Homework 1:

Q1(a):

1.

(a) $x_1^T = [1, 0]$ $\quad x_2^T = [1, 0]$ $\quad x_3^T = [0, 1]$ $\quad x_4^T = [0, 1]$

$$y^T = [1, 2, 3, 4].$$

$$X = \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}}_{4\times2 = n\times p} \qquad y = \underbrace{\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}}_{4\times1 = n\times p} \qquad X^T = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}}_{2\times4 = n\times p}$$

The data matrix X is as given above in the figure because the X^T_1 denotes data for experimental unit 1, the X^T_2 denotes data for experimental unit 2 and so on.

Q1 (b):

(b)

$$X^T X = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}}_{2\times2} = A \text{ (say).}$$

Computation example: (1*1) + (1*1) + (0*1) + (0 *1) = 2 for A_{11} of matrix A. Rest of the elements of matrix A can be computed similarly.

Q1(c):

(c)

$$[X^T X]^{-1} = A^{-1} = \frac{1}{\det(A)} \text{Cofactor}[A] = \frac{1}{|A|} \text{adj}(A)$$

$$= \frac{1}{(2\times2 - 0\times0)} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}}_{2\times2}$$

For cofactor matrix, flip numbers in the diagonal from left-top to right-bottom and flip signs in the diagonal from left-bottom to right-top.

Q1(d):

(d) say

$$B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} ; \; B^{-1} = \frac{1}{|x|-|x|} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \frac{1}{0} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = 0 \text{ Nope.}$$

B is a singular matrix ($\det(B) = 0$). So, inverse computation is not possible.

Matrix is singular if $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ & $\det(M) \Rightarrow |M| \Rightarrow ad - bc = 0$

& Non singular if $|M| \neq 0$.

Q1 (e):

(e) $$[X^T X]^{-1} [X^T X] = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

$AA^{-1} = I$. (Identity matrix)

Q1 (f):

(f)

$$[X^T X]^{-1} [X^T Y] = ?$$

$$X^T y = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \end{bmatrix}$$

$$[X^T X]^{-1} [X^T Y] = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 3 \\ 7 \end{bmatrix} = \begin{bmatrix} 3/2 \\ 7/2 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 3.5 \end{bmatrix}$$

X1 = 1.5 and x2 = 3.5.

Q1. (g):

R Codes:
x = matrix (c(1, 1, 0, 0, 0, 0, 1, 1), nrow = 4, ncol = 2, byrow = F) # matrix x
dim(x) # dimension (n, p) of x
y = matrix (c(1, 2, 3, 4), nrow = 4, ncol = 1, byrow = F) #matrix y
dim(y) # dimension (n, p) of y
xt = t(x) # transverse of matrix x.
yt = t(y) # transverse of y
xxtdet = det(t(x)%*%x) # determinant of x and x transverse.
xxtinv = solve(t(x)%*%x) #Multiply x & t(x) and inverse resulting matrix.
xty = t(x)%*%y # multiply t(x) and y.

betas = xxtinv%*%xty #beta coefficients gives values of x1 and x2.
betas1 = solve(t(x)%*%x)%*%(t(x))%*%y #compute everything done above in one line.
Identical (betas, betas1) #check if both ways give same results or not. The answer must be "TRUE".
betas #check values of coefficients. These are slopes of each variable.

Answers:

```
> betas1 = solve(t(x)%*%x)%*%(t(x))%*%y #compute everything done above in one lin
e.
> betas1
     [,1]
[1,]  1.5
[2,]  3.5
```

X1 = 1.5 and x2 = 3.5.

Q1. (h)

R Codes:

x1 = x[,1] #Extract variable x1 from matrix x. This is p = 1.
x2 = x[,2] #Extract variable x2 from matrix x. This is p = 2.

model = lm (y ~ x1 + x2)
summary(model)
model1 = lm (y ~ x1 + x2 - 1)
summary(model1)

Results:

```
> model = lm (y ~ x1 + x2)
> summary(model)

Call:
lm(formula = y ~ x1 + x2)

Residuals:
   1    2    3    4 
-0.5  0.5 -0.5  0.5 

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)   3.5000     0.5000   7.000   0.0198 *
x1           -2.0000     0.7071  -2.828   0.1056  
x2                NA         NA      NA       NA  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7071 on 2 degrees of freedom
Multiple R-squared:    0.8,	Adjusted R-squared:    0.7 
F-statistic:     8 on 1 and 2 DF,  p-value: 0.1056
```

```
> model1 = lm (y ~ x1 + x2 - 1)
> summary(model1)

Call:
lm(formula = y ~ x1 + x2 - 1)

Residuals:
   1    2    3    4 
-0.5  0.5 -0.5  0.5 

Coefficients:
   Estimate Std. Error t value Pr(>|t|)  
x1      1.5        0.5       3   0.0955 .
x2      3.5        0.5       7   0.0198 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7071 on 2 degrees of freedom
Multiple R-squared:  0.9667,	Adjusted R-squared:  0.9333 
F-statistic:    29 on 2 and 2 DF,  p-value: 0.03333
```

Models	X1	X2	Remarks
lm (y ~ x1 + x2)	3.5	NA	y- intercept = -2.0
lm (y ~ x1 + x2 - 1)	1.5	3.5	

The intercept in the first model suppressed x2 and thus we did not see x2 in the model. But in second model, the intercept was suppressed so we see both x1 and x2 with values same as manual calculation.

Q1 (i):

(i) $$y_{ij} = \mu_i + \varepsilon_{ij}$$

$$\hat{\mu}_i = (X^T X)^{-1}(X^T Y)$$

Q2 (a & b).

R Codes:

```
# install.packages("ISLR")
library(ISLR)
try(data(package = "ISLR"))
data("Hitters", "Auto")
Hitters
Auto
dim(Hitters)
dim(Auto)
```

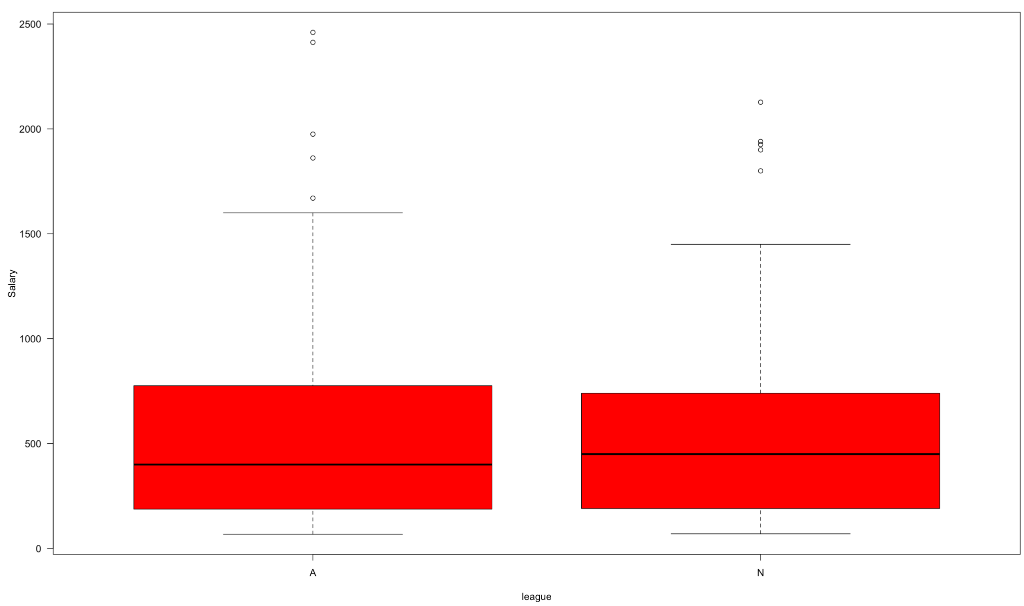
```
Data sets in package 'ISLR':

Auto                 Auto Data Set
Caravan              The Insurance Company (TIC)
                     Benchmark
Carseats             Sales of Child Car Seats
College              U.S. News and World Report's
                     College Data
Credit               Credit Card Balance Data
Default              Credit Card Default Data
Hitters              Baseball Data
Khan                 Khan Gene Data
NCI60                NCI 60 Data
OJ                   Orange Juice Data
Portfolio            Portfolio Data
Smarket              S&P Stock Market Data
Wage                 Mid-Atlantic Wage Data
Weekly               Weekly S&P Stock Market Data
```

```
> dim(Hitters)
[1] 322  20
> dim(Auto)
[1] 392   9
```

2(a): The Hitters data is Baseball data. N and p of Hitters data are 392 and 9 respectively.

2(b): The Auto data Auto data set. N and p of Auto data are 392 and 9 respectively.

2(c):
R Code:
boxplot(Salary ~ League, data = Hitters, xlab = "league", ylab = "Salary", col = "red")



The bold line in the middle of red box is median which is mid-point value, lower boundary of red box is 25th percentile and upper boundary of red box is 75th percentile value. The vertical lines above and below each box are maximum and minimum values of respective variables which gives spread of each variable. The minimum, median and maximum for league A is approximately 200, 400 and 800 and those for league N are 200, little more than 400 and little less than 800 respectively. Values upper and lower horizontal lines above and below each box are outliers.

R Code Compilation:

```
#Homework 1
x = matrix(c(1, 1, 0, 0, 0, 0, 1, 1), nrow = 4, ncol = 2, byrow = F) # matrix x
dim(x) #dimension (n, p) of x
x
y = matrix(c(1, 2, 3, 4), nrow = 4, ncol = 1, byrow = F) #matrix y
dim(y) #dimension (n, p) of y
y
xt = t(x) #trasverse of matrix x.
dim(xt) #dimension of transverse of matrix x.
xt
yt = t(y) #transverse of y
dim(yt) #dimension of tranverse of matrix y.
yt
xxtdet = det(t(x)%*%x) # determinant of x and x transverse.
xxtdet
xxtinv = solve(t(x)%*%x) #Multiply x & t(x) and inverse resulting matrix.
xxtinv
xty = t(x)%*%y # multiply t(x) and y.
xty
betas = xxtinv%*%xty #beta coefficients gives values of x1 and x2.
betas
betas1 = solve(t(x)%*%x)%*%(t(x))%*%y #compute everything done above in one
line.
betas1
identical(betas, betas1) #check if both ways give same results or not. The answer
must be "TRUE".
betas #check values of coefficients. These are slopes of each variables.

x1 = x[,1] #Extract variable x1 from matrix x. This is p = 1.
x2 = x[,2] #Extract variable x2 from matrix x. This is p = 2.

model = lm (y ~ x1 + x2)
summary(model)
model1 = lm (y ~ x1 + x2 - 1)
summary(model1)
# rm(list = ls())

# 2 (a & b):
# install.packages("ISLR")
library(ISLR)
try(data(package = "ISLR"))
data("Hitters", "Auto")
```

```
Hitters
Auto
dim(Hitters)
dim(Auto)

# 2(c)
head(Hitters)
head(Auto)
boxplot(Salary ~ League, data = Hitters, xlab = "league", ylab = "Salary", col = "red",
add = TRUE)
```