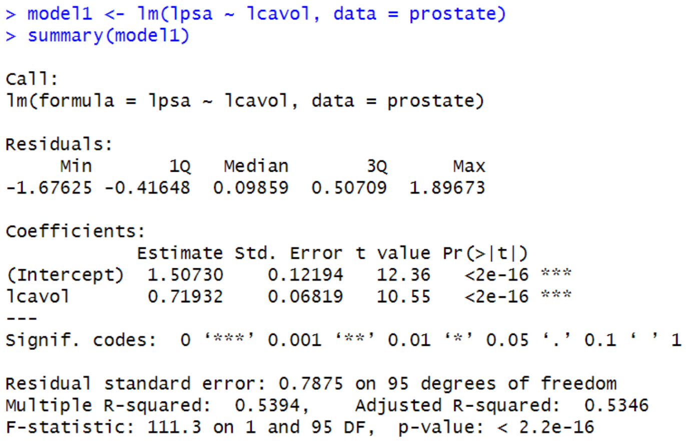
STAT514 HW1 Solution

1 Q.1

1.1 a.

```
library(faraway)
data(prostate)
model1 <- lm(lpsa ˜ lcavol, data = prostate)
```

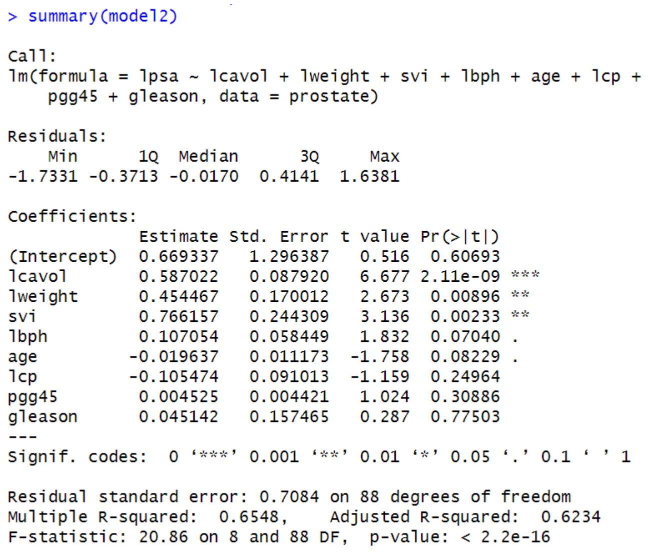


```
> model1 <- lm(lpsa ~ lcavol, data = prostate)
> summary(model1)

Call:
lm(formula = lpsa ~ lcavol, data = prostate)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.67625 -0.41648  0.09859  0.50709  1.89673 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.50730    0.12194   12.36   <2e-16 ***
lcavol       0.71932    0.06819   10.55   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7875 on 95 degrees of freedom
Multiple R-squared:  0.5394,	Adjusted R-squared:  0.5346 
F-statistic: 111.3 on 1 and 95 DF,  p-value: < 2.2e-16
```

R-squared: 0.5394, Adjusted R-squared: 0.5346

1.2 b.

```
model2 <- lm(lpsa ˜ lcavol + lweight + svi + lbph + age + lcp + pgg45 + gleason,
             data = prostate)
```



```
> summary(model2)

Call:
lm(formula = lpsa ~ lcavol + lweight + svi + lbph + age + lcp +
    pgg45 + gleason, data = prostate)

Residuals:
    Min      1Q  Median      3Q     Max
-1.7331 -0.3713 -0.0170  0.4141  1.6381

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.669337   1.296387   0.516  0.60693
lcavol       0.587022   0.087920   6.677 2.11e-09 ***
lweight      0.454467   0.170012   2.673  0.00896 **
svi          0.766157   0.244309   3.136  0.00233 **
lbph         0.107054   0.058449   1.832  0.07040 .
age         -0.019637   0.011173  -1.758  0.08229 .
lcp         -0.105474   0.091013  -1.159  0.24964
pgg45        0.004525   0.004421   1.024  0.30886
gleason      0.045142   0.157465   0.287  0.77503
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7084 on 88 degrees of freedom
Multiple R-squared:  0.6548,    Adjusted R-squared:  0.6234
F-statistic: 20.86 on 8 and 88 DF,  p-value: < 2.2e-16
```

R-squared: 0.6548, Adjusted R-squared: 0.6234

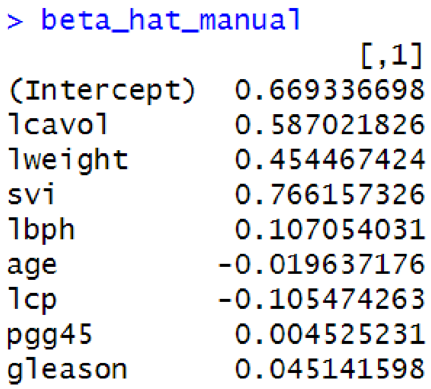
1.3 c.

The R-squared of the linear model with multiple predictors is larger than the one with R-squared with only one predictor. This means that the multiple predictors model's predictors account for a large portion of the variation in the dependent variable, suggesting a strong fit for the data.

1.4 d.

- Manual Computation in R

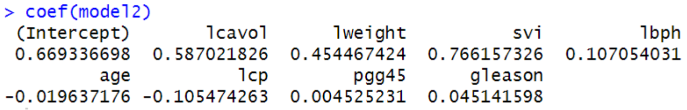
```
y <- prostate$lpsa

X <- model.matrix(~ lcavol + lweight + svi + lbph + age + lcp + pgg45 + gleason,
                  data = prostate)

beta_hat_manual <- solve(t(X) %*% X) %*% t(X) %*% y
```



```
> beta_hat_manual
                    [,1]
(Intercept)  0.669336698
lcavol       0.587021826
lweight      0.454467424
svi          0.766157326
lbph         0.107054031
age         -0.019637176
lcp         -0.105474263
pgg45        0.004525231
gleason      0.045141598
```

- Compare with lm()

```
model2 <- lm(lpsa ˜ lcavol + lweight + svi + lbph + age + lcp + pgg45 + gleason,
             data = prostate)

coef(model2)
```



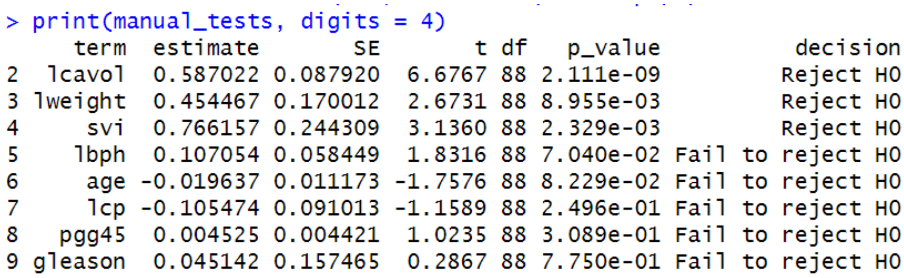
```
> coef(model2)
 (Intercept)       lcavol      lweight          svi         lbph 
 0.669336698  0.587021826  0.454467424  0.766157326  0.107054031 
         age          lcp        pgg45      gleason 
-0.019637176 -0.105474263  0.004525231  0.045141598 
```

The manually estimated parameters using the matrix formulation of least squares are basically identical to those obtained from the lm() function.

1.5 e.

```
y <- prostate$lpsa

X <- model.matrix(˜ lcavol + lweight + svi + lbph + age + lcp + pgg45 + gleason,
                  data = prostate)

n <- nrow(X); p <- ncol(X)

XtX_inv <- solve(t(X) %*% X)
beta_hat <- XtX_inv %*% (t(X) %*% y)

yhat <- X %*% beta_hat
res  <- y - yhat

RSS <- as.numeric(crossprod(res))
sigma2_hat <- RSS / (n - p)
```

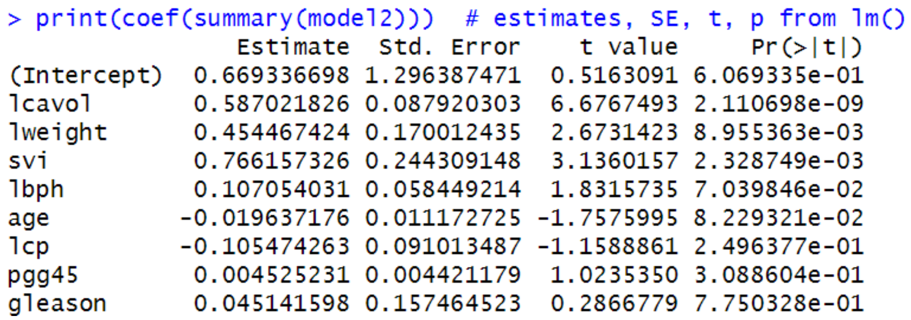
```
var_beta <- sigma2_hat * XtX_inv
se_beta  <- sqrt(diag(var_beta))
t_vals   <- as.vector(beta_hat) / se_beta
df       <- n - p
p_vals   <- 2 * pt(abs(t_vals), df = df, lower.tail = FALSE)

alpha <- 0.05
out <- data.frame(
  term     = colnames(X),
  estimate = as.vector(beta_hat),
  SE       = se_beta,
  t        = t_vals,
  df       = df,
  p_value  = p_vals,
  decision = ifelse(p_vals < alpha, "Reject H0", "Fail to reject H0"),
  row.names = NULL
)

manual_tests <- subset(out, term != "(Intercept)")
print(manual_tests, digits = 4)
```



```
> print(manual_tests, digits = 4)
     term  estimate       SE       t df   p_value          decision
2  lcavol  0.587022 0.087920  6.6767 88 2.111e-09         Reject H0
3 lweight  0.454467 0.170012  2.6731 88 8.955e-03         Reject H0
4     svi  0.766157 0.244309  3.1360 88 2.329e-03         Reject H0
5    lbph  0.107054 0.058449  1.8316 88 7.040e-02 Fail to reject H0
6     age -0.019637 0.011173 -1.7576 88 8.229e-02 Fail to reject H0
7     lcp -0.105474 0.091013 -1.1589 88 2.496e-01 Fail to reject H0
8   pgg45  0.004525 0.004421  1.0235 88 3.089e-01 Fail to reject H0
9 gleason  0.045142 0.157465  0.2867 88 7.750e-01 Fail to reject H0
```

1.6 f.

```
model2 <- lm(lpsa ~ lcavol + lweight + svi + lbph + age + lcp + pgg45 + gleason,
             data = prostate)

print(coef(summary(model2)))
```

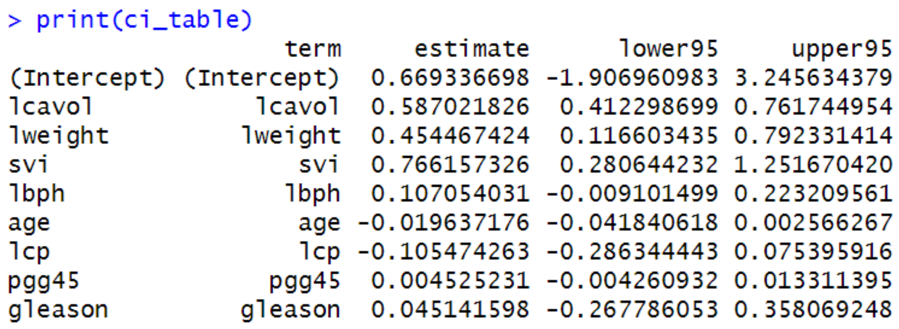


```
> print(coef(summary(model2)))  # estimates, SE, t, p from lm()
                Estimate  Std. Error    t value     Pr(>|t|)
(Intercept)  0.669336698 1.296387471  0.5163091 6.069335e-01
lcavol       0.587021826 0.087920303  6.6767493 2.110698e-09
lweight      0.454467424 0.170012435  2.6731423 8.955363e-03
svi          0.766157326 0.244309148  3.1360157 2.328749e-03
lbph         0.107054031 0.058449214  1.8315735 7.039846e-02
age         -0.019637176 0.011172725 -1.7575995 8.229321e-02
lcp         -0.105474263 0.091013487 -1.1588861 2.496377e-01
pgg45        0.004525231 0.004421179  1.0235350 3.088604e-01
gleason      0.045141598 0.157464523  0.2866779 7.750328e-01
```

Part E and F shows the same result.

1.7 g.

```
y <- prostate$lpsa
X <- model.matrix(˜ lcavol + lweight + svi + lbph + age + lcp + pgg45 + gleason,
                  data = prostate)

n <- nrow(X); p <- ncol(X)

XtX_inv <- solve(t(X) %*% X)
beta_hat <- XtX_inv %*% (t(X) %*% y)

res <- y - X %*% beta_hat
RSS <- as.numeric(crossprod(res))
sigma2_hat <- RSS / (n - p)

var_beta <- sigma2_hat * XtX_inv
se_beta <- sqrt(diag(var_beta))

t_crit <- qt(0.975, df = n - p)

lower <- as.vector(beta_hat) - t_crit * se_beta
upper <- as.vector(beta_hat) + t_crit * se_beta

ci_table <- data.frame(
  term = colnames(X),
  estimate = as.vector(beta_hat),
  lower95 = lower,
  upper95 = upper
)
```



```
> print(ci_table)
                   term     estimate      lower95     upper95
(Intercept) (Intercept)  0.669336698 -1.906960983 3.245634379
lcavol           lcavol  0.587021826  0.412298699 0.761744954
lweight         lweight  0.454467424  0.116603435 0.792331414
svi                 svi  0.766157326  0.280644232 1.251670420
lbph               lbph  0.107054031 -0.009101499 0.223209561
age                 age -0.019637176 -0.041840618 0.002566267
lcp                 lcp -0.105474263 -0.286344443 0.075395916
pgg45             pgg45  0.004525231 -0.004260932 0.013311395
gleason         gleason  0.045141598 -0.267786053 0.358069248
```

2 Q.2

2.1 a.

Minimize the residual sum of squares

$$S(\beta) = \|y - X\beta\|_2^2 = (y - X\beta)^\top (y - X\beta)$$

Differentiate and set the gradient to zero:

$$\nabla_\beta S(\beta) = -2X^\top (y - X\beta) = 0 \quad \Rightarrow \quad X^\top X\,\hat{\beta} = X^\top y$$

Since X has full column rank, $X^\top X$ is invertible, so the unique minimizer is

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

(uniqueness follows because S is strictly convex: $X^\top X \succ 0$)

2.2 b.

Using $y = X\beta + \varepsilon$ and $E[\varepsilon] = 0$:

$$E[\hat{\beta}] = (X^\top X)^{-1} X^\top E[y] = (X^\top X)^{-1} X^\top (X\beta) = \beta$$

So $E[\hat{\beta}] = \beta$

2.3 c.

$$\begin{aligned} \mathrm{Var}(\hat{\beta}) &= \mathrm{Var}\big((X^\top X)^{-1} X^\top y\big) \\ &= (X^\top X)^{-1} X^\top \,\mathrm{Var}(y)\, X (X^\top X)^{-1} \\ &= (X^\top X)^{-1} X^\top (\sigma^2 I_n)\, X (X^\top X)^{-1} \\ &= \sigma^2 (X^\top X)^{-1} \end{aligned}$$

Thus $\mathrm{Var}(\hat{\beta}) = \sigma^2 (X^\top X)^{-1}$

2.4 d.

Write $\hat{\beta} = Ay$ with $A = (X^\top X)^{-1}X^\top$. Because $y \sim \mathcal{N}(X\beta, \sigma^2 I_n)$ and linear transforms of multivariate normals are normal,

$$\hat{\beta} \sim \mathcal{N}\big(A(X\beta),\ A(\sigma^2 I_n)A^\top\big) = \mathcal{N}\big(\beta,\ \sigma^2(X^\top X)^{-1}\big)$$

So

$$\hat{\beta} \sim \mathcal{N}\big(\beta,\ \sigma^2(X^\top X)^{-1}\big)$$

3 Q.3

3.1 a.

Define the $(p+1) \times (p+1)$ matrix

$$C = \begin{bmatrix} 1 & c_{10} & c_{20} & \cdots & c_{p0} \\ 0 & c_{11} & c_{21} & \cdots & c_{p1} \\ 0 & c_{12} & c_{22} & \cdots & c_{p2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & c_{1p} & c_{2p} & \cdots & c_{pp} \end{bmatrix}$$

Its first column is $(1, 0, \ldots, 0)^\top$ so the intercept is unchanged; for $j = 1, \ldots, p$ the j+1-st column of C is

$$\begin{bmatrix} c_{j0} \\ c_{j1} \\ \vdots \\ c_{jp} \end{bmatrix}$$

which encodes $z_j = c_{j0} \cdot \mathbf{1} + \sum_{k=1}^{p} c_{jk}x_k$. Then column-by-column, $Z = XC$ because

$$Z_{\cdot,1} = X_{\cdot,1}, \qquad Z_{\cdot,j+1} = X\,C_{\cdot,j+1} = c_{j0}\mathbf{1} + \sum_{k=1}^{p} c_{jk}x_k = z_j$$

3.2 b.

OLS predictions are $\hat{y}_X = H_X y$ and $\hat{y}_Z = H_Z y$ with

$$H_X = X(X^\top X)^{-1}X^\top, \qquad H_Z = Z(Z^\top Z)^{-1}Z^\top$$

Using $Z = XC$ and invertible C,

$$
\begin{aligned}
H_Z &= XC\,(C^\top X^\top XC)^{-1}\,C^\top X^\top \\
&= XC\left(C^{-1}(X^\top X)^{-1}C^{-\top}\right)C^\top X^\top \\
&= X(X^\top X)^{-1}X^\top \\
&= H_X
\end{aligned}
$$

Hence $\hat{y}_Z = \hat{y}_X$. Equivalently, since C is invertible, $\mathcal{C}(Z) = \mathcal{C}(X)$; both models project y onto the same column space, so fitted values (and thus residuals, RSS, R^2) are identical.