HW3

Nathan Krieger

2026-02-06

```
library(ISLR2)
#head(Carseats)
```

Problem 1

(a)

(i)

```
knitr::kable(head(Carseats))
```

Sales	CompPrice	Income	Advertising	Population	Price	ShelveLoc	Age	Education	Urban	US
9.50	138	73	11	276	120	Bad	42	17	Yes	Yes
11.22	111	48	16	260	83	Good	65	10	Yes	Yes
10.06	113	35	10	269	80	Medium	59	12	Yes	Yes
7.40	117	100	4	466	97	Medium	55	14	Yes	Yes
4.15	141	64	3	340	128	Bad	38	13	Yes	No
10.81	124	113	13	501	72	Bad	78	16	No	Yes

```
m1 <- lm(Sales ~ CompPrice + Income +Advertising + Population + Price + Age + Education + Urban + US, da

#summary(m1)$coefficients
knitr::kable(summary(m1)$coefficients)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	7.8243876	1.1298792	6.9249770	0.0000000
CompPrice	0.0942545	0.0078637	11.9859556	0.0000000
Income	0.0130501	0.0034908	3.7383797	0.0002129
Advertising	0.1369399	0.0210394	6.5087503	0.0000000
Population	-0.0002007	0.0007010	-0.2863272	0.7747796
Price	-0.0924395	0.0050621	-18.2610052	0.0000000
Age	-0.0447919	0.0060226	-7.4373656	0.0000000
Education	-0.0423034	0.0373738	-1.1319018	0.2583712
UrbanYes	-0.1559036	0.2133519	-0.7307344	0.4653802
USYes	-0.1062926	0.2832192	-0.3753018	0.7076401

(ii)

Null hypothesis (H0): The CompPrice of the car seat has zero effect on the sale of the car seats

Alternative hypothesis (H1): The CompPrice of the car seat has an effect on the sale of the car seats

Test statistic:

```r
comp_price_t_val <- summary(m1)$coefficients["CompPrice", "t value"]

knitr::kable(comp_price_t_val)
```

x
11.98596

Null Distribution:

```r
n <- nrow(Carseats)

null_dist <- n - 10
cat("Degrees of freedom - ", null_dist)
```

```
## Degrees of freedom -  390
```

The null distribution assumes the null hypothesis is true. We can calculate the degrees of freedom to see what the distribution looks like. 400 - 9 - 1 = 390. This number indicates a distribution that is close to a normal distribution.

```r
comp_price_p_val <- summary(m1)$coefficients["CompPrice", "Pr(>|t|)"]

print(comp_price_p_val)
```

```
## [1] 2.131214e-28
```

Since the p-value is less than $\alpha = 0.05$ (2.131214e-28), we reject the null hypothesis and conclude that CompPrice has a statistically significant relationship with car seat Sales.

(b)

I needed to assume the ϵ (errors) are normally distributed in the t-test for the regression coefficients to be valid.

(c)

```r
knitr::kable(summary(m1)$sigma^2)
```

x
3.733413

$\hat{\sigma}^2 = 3.732624$

This is the estimated variance of the error term. It represents the average squared deviation of the actual sales values from the predicted regression line.

More simple: It is how far away the real value is from the predicted value.

(d)

```
knitr::kable(summary(m1)$coefficients["Advertising", ])
```

	x
Estimate	0.1369399
Std. Error	0.0210394
t value	6.5087503
Pr(>|t|)	0.0000000

Holding all other variables fixed, an increase of $1,000 in advertising is associated with an increase of approximately 0.137 thousand units (about 137 units) in expected car seat sales.

(e)

```
rss_full <- sum(resid(m1)^2)

# Reduced model (intercept only)
m0 <- lm(Sales ~ 1, data = Carseats)
rss_reduced <- sum(resid(m0)^2)

knitr::kable(rss_full)
```

x
1456.031

```
knitr::kable(rss_reduced)
```

x
3182.275

RSS for the full model: 1456.031

RSS for the reduced model: 3182.275

(f)

```r
knitr::kable(anova(m0, m1))
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
399	3182.275	NA	NA	NA	NA
390	1456.031	9	1726.244	51.3752	0

Null Hypothesis (H0): $\beta_1 = \beta_2 = ... = \beta_p = 0$

This means that none of the predictors are useful for predicting car seat sales.

Alternative Hypothesis (H1): At least 1 β_j is non-zero

This means that at least 1 predictor is useful for predicting car seat sales.

Test Statistic:

F = 51.375

Null Distribution:

$F \sim F_{9,390}$

9 is the number of predictors and 390 is the residual degrees of freedom from the full model

P-value:

P-value < 2.2e-16

Conclusion:

Since the p-value is way smaller than 0.05 we reject the null hypothesis. There is significant evidence that at least 1 of the predictors is useful when predicting car seat sales.

(g)

```r
# Create predictor values
new_X <- data.frame(
  CompPrice = mean(Carseats$CompPrice),
  Income = median(Carseats$Income),
  Advertising = 15,
  Population = 500,
  Price = 50,
  Age = 30,
  Education = 10,
  Urban = "Yes",
  US = "Yes"
)

# Confidence interval for f(X)
knitr::kable(predict(m1, newdata = new_X, interval = "confidence", level = 0.95))
```

fit	lwr	upr
15.80707	14.91953	16.69461

The estimate is 15.807. 14.9 is the lower bound and 16.7 is the upper bound.

(h)

```r
# Prediction interval for Y
knitr::kable(predict(m1, newdata = new_X, interval = "prediction", level = 0.95))
```

fit	lwr	upr
15.80707	11.90593	19.70821

The sources that need to be reflected in the interval are reducable and irreducable.

The estimate of 15.807 stays the same but the interval changes. 11.9 is the lower bound and 19.7 is the upper bound.

(i)

Claim: $F(X) = E(\hat{Y})$ for fixed values of X.

Proof: Consider the multiple regression model: $Y = X\beta + \epsilon$

X is the fixed design matrix β is the vector of regression coefficients ϵ is the error vector satisfying

$E(\epsilon) = 0, Var(\epsilon) = \sigma^2 I$

We define regression function

$f(X) = X\beta$

Taking expectations of Y which is conditional on fixed X

$E(Y|X) = E(X\beta + \epsilon|X)$

By linearity of expectation

$E(Y|X) = X\beta + E(\epsilon|X)$

But $E(\epsilon|X) = 0$

So

$E(Y|X) = X\beta = f(X)$

Since X is fixed, $E(Y \mid X) = E(Y)$.

Therefore $f(X) = E(\hat{Y})$

(j)

```r
new_X_high_price <- new_X
new_X_high_price$Price <- 450

knitr::kable(predict(m1, newdata = new_X_high_price))
```

x
-21.16873

So when the car seat sales price is $450 the predicted number of units sold is -21.16873 which is not possible because it is negative. This shows the a limitation of the model because its assuming there is a linear relationship between the price of car seats and the number sold which is not always the case. A lower bound of $0 should be put in place because price cannot go negative.

(k)

No. $\hat{Y}$ is not an unbiased estimator of the actual observed Y because Y includes an error term that $\hat{Y}$ cannot predict.

Assume $E(\hat{Y}) = Y$

Then, $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 * X_1 + \cdots + \hat{\beta}_p * X_p$

But Y needs an error term

$Y = \beta_0 + \beta_1 * X_1 + \cdots + \beta_p * X_p + \epsilon$

So, Y has an error term and $\hat{Y}$ doesn't

Therefore

$E(\hat{Y}) \neq Y$

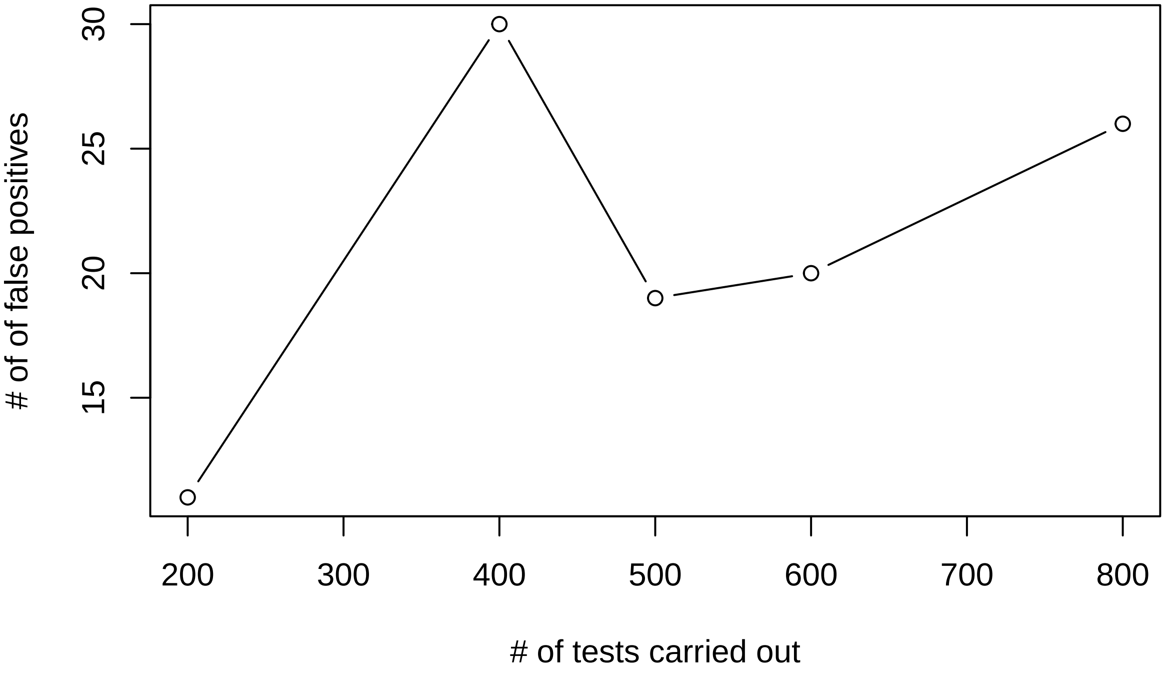
Problem 2

(a)

$m * \alpha$

(b)

```r
set.seed(123)
n = 1000  # Number of observations
p_test = c(200, 400, 500, 600, 800)
alpha = 0.05

false_positives <- c()

for (p in p_test) {
  x <- matrix(rnorm(n * p), n, p)
  y <- rnorm(n)

  data <- as.data.frame(x)

  model <- lm(y ~ ., data = data)

  #Step 6: Extract p-values for all predictors
  p_values <- summary(model)$coefficients[-1,4]  # use -1 to Remove intercept

  #Step 7: Count number of significant predictors at 0.05 level
  significant_count = sum(p_values < alpha)
  # or
  false_positives <- c(false_positives, significant_count)
}

plot(p_test, false_positives, type="b",
     xlab = "# of tests carried out",
     ylab = "# of of false positives"
)
```



The number of false positives changing based on the amount of tests is clearly represented by the graph above. It looks like as the number of tests increase, the number of false positives increase also. However, there is a clear spike when 400 tests are carried out. In general it is safe to say that as the number of tests increase so does the number of false positives.

Problem 3

(a)

$\beta_0 = 2$ $\beta_1 = 3$ $\beta_2 = 5$

(b)

```r
set.seed(123)

X1 = seq(0,10,length.out =100) #generates 100 equally spaced values from 0 to 10.
X2 = runif(100) #generates 100 uniform values.

epsilon = rnorm(100, mean = 0, sd = 2)

Y = 2 + 3*X1 + 5*X2 + epsilon
Y
```

```
##   [1]  3.944525  6.187462  4.565204 10.061382  7.462916  6.775875  3.361204
##   [8]  9.752535  7.429126  7.442230 10.573749  6.595357  8.357802  6.765410
##  [15]  4.613465 11.651637  8.975343  7.467821 10.938684 16.630263 11.526240
##  [22]  7.209316 13.880678 12.522644 11.175239 15.169553 12.029572 10.711093
##  [29] 12.293254 11.245664 15.917559 16.675995 14.409176 17.266090 11.985126
##  [36] 15.658604 18.895067 15.164524 14.454193 17.273926 16.822220 17.593768
##  [43] 17.273358 15.618718 18.816862 15.129875 21.479230 21.637458 17.403917
##  [50] 19.084782 15.959858 20.179313 21.258816 17.975017 19.265139 19.609268
##  [57] 18.037546 19.703383 23.290531 23.589095 22.356700 22.174980 19.471962
##  [64] 22.351703 26.505954 24.541858 26.261674 25.083566 24.878364 23.059992
##  [71] 27.219790 24.766308 26.387979 23.612152 30.488549 24.523967 27.400159
##  [78] 28.553110 25.471640 26.352455 30.349623 30.788741 29.019185 30.247501
##  [85] 23.862374 32.194714 30.064111 34.308787 36.917219 26.957174 31.329774
##  [92] 32.316872 28.452082 30.436274 28.883642 30.664521 32.078869 33.237748
##  [99] 38.231083 31.983466
```

(c)

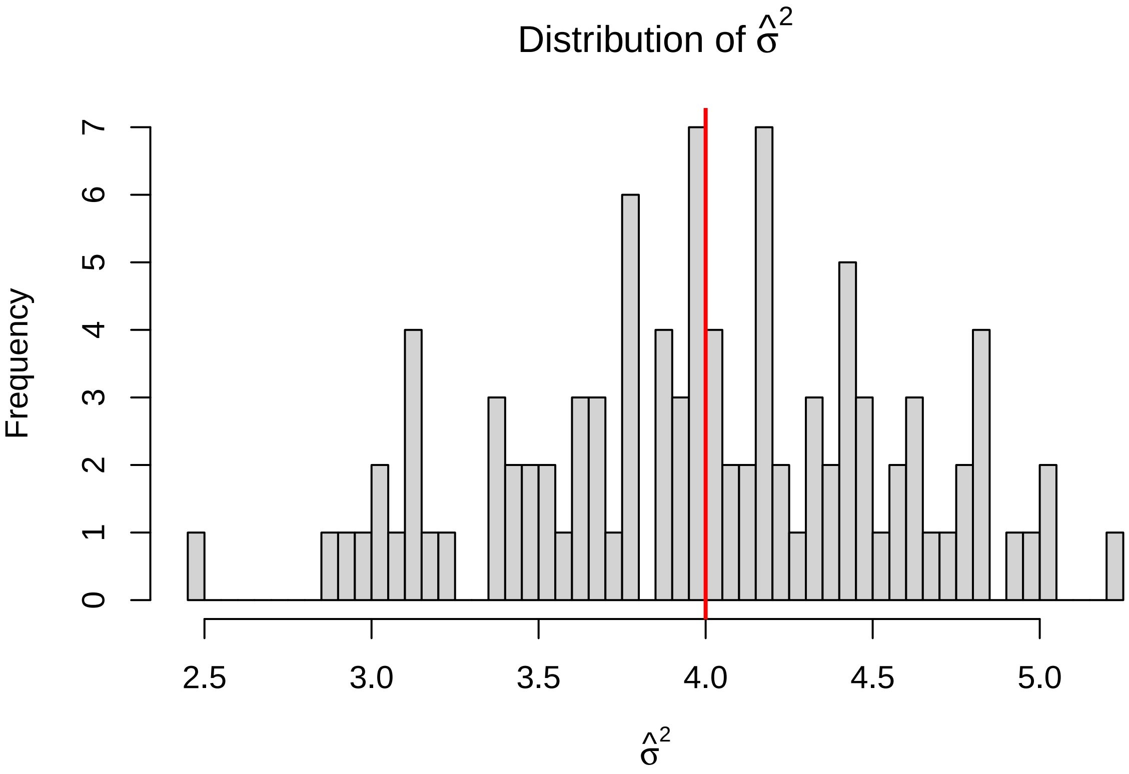
```r
set.seed(123)

B <- 100
sigma2_hat <- numeric(B)

for (b in 1:B) {
  epsilon <- rnorm(100, 0, 2)
  Y <- 2 + 3*X1 + 5*X2 + epsilon

  model <- lm(Y ~ X1 + X2)
  sigma2_hat[b] <- summary(model)$sigma^2
}

knitr::kable(mean(sigma2_hat))
```

x
3.997873

I am getting a mean very close to 4.

(d)

```r
hist(sigma2_hat,
     breaks = 50,
     main = expression("Distribution of " * hat(sigma)^2),
     xlab = expression(hat(sigma)^2))

abline(v = 4, col = "red", lwd = 2)
```



(e)

An accurate estimate of σ^2 is important in multiple linear regression because it directly affects the standard errors of the regression coefficients. The standard errors are used to make confidence intervals, do hypothesis tests, and make prediction intervals.

If $\hat{\sigma}^2$ is not well estimated, then standard errors will be incorrect, which leads to improper interpretation. The entire statistical inference pipeline breaks down.

Problem 4

(a)

Setting the significance level to $a = 0.05$ means that before we look at the data, we are deciding that we're willing to accept a 5% chance of making a false alarm.

(b)

I do not agree that a p-value of 0.0647 implies the predictor is not meaningful. It just means the evidence is not quite strong enough to reject the null at the chosen significance level. More thought should go into this decision than simply the p-value.

(c)

This is a bad idea because the chances of a false positive are high. Each t-test has a 10% chance of a false positive so running 12 of these tests will result in a much higher probability than 10%:

$1 - 0.9^{12} = 0.718$

So, there is a 71.8% chance of at least 1 significant result just by chance.