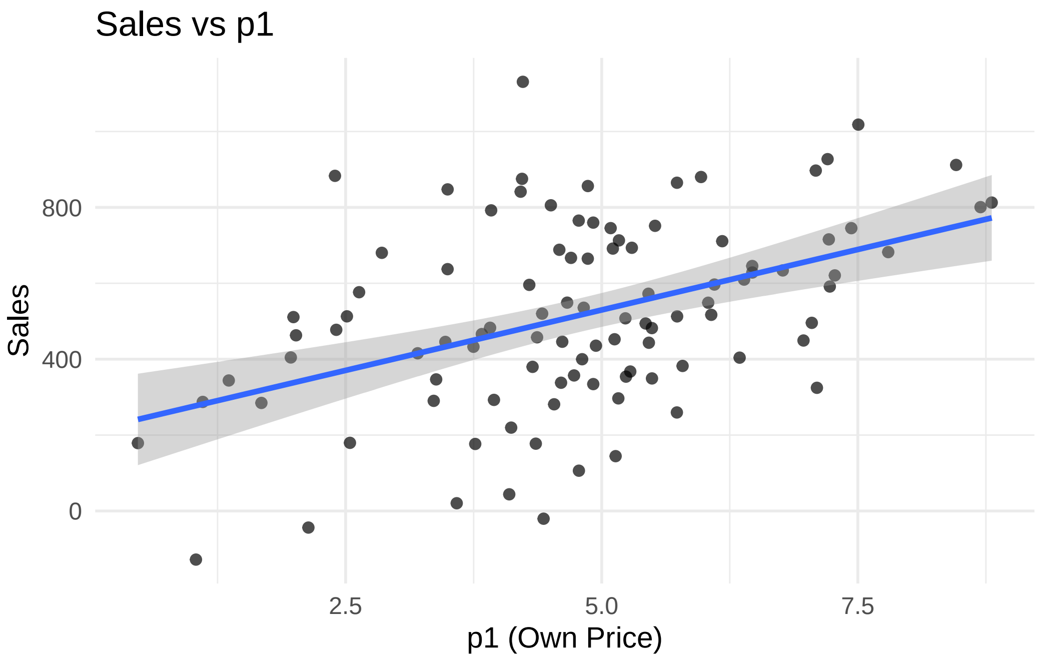
```
load_multi <- function() {
  if (file.exists("multi.RData")) {
    before <- ls()
    load("multi.RData")
    after <- ls()
    new_objs <- setdiff(after, before)
    if ("multi" %in% new_objs) obj <- get("multi") else {
      df_names <- new_objs[vapply(new_objs, function(nm) is.data.frame(get(nm)), logical(1))]
      if (length(df_names) == 0) stop("multi.RData loaded but no data.frame found.")
      obj <- get(df_names[1])
    }
    tibble::as_tibble(obj)
  } else if (file.exists("multi.csv")) {
    readr::read_csv("multi.csv", show_col_types = FALSE) |> tibble::as_tibble()
  } else {
    stop("Neither multi.RData nor multi.csv found in the working directory.")
  }
}

multi <- load_multi()
names(multi) <- tolower(names(multi))
```

```
pick_col <- function(nms, candidates) {
  for (pat in candidates) {
    hit <- grep(pat, nms, ignore.case = TRUE, value = TRUE)
    if (length(hit) > 0) return(hit[1])
  }
  NA_character_
}

nms <- names(multi)
sales_col <- if ("sales" %in% nms) "sales" else pick_col(nms, c("^sales$", "qty", "quantity"
p1_col    <- if ("p1"    %in% nms) "p1"    else pick_col(nms, c("^p1$", "price1", "own", "^x
p2_col    <- if ("p2"    %in% nms) "p2"    else pick_col(nms, c("^p2$", "price2", "comp", "co

if (is.na(sales_col) || is.na(p1_col) || is.na(p2_col)) {
  stop("Could not find required columns. Found names: ", paste(nms, collapse=", "),
       ". Need something like sales, p1, p2 (case-insensitive).")
}

multi <- multi |>
  dplyr::rename(
    sales = dplyr::all_of(sales_col),
    p1    = dplyr::all_of(p1_col),
    p2    = dplyr::all_of(p2_col)
  )

stopifnot(is.numeric(multi$sales), is.numeric(multi$p1), is.numeric(multi$p2))

glimpse(multi)
```

```
Rows: 100
Columns: 3
$ p1    <dbl> 5.135670, 3.495460, 7.275341, 4.662816, 3.584537, 5.167917, 3.38~
$ p2    <dbl> 5.204186, 8.059732, 11.675979, 8.364421, 2.150292, 10.153037, 4.~
$ sales <dbl> 144.48788, 637.24524, 620.78693, 549.00714, 20.42542, 713.00665,~
```

3.1 2.a) Sales vs p1

```
ggplot(multi, aes(x = p1, y = sales)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = TRUE) +
  labs(title = "Sales vs p1", x = "p1 (Own Price)", y = "Sales")
```

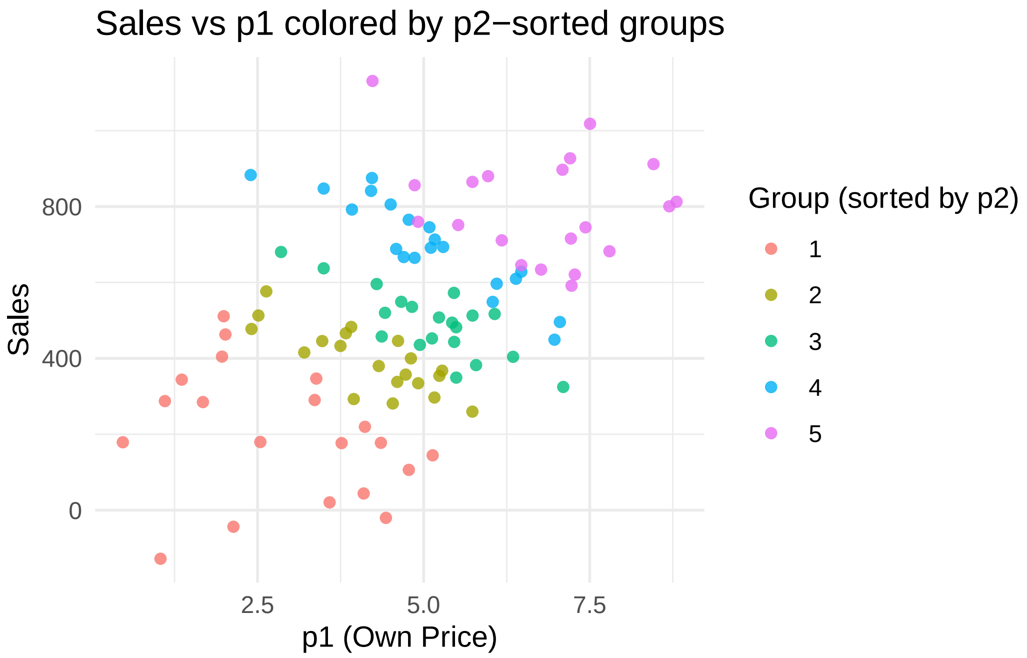


```
m1 <- lm(sales ~ p1, data = multi)
broom::tidy(m1)
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic    p.value
  <chr>          <dbl>     <dbl>     <dbl>      <dbl>
1 (Intercept)    211.       66.5      3.18 0.00200
2 p1              63.7      13.0      4.89 0.00000401
```

3.2 2.b) Grouped colors + multiple regression

```
multi_g <- multi |>
  arrange(p2) |>
  mutate(group20 = rep(1:ceiling(n()/20), each = 20, length.out = n()) |> factor())

ggplot(multi_g, aes(x = p1, y = sales, color = group20)) +
  geom_point(alpha = 0.8) +
  labs(title = "Sales vs p1 colored by p2-sorted groups",
       x = "p1 (Own Price)", y = "Sales", color = "Group (sorted by p2)")
```



```
m2 <- lm(sales ~ p1 + p2, data = multi)
broom::tidy(m2)
```

```
# A tibble: 3 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    116.       8.55      13.5 4.45e-24
2 p1             -97.7      2.67     -36.6 1.43e-58
3 p2             109.       1.41      77.2 6.80e-89
```

```
broom::glance(m2)
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
1     0.987         0.987  28.4     3717. 2.14e-92     2  -475.  958.  969.
# i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

3.3 2.c) p1 on p2

```r
m3 <- lm(p1 ~ p2, data = multi)
broom::tidy(m3)
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    1.49     0.286       5.21 1.03e- 6
2 p2             0.414    0.0332     12.5  5.92e-22
```

3.4 2.d) FWL verification

```r
r_p1 <- resid(lm(p1 ~ p2, data = multi))
fwl_df <- tibble::tibble(sales = multi$sales, r_p1 = r_p1)
m_fwl <- lm(sales ~ r_p1, data = fwl_df)
cbind(
  beta1_from_m2 = coef(m2)["p1"],
  beta_from_fwl = coef(m_fwl)["r_p1"]
)
```

```
   beta1_from_m2 beta_from_fwl
p1     -97.65737     -97.65737
```

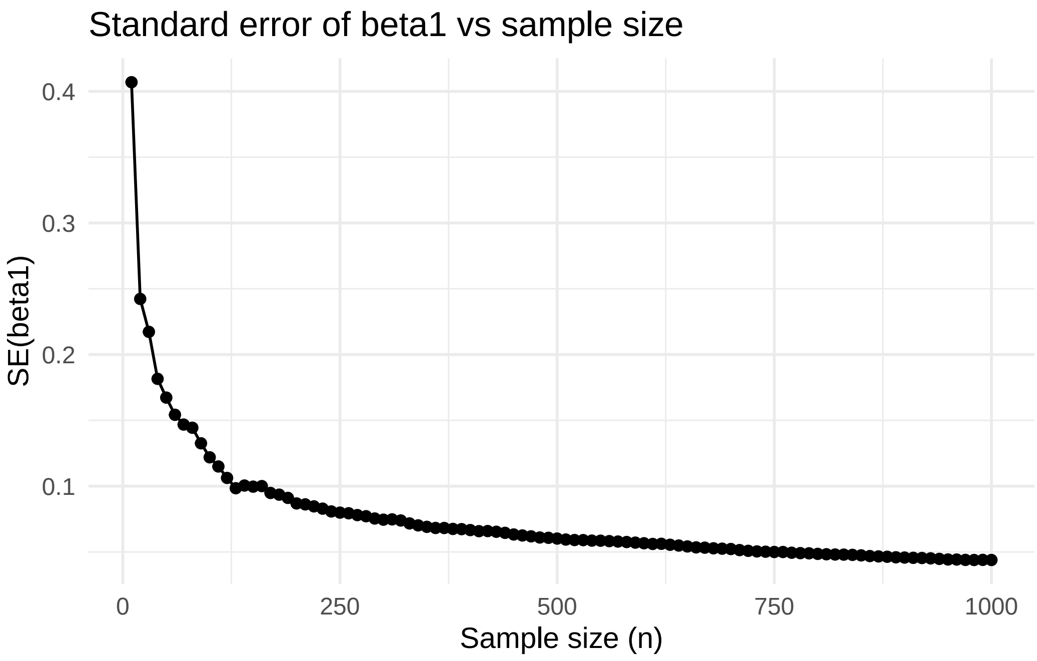
4 Question 3: Standard Errors

4.1 3.a) Generator

```r
gen_xy <- function(mu, sd, rho, n, beta) {
  stopifnot(length(mu) == 2, length(sd) == 2, length(beta) == 3)
  Sigma <- matrix(c(sd[1]^2, rho*sd[1]*sd[2],
                    rho*sd[1]*sd[2], sd[2]^2), nrow = 2, byrow = TRUE)
  X <- MASS::mvrnorm(n = n, mu = mu, Sigma = Sigma)
  X1 <- X[,1]; X2 <- X[,2]
  e  <- rnorm(n, mean = 0, sd = 2)
  Y  <- beta[1] + beta[2]*X1 + beta[3]*X2 + e
  tibble::tibble(Y = Y, X1 = X1, X2 = X2)
}
```

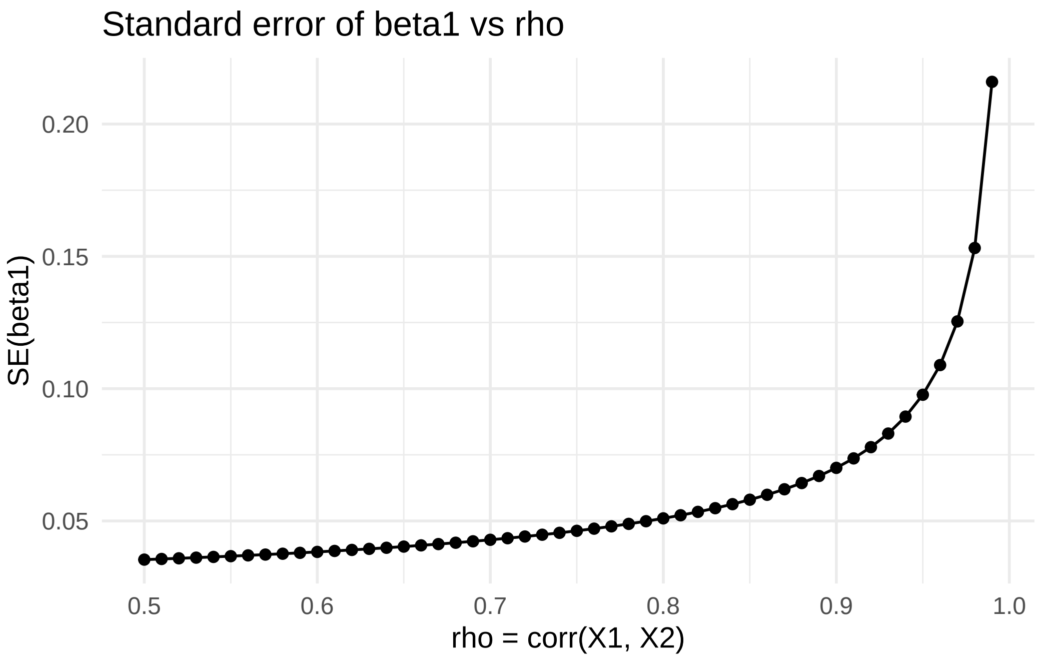
4.2 3.b) Standard error of beta1 vs sample size

```
set.seed(123)
dat <- gen_xy(mu = c(3,7), sd = c(2,3), rho = 0.7, n = 1000, beta = c(0,1,1))

ns <- seq(10, 1000, by = 10)
se_b1 <- purrr::map_dbl(ns, function(n_i) {
  fit <- lm(Y ~ X1 + X2, data = dat[1:n_i, ])
  sqrt(diag(vcov(fit)))[["X1"]]
})

tibble::tibble(n = ns, se_b1 = se_b1) |>
  ggplot(aes(x = n, y = se_b1)) +
  geom_line() + geom_point() +
  labs(title = "Standard error of beta1 vs sample size",
       x = "Sample size (n)", y = "SE(beta1)")
```



4.3 3.c) Standard error of beta1 vs corr(X1, X2)

```r
rhos <- seq(0.50, 0.99, by = 0.01)
se_b1_rho <- purrr::map_dbl(rhos, function(rh) {
  set.seed(567)
  d <- gen_xy(mu = c(3,7), sd = c(2,3), rho = rh, n = 1000, beta = c(0,1,1))
  fit <- lm(Y ~ X1 + X2, data = d)
  sqrt(diag(vcov(fit)))[["X1"]]
})

tibble::tibble(rho = rhos, se_b1 = se_b1_rho) |>
  ggplot(aes(x = rho, y = se_b1)) +
  geom_line() + geom_point() +
  labs(title = "Standard error of beta1 vs rho",
       x = "rho = corr(X1, X2)", y = "SE(beta1)")
```



5 Question 4: Homoskedasticity vs Heteroskedasticity

```
# Use base subsetting to avoid namespace issues during PDF compilation.
d <- ISLR::Hitters
d <- d[, c("Salary", "Hits", "Years")]
d <- d[!is.na(d$Salary), ]

n <- nrow(d); k <- 3
X <- model.matrix(~ Hits + Years, data = d)
y <- d$Salary
```

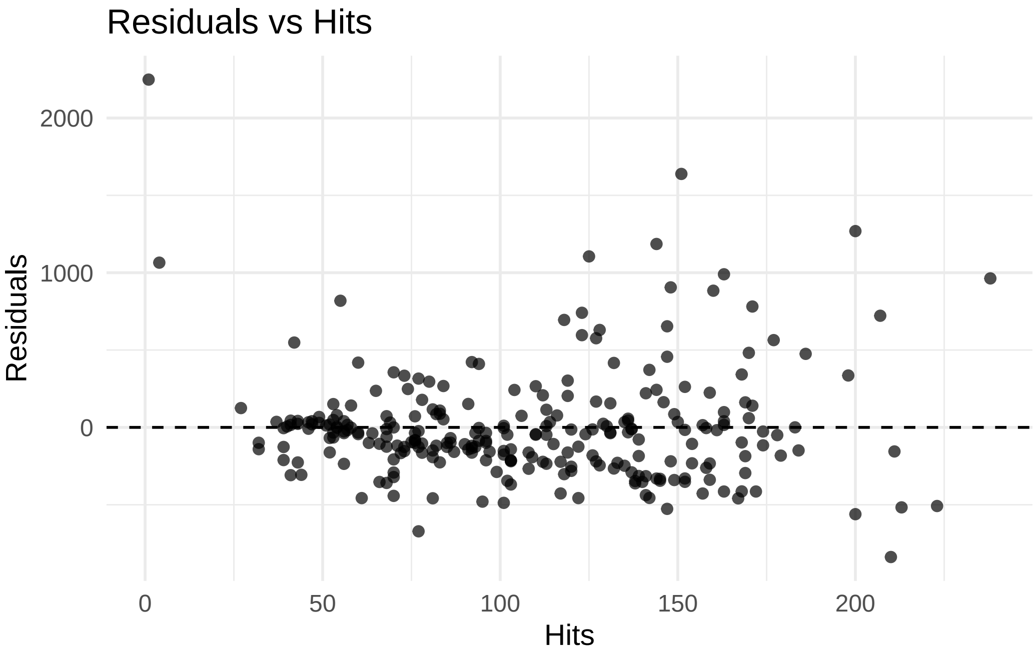
5.1 4.a) OLS by hand

```
XtX   <- crossprod(X)
XtX_i <- solve(XtX)
Xty   <- crossprod(X, y)
b_hat <- XtX_i %*% Xty

tibble::as_tibble(t(b_hat), .name_repair = "minimal") |>
  setNames(colnames(X)) |>
  dplyr::mutate(.rows = "beta_hat (by hand)")
```

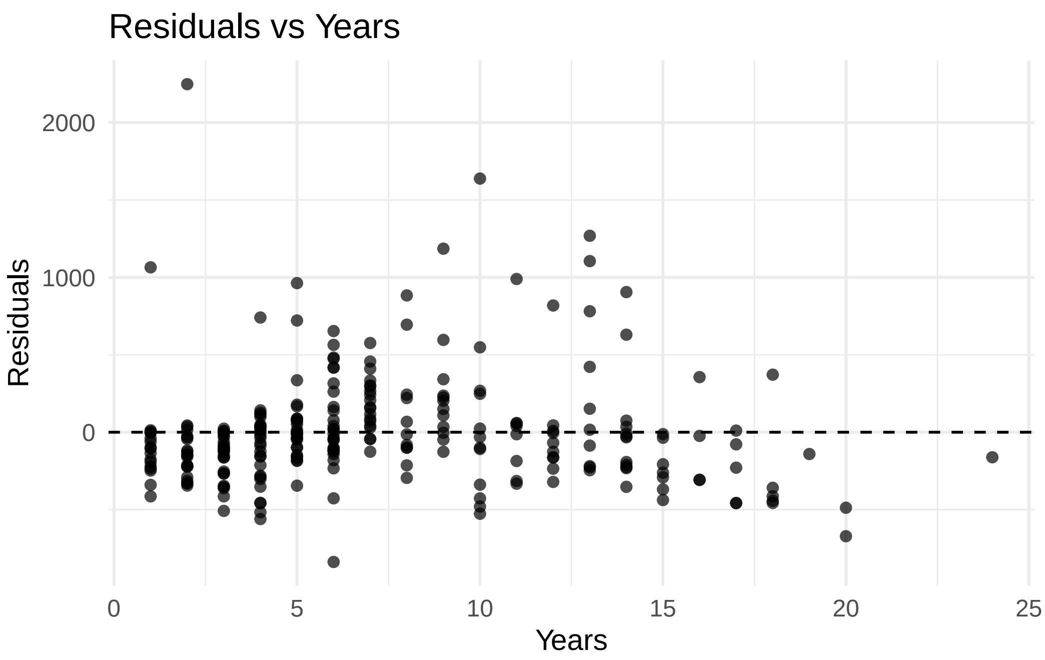
```
# A tibble: 1 x 4
  `(Intercept)`  Hits Years .rows
          <dbl> <dbl> <dbl> <chr>
1         -199.  4.31  37.0 beta_hat (by hand)
```

5.2 4.b) Residual plots

```
y_hat <- as.vector(X %*% b_hat)
e_hat <- y - y_hat

ggplot(d, aes(x = Hits, y = e_hat)) +
  geom_point(alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuals vs Hits", x = "Hits", y = "Residuals")
```



```
ggplot(d, aes(x = Years, y = e_hat)) +
  geom_point(alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuals vs Years", x = "Years", y = "Residuals")
```



5.3 4.c) Homoskedastic SEs

```
SSE <- sum(e_hat^2)
sigma2_hat <- SSE / (n - k)
V_homo <- sigma2_hat * XtX_i
se_homo <- sqrt(diag(V_homo))
tibble::tibble(term = colnames(X), se_homoskedastic = se_homo)
```

```
# A tibble: 3 x 2
  term        se_homoskedastic
  <chr>                  <dbl>
1 (Intercept)           67.5  
2 Hits                   0.501
3 Years                  4.72 
```

5.4 4.d) HC1 SEs

```r
Omega_hat <- diag(as.numeric(e_hat^2))
meat <- t(X) %*% Omega_hat %*% X
V_hc1 <- (n/(n-k)) * XtX_i %*% meat %*% XtX_i
se_hc1 <- sqrt(diag(V_hc1))
tibble::tibble(term = colnames(X), se_HC1 = se_hc1)
```

```
# A tibble: 3 x 2
  term        se_HC1
  <chr>        <dbl>
1 (Intercept) 96.7
2 Hits         0.755
3 Years        4.90
```

5.5 4.e) R^2 and adjusted R^2

```r
y_bar <- mean(y)
SST <- sum( (y - y_bar)^2 )
R2 <- 1 - SSE / SST
adjR2 <- 1 - (SSE/(n - k)) / (SST/(n - 1))
tibble::tibble(R2 = R2, adj_R2 = adjR2)
```

```
# A tibble: 1 x 2
     R2 adj_R2
  <dbl>  <dbl>
1 0.347  0.342
```