Problem 1

a) Missing values and # of observations

Note that $(X^T X)^T = X^T X$, it's obvious that

$$X^T X = \begin{bmatrix} 30 & 0 & 0 \\ 0 & 10 & 7 \\ 0 & 7 & 15 \end{bmatrix} \tag{1}$$

Consider the partition of matrices,

$$X^T X = \begin{bmatrix} \mathbf{1}^T \\ x^T \\ z^T \end{bmatrix} \begin{bmatrix} \mathbf{1} & x & z \end{bmatrix} \tag{2}$$

The first entry of $X^T X$ is calculated by

$$30 = \sum_{i=1}^{n} 1 \tag{3}$$

which gives us $n = 30$

b) Correlation between X and Z

$$corr(X, Z) = \frac{cov(X, Z)}{\sqrt{var(X)var(Z)}} \tag{4}$$

Note that we already have mean-centered data, indicated by the 0-values in the first row

$$X^T X(1, 2) = \mathbf{1}^T x = n * Mean(x) = 0 \tag{5}$$

So the population variance/covariance are directly calculated by $\frac{1}{n} X^T X$

$$corr(X, Z) = \frac{7}{\sqrt{10 * 15}} = 0.57 \tag{6}$$

c) Mean of Y

Any OLS result will pass through the center of the data $(\bar{X}, \bar{Y})$, proved by the first row of the matrix-form normal equation.

$$X^T X\beta = X^T Y \tag{7}$$

Also, as is proved in section (b), $E(X) = E(Z) = 0$.
We can easily conclude that $\bar{Y} = -2$

d) The value of R^2

$$R^2 = \frac{ESS}{TSS} = \frac{ESS}{ESS + RSS} \tag{8}$$

where $ESS = var(\hat{Y})$, which is calculated by

$$var(\hat{Y}) = var(X) + 4var(Z) + 4cov(X, Z) = 10 + 60 + 28 = 98 \tag{9}$$

$$R^2 = 0.89 \tag{10}$$

Problem 2

For the MLE estimation, we shall first define the joint conditional probability distribution (Likelihood Function) of the residuals

$$\begin{aligned} Likelihood &= \prod_{i=1}^{n} P(\epsilon_i | x_i, y_i, w, \sigma^2) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} exp(\frac{-(y_i - w^T x_i)^2}{2\sigma^2}) \\ LogLikelihood \quad L_n &= \sum_{i=1}^{n} \{const - \ln(\sigma) - \frac{(y_i - w^T x_i)^2}{2\sigma^2}\} \end{aligned} \tag{11}$$

Take the derivative $w.r.t. \, \sigma$

$$\begin{aligned} n\frac{1}{\sigma} &= \sum_{i=1}^{n} \frac{(y_i - w^T x_i)^2}{\sigma^3} \\ \sigma^2 &= \sum_{i=1}^{n} \frac{(y_i - w^T x_i)^2}{n} \\ &= \frac{1}{n} u^T u \end{aligned} \tag{12}$$

Problem 3

Due to the uncorrelated property of X_1, X_2, X_3

$$var(X_4) = var(X_1 + X_2 + X_3) = var(X_1) + var(X_2) + var(X_3) = 3 \tag{13}$$

The covariance of X_1 and X_4 are also easy to calculate

$$cov(X_1, X_4) = cov(X_1, X_1 + X_2 + X_3) = var(X_1) = 1 \tag{14}$$

So $r_{14} = \frac{cov(X_1, X_4)}{\sqrt{var(X_1, X_4)}} = 0.577$ The same reason can be applied to r_{24} and r_{34}

Problem 4

For a matrix A to be semi-definite, A must be symmetric.
Take the first-order and second-order derivative of $f(.)$ w.r.t x

$$\begin{aligned} f^{'}(x) &= A^T x - b = 0 \\ f^{''}(x) &= A \geq 0 \end{aligned} \tag{15}$$

Given A is invertible and symmetric, the minimum value of $f(.)$ is realized iff

$$\begin{aligned} A^T x^* &= b \\ x^* &= (A^T)^{-1} b = A^{-1} b \end{aligned} \tag{16}$$

Problem 5

Implement Gradient Descent to minimize both $f_1(x)$ and $f_2(x)$

```
A1 <- diag(c(1,2,2), nrow = 3)
A2 <- diag(c(1,2,0), nrow = 3)
b <- matrix(c(1,1,0), ncol = 1)
epsilon <- c(1e-8,1e-8,1e-8)
lambda <- 0.1
converge <- function(X, lastX){
  for (i in 1:length(X)){
    if (abs(X[i]-lastX[i])>epsilon){
      return(FALSE)
    }
  }
  return(TRUE)
}
for (i in 1:5){
  X = rnorm(3)
  lastX <- X+1
  while (converge(X, lastX) == FALSE){
    lastX = X
    X = X - lambda*(A1 %*% X - b)
  }
  print(X)
}
```

```
##                [,1]
## [1,]  9.999999e-01
## [2,]  5.000000e-01
## [3,] -3.013299e-16
##                [,1]
## [1,]  9.999999e-01
## [2,]  5.000000e-01
## [3,] -3.698993e-15
##                [,1]
## [1,]  9.999999e-01
## [2,]  5.000000e-01
## [3,] -1.410955e-15
##                [,1]
## [1,]  1.000000e+00
## [2,]  5.000000e-01
## [3,] -5.005869e-14
##                [,1]
## [1,]  9.999999e-01
## [2,]  5.000000e-01
## [3,] -2.037503e-17
```

```r
# Compare the results to $x* = A1^{-1}b$
solve(A1) %*%b
```

```
##      [,1]
## [1,]  1.0
## [2,]  0.5
## [3,]  0.0
```

```r
# They converged to the same x*
```

But that is not the case when A is not invertible. Mathematically, the optimization problem has infinite set of solutions. The converged result will thus depends on the initialization process.

```r
for (i in 1:5){
  X = rnorm(3)
  lastX <- X+1
  while (converge(X, lastX) == FALSE){
    lastX = X
    X = X - lambda*(A2 %*% X - b)
  }
  print(X)
}
```

```
##           [,1]
## [1,] 0.9999999
## [2,] 0.5000000
## [3,] 1.1574709
##            [,1]
## [1,]  0.9999999
## [2,]  0.5000000
## [3,] -0.8228044
##            [,1]
## [1,]  0.9999999
## [2,]  0.5000000
## [3,] -0.3215081
##            [,1]
## [1,]  0.9999999
## [2,]  0.5000000
## [3,] -1.2682850
##           [,1]
## [1,] 0.9999999
## [2,] 0.5000000
## [3,] 1.5508908
```