Homework 1

Name : Shashi Roshan. NetID : sroshan2

Answer 1 (a)

```
cship = read.table("cship.dat")
passengers = cship$passengers
crew = cship$crew
summary(passengers)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.66   12.54   19.50   18.46   24.84   54.00
```

```
summary(crew)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.590   5.480   8.150   7.794   9.990  21.000
```

The minimum number of passengers is 0.66, whereas the maximum number of passengers is 54.00. The mean value of passengers is 18.46, which is found by taking the average of all the observations. The minimum number of crew is 0.59, whereas the maximum number of crew is 21.00. The mean value of crew is around 7.79. The median number of crew is around 8.15, which is the middle value when all the observations are sorted in ascending order.

Min : It is the minimum value of all the observations.

1st Qu : The first quartile (Q1) is defined as the middle number between the smallest number and the median of the data set.

Median : The median is the value separating the higher half from the lower half of a data sample.

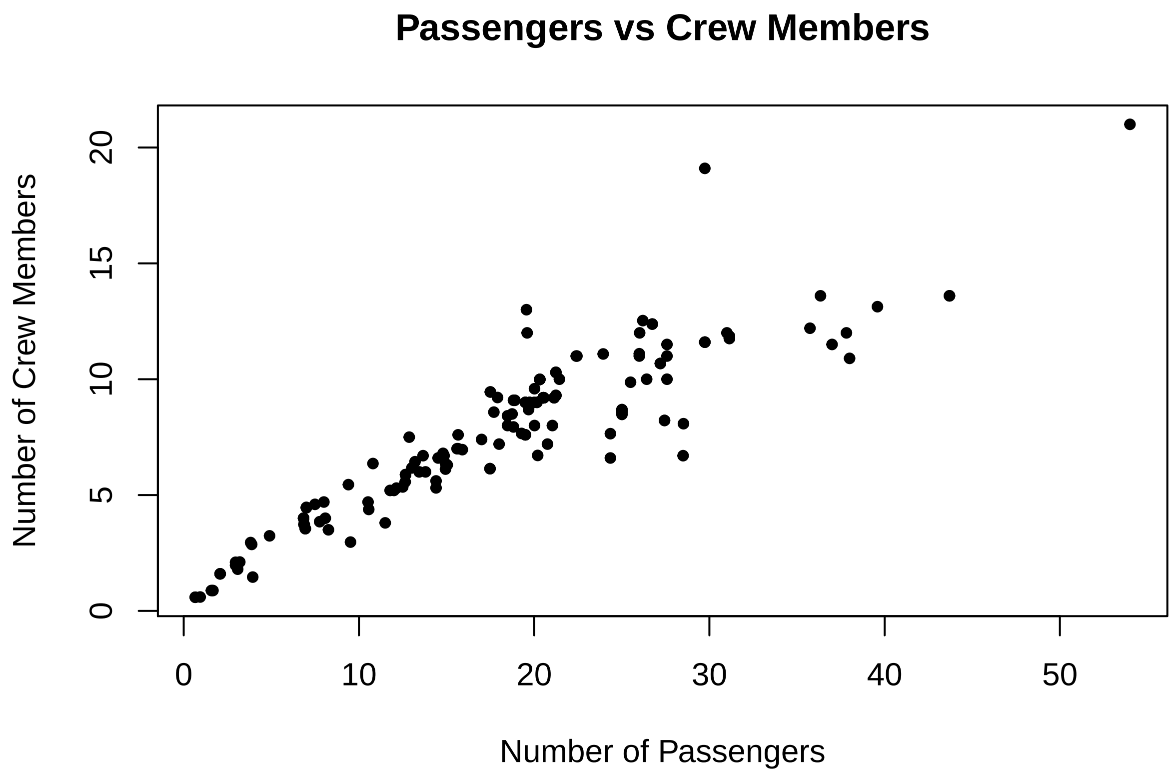
Mean: It is the mean of all the observations.

3rd Qu : The third quartile (Q3) is the middle value between the median and the highest value of the data set.

Max : It is the maximum value among all the observations.

Answer 1 (b)

```
plot(
  passengers,
  crew,
  xlab = "Number of Passengers",
  ylab = "Number of Crew Members",
  main = "Passengers vs Crew Members",
  pch = 20
)
```



The plot between number of passengers and number of crew shows that the number of crew members increase with the number of passengers. They are positively coorelated.

Answer 1 (c)

```
cor_passengers_ship = cor(passengers, crew)
```

Correlations between the number of passengers and crew members is : 0.9152341

Answer 1 (d)

```
model1 = lm(crew ~ passengers, data = cship)
summary(model1)
```

```
##
## Call:
## lm(formula = crew ~ passengers, data = cship)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4218 -0.6446  0.0068  0.7224  7.5673
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.67831    0.24323    6.90 1.23e-10 ***
## passengers   0.33135    0.01168   28.37  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.416 on 156 degrees of freedom
## Multiple R-squared:  0.8377, Adjusted R-squared:  0.8366
## F-statistic: 804.9 on 1 and 156 DF,  p-value: < 2.2e-16
```

The p value of the predictor "passengers" is lower than the threshold (0.05), hence we conclude that "passengers" predictor is significant. Also, the degrees of freedom is 156, which is obtained by subtracting the number of predictors (intercept and passengers) from the number of observations (158).

Residuals represent the difference between the actual value and the predicted value of the observations.
Coefficients represent the coefficients of the predictor variables.
Standard error represents the standard deviation of the coefficients.
Residual standard error measures how well our model fits the data.

Answer 2 (a)

```
x = cship["passengers"]
y = cship["crew"]
SXX = sum((x - lapply(x, mean, na.rm = TRUE)) ^ 2)
SXX
```

```
## [1] 14702.45
```

Answer 2 (b)

```
SXY = sum((x - lapply(x, mean, na.rm = TRUE)) * (y - lapply(y, mean, na.rm = TRUE)))
SXY
```

```
## [1] 4871.664
```

Answer 2 (c)

```
SYY = sum((y - lapply(y, mean, na.rm = TRUE)) ^ 2)
SYY
```

```
## [1] 1927.084
```

Answer 2 (d)

```r
model_matrix = model.matrix(model1)
head(model_matrix)
```

```
##   (Intercept) passengers
## 1           1       6.94
## 2           1       6.94
## 3           1      14.86
## 4           1      29.74
## 5           1      26.42
## 6           1      20.52
```

```r
dim(model_matrix)
```

```
## [1] 158   2
```

The dimensions of model matrix is 158*2. There are 158 rows and 2 columns. First column denotes the intercept value, and it has value 1 for all rows. Second column denotes the number of passengers.

Answer 2 (e)

```r
beta_hat_matrix = solve(t(model_matrix) %*% (model_matrix)) %*% (t(model_matrix)) %*% (crew)
beta_hat_matrix
```

```
##                  [,1]
## (Intercept) 1.6783063
## passengers  0.3313505
```

$\hat{\beta}_0$: 1.6783063

$\hat{\beta}_1$: 0.3313505

Answer 2 (f)

```r
beta1_hat = cor_passengers_ship * sqrt(SYY / SXX)
beta1_hat
```

```
## [1] 0.3313505
```

Answer 2 (g)

```r
residuals = model1$residuals
rss = sum(residuals ^ 2)
estimate_error_variance = rss / (nrow(y)-2)
rss
```

```
## [1] 312.8553
```

```
estimate_error_variance
```

```
## [1] 2.005482
```

RSS : 312.855258

Estimate of Error variance is : 2.0054824

Answer 2 (h)

```
model_matrix = model.matrix(model1)
var_cov_matrix = estimate_error_variance * (solve(t(model_matrix)%*%(model_matrix)))
var_cov_matrix
```

```
##              (Intercept)    passengers
## (Intercept)  0.059162695 -0.0025176761
## passengers  -0.002517676  0.0001364047
```

Element in row 2, col 2 represents the variance of β_1.
Element in row 2, col 1 represehts the covariance between β_0 and β_1.

Answer 2 (i)

```
y_hat = predict(model1, x)
y = cship$crew
r_2 = (sum((y_hat - mean(y))^2)) / (sum((y - mean(y))^2))
r_2
```

```
## [1] 0.8376535
```

Answer 3 (a)

```
set.seed(217)
x1 = rexp(30, rate = 1 / 5)
x1
```

```
##  [1]  6.8195610  7.8104816  3.6355385  3.3447575 16.1531259  2.4394748
##  [7] 16.7663162  9.0753149  8.9110977  3.9006997  7.5098447  0.3703376
## [13]  4.7879185  5.7970030  1.8140738  4.9651593  0.4478587  0.4249035
## [19] 15.8782526  1.7234480  6.2991885  3.0500740  3.3202650  3.8183520
## [25]  6.6633466  0.2577323  1.4619528  2.5735161  0.9237832  0.4454144
```

Answer 3 (b)

```r
x2 = rnorm(30, mean = 5, sd = 3)
x2
```

```
##  [1]  5.0876330  5.6508350  4.9289830  5.8298614 10.4766158  5.2704834
##  [7]  6.1145465  6.3792997 10.6019073  9.7848150  7.7919125  6.5874363
## [13]  6.7674992  1.2348815  1.4040839  4.6910677  3.2543531  1.9936569
## [19]  0.6397992  8.9389485  3.9977444  5.7890879  8.5374721  2.7705069
## [25]  6.3991604  2.9227973  8.1632154  7.3234117 10.0069486  2.4881311
```

Answer 3 (c)

```r
error = rnorm(30, mean = 0, sd = 2)
y = (2 * x1) - (6 * x2) + error
head(y)
```

```
## [1] -16.65432 -20.23328 -21.43552 -28.49523 -28.01137 -28.50636
```

Answer 3 (d)

```r
mlr_fit = lm(y ~ x1 + x2, data = data.frame(cbind(y,x1,x2)))
summary(mlr_fit)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = data.frame(cbind(y, x1, x2)))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.044 -1.233  0.013  1.120  4.052
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.84383    0.78570  -1.074    0.292
## x1           1.93469    0.07114  27.197   <2e-16 ***
## x2          -5.79826    0.11609 -49.947   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.756 on 27 degrees of freedom
## Multiple R-squared:  0.9909, Adjusted R-squared:  0.9903
## F-statistic:  1477 on 2 and 27 DF,  p-value: < 2.2e-16
```

Answer 3 (e)

```r
y_hat = predict(mlr_fit, newdata =  data.frame(cbind(x1, x2)))
epsilon_hat = y - y_hat
t(model.matrix(mlr_fit)) %*% epsilon_hat
```

```
##                         [,1]
## (Intercept) -1.243450e-14
## x1          -3.496127e-13
## x2          -2.372824e-13
```

Answer 3 (f)

```r
summary(mlr_fit)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = data.frame(cbind(y, x1, x2)))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.044 -1.233  0.013  1.120  4.052
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.84383    0.78570  -1.074    0.292
## x1           1.93469    0.07114  27.197   <2e-16 ***
## x2          -5.79826    0.11609 -49.947   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.756 on 27 degrees of freedom
## Multiple R-squared:  0.9909, Adjusted R-squared:  0.9903
## F-statistic:  1477 on 2 and 27 DF,  p-value: < 2.2e-16
```

Both x1 and x2 are significant at p=0.05. They are significant since they were used to generate the dependent variable y.

Answer 3 (g)

```r
mlr_model_matrix = model.matrix(mlr_fit)
hat_matrix = (mlr_model_matrix) %*% (solve(t(mlr_model_matrix) %*% mlr_model_matrix)) %*% (t(mlr_model_
hat_matrix[1:6, 1:6]
```

```
##            1          2            3           4            5           6
## 1 0.04099921 0.04219162  0.031624879 0.027695653  0.051948089  0.02675687
## 2 0.04219162 0.04602698  0.027869090 0.025452928  0.078215347  0.02200332
## 3 0.03162488 0.02786909  0.038670543 0.036529445 -0.004007154  0.04009708
## 4 0.02769565 0.02545293  0.036529445 0.038238500  0.006654172  0.04024319
## 5 0.05194809 0.07821535 -0.004007154 0.006654172  0.300811242 -0.01811574
## 6 0.02675687 0.02200332  0.040097080 0.040243193 -0.018115741  0.04463846
```

Hat matrix maps the vector of response values (dependent variable values) to the vector of fitted values (or predicted values). It describes the influence each response value has on each fitted value.