Q1

```r
install.packages("scatterplot3d")
library(scatterplot3d)

#a)
x <- c(1, -3, 0, 5)
y <- c(3, 1, 7, 0)
sum_xy <- x + y


#b)
operation_xy <- 1.9 * x - 0.75 * y


#c)
dot_product <- sum(x * y)


#d)
cross_product <- c(x[2]*y[3] - x[3]*y[2], x[3]*y[1] - x[1]*y[3], x[1]*y[2] -
x[2]*y[1])



#e)
matrix_xy <- rbind(x, y)

rank_xy <- qr(matrix_xy)$rank


is_independent <- rank_xy == 2
is_independent
```

Q2

```r

# Q2.

# a)
A <- matrix(c(1, 3, 1, 1, 0, 0, 0, 5, 0), nrow = 3, byrow = TRUE)
B <- matrix(c(0, 0, 5, 7, 5, 0, 0, 0, 0), nrow = 3, byrow = TRUE)


sum_ab <- A + B

sum_ab
```

```r
#b)
A <- matrix(c(1, 3, 1, 1, 0, 0, 0, 5, 0), nrow = 3, byrow = TRUE)
B <- matrix(c(0, 0, 5, 7, 5, 0, 0, 0, 0), nrow = 3, byrow = TRUE)

scalar_ab <- 4.5 * A - 3.2 * B

scalar_ab



#c) #
A <- matrix(c(1, 3, 1, 1, 0, 0, 0, 5, 0), nrow = 3, byrow = TRUE)
B <- matrix(c(0, 0, 5, 7, 5, 0, 0, 0, 0), nrow = 3, byrow = TRUE)

sum_ab <- A + B

sum_ab



#d)
A <- matrix(c(1, 3, 1, 1, 0, 0, 0, 5, 0), nrow = 3, byrow = TRUE)
B <- matrix(c(0, 0, 5, 7, 5, 0, 0, 0, 0), nrow = 3, byrow = TRUE)

scalar_ab <- 4.5 * A - 3.2 * B

scalar_ab


#e)
A <- matrix(c(1, 3, 1, 1, 0, 0, 0, 5, 0), nrow = 3, byrow = TRUE)
B <- matrix(c(0, 0, 5, 7, 5, 0, 0, 0, 0), nrow = 3, byrow = TRUE)

product_ab <- A %*% B

product_ab


#f)
A <- matrix(c(1, 3, 1, 1, 0, 0, 0, 5, 0), nrow = 3, byrow = TRUE)
B <- matrix(c(0, 0, 5, 7, 5, 0, 0, 0, 0), nrow = 3, byrow = TRUE)

rank_A <- qr(A)$rank

rank_B <- qr(B)$rank
```

```
rank_A
rank_B
```

Q3

```
# Define matrix A
A <- matrix(c(1, 3, 1, 1, 0, 0, 0, 5, 0), nrow = 3, byrow = TRUE)

#a)
C <- A %*% A
print("Matrix C:")
print(C)



#b)
D <- diag(C)
print("Diagonal elements of C (Matrix D):")
print(D)



#c)

A <- matrix(c(1, 2, 3, 4), nrow=2, ncol=2)


C <- A %*% A

if (all(eigen(C)$values > 0)) {

  L <- chol(C)

  L <- t(L)
  print(L)
} else {
  print
}
```

Q4

```
E <- matrix(c(2, 7, 4, 8,
              1, 0, 4, 5,
              1, 6, 9, 1), nrow = 3, byrow = TRUE)

F <- matrix(c(6, 5, 4, 0,
              2, 1, 5, 4,
              0, 0, 0, 2), nrow = 3, byrow = TRUE)




#SVD on Matrix E
```

```r
svd_E <- svd(E)
cat("SVD for Matrix E:\n")
cat("U:\n")
print(svd_E$u)
cat("Singular Values (S):\n")
print(svd_E$d)
cat("V (transposed):\n")
print(svd_E$v)


#SVD on Matrix F
svd_F <- svd(F)
cat("SVD for Matrix F:\n")
cat("U:\n")
print(svd_F$u)
cat("Singular Values (S):\n")
print(svd_F$d)
cat("V (transposed):\n")
print(svd_F$v)
```

Q5

```r

G <- matrix(c(6, 12, 4, 0,
              4, 6, 1, 3,
              2, 6, 11, 9,
              3, 5, 0, 7), nrow = 4, byrow = TRUE)

#a)
G_inv <- solve(G)
print("G inverse:")
print(G_inv)

#b)
trace_G <- sum(diag(G))
cat("Trace of G:", trace_G, "\n")

#c)
det_G <- det(G)
cat("Determinant of G:", det_G, "\n")
```

Q6

```r
# (a): Covariance Matrix
data(cars)

cov_matrix <- cov(cars$speed, cars$dist)

print("Covariance matrix for the cars dataset:")
print(cov_matrix)
```

```r
# (b): Correlation Matrix
data(cars)

corr_matrix <- cor(cars$speed, cars$dist)

print("Correlation matrix for the cars dataset:")
print(corr_matrix)


# (c): Scatterplot
data(cars)

plot(cars$speed, cars$dist, main = "Scatterplot of Speed vs. Stopping
Distance",
     xlab = "Speed (mph)", ylab = "Stopping Distance (ft)", pch = 19)


# (d): Column Means
data(cars)

mean_speed <- mean(cars$speed)
mean_dist <- mean(cars$dist)

cat("Mean Speed:", mean_speed, "mph\n")
cat("Mean Stopping Distance:", mean_dist, "feet\n")
```

Q7

```r
install.packages("IRISSeismic")
library(IRISSeismic)

#a)
data(iris)

cov_matrix_iris <- cov(iris[, 1:4])

print("Covariance Matrix for Iris Dataset:")
print(cov_matrix_iris)


#b)
data(iris)

iris_quant <- iris[, 1:4]

cov_matrix <- cov(iris_quant)
```

```r
print("Covariance matrix for the iris dataset:")
print(cov_matrix)

eigen_values <- eigen(cov_matrix)$values
if(all(eigen_values > 0)) {
  cat("The covariance matrix is positive-definite.\n")
} else if(all(eigen_values >= 0)) {
  cat("The covariance matrix is semi-positive definite.\n")
} else {
  cat("The covariance matrix is neither positive-definite nor semi-positive
definite.\n")
}




#c)
data(iris)


list_iris <- split(iris[,1:4], iris$Species)

cor_matrices <- list()

for(species in names(list_iris)) {
  cor_matrices[[species]] <- cor(list_iris[[species]])
}


for(species in names(cor_matrices)) {
  cat(paste("Correlation matrix for", species, ":\n"))
  print(cor_matrices[[species]])
  cat("\n")
}




#d)
data(iris)

split_data <- split(iris[, 1:4], iris$Species)

corr_matrices <- list()

for(species in names(split_data)) {
  corr_matrices[[species]] <- cor(split_data[[species]])
}
```

```
print("Correlation Matrix for Setosa:")
print(corr_matrices[["setosa"]])

print("Correlation Matrix for Versicolor:")
print(corr_matrices[["versicolor"]])

print("Correlation Matrix for Virginica:")
print(corr_matrices[["virginica"]])
```