BHao_Assign1

Problem Set 1

1)

```
u = c(0.5, 0.5)
v = c(3.0, -4.0)

uv_dot = u %*% v
uv_dot
```

```
##      [,1]
## [1,] -0.5
```

2)

```
# length of u
u_len = sqrt(sum(u**2))
u_len
```

```
## [1] 0.7071068
```

```
#length of v
v_len = sqrt(sum(v**2))
v_len
```

```
## [1] 5
```

3)

```
3*u - 2*v
```

```
## [1] -4.5  9.5
```

4)

```
# u.v = |u| * |v| * cos(theta)
# theta = arccos(u.v / (|u| * |v|))
theta = acos( uv_dot / ( u_len * v_len ) )

# convert from radians to degrees
theta = theta * 180 / pi
theta
```

```
##          [,1]
## [1,] 98.1301
```

Problem Set 2

```
A = matrix(c(1, 1, 3,
             2, -1, 5,
             -1, -2, 4),
           nrow = 3, ncol = 3, byrow = T)

b = matrix(c(1, 2, 6), nrow = 3, ncol = 1)
```

```
solve_by_elim = function(A, b) {
  # placeholder for solution
  solution = matrix(c(0, 0, 0), nrow = 3, ncol = 1)

  # check for zero pivot in first column
  # use row counter and while loop in case multiple rows have zeroes in first column
  r = 2
  while (A[1, 1] == 0) {
    B = A
    t = b
    B[r, ] = A[1, ]
    B[1, ] = A[r, ]
    t[r, ] = b[1, ]
    t[1, ] = b[r, ]
    A = B
    b = t
    r = r + 1
  }

  # solve first pivot
  mult_2_1 = A[2, 1] / A[1, 1]
  A[2, ] = A[2, ] - mult_2_1 * A[1, ]
  b[2] = b[2] - mult_2_1 * b[1]

  mult_3_1 = A[3, 1] / A[1, 1]
  A[3, ] = A[3, ] - mult_3_1 * A[1, ]
  b[3] = b[3] - mult_3_1 * b[1]

  # check for zero pivot in second column
  if (A[2, 2] == 0) {
    B = A
    t = b
    B[3, ] = A[2, ]
    B[2, ] = A[3, ]
    t[3, ] = b[2, ]
    t[2, ] = b[3, ]
    A = B
    b = t
  }

  # solve second pivot
  mult_3_2 = A[3, 2] / A[2, 2]
  A[3, ] = A[3, ] - mult_3_2 * A[2, ]
  b[3] = b[3] - mult_3_2 * b[2]


  # backsolve
  solution[3] = b[3] / A[3, 3]
  solution[2] = ( b[2] - A[2, 3] * solution[3] ) / A[2, 2]
  solution[1] = ( b[1] - A[1, 3] * solution[3] - A[1, 2] * solution[2] ) / A[1, 1]

  return(solution)
}
```

```
solution_elim = solve_by_elim(A, b)
solution_elim
```

```
##            [,1]
## [1,] -1.5454545
## [2,] -0.3181818
## [3,]  0.9545455
```

```
# check - use solve to invert matrix A
solution_solve = solve(A) %*% b
solution_solve
```

```
##            [,1]
## [1,] -1.5454545
## [2,] -0.3181818
## [3,]  0.9545455
```

```
# test zero pivot cases
A = matrix(c(2, 1, 1,
             4, 2, 0,
             -2, 7, 2),
           nrow = 3, ncol = 3, byrow = TRUE)

solve_by_elim(A, b)
```

```
##        [,1]
## [1,] 0.0625
## [2,] 0.8750
## [3,] 0.0000
```

```
solve(A) %*% b
```

```
##        [,1]
## [1,] 0.0625
## [2,] 0.8750
## [3,] 0.0000
```

```
A = matrix(c(0, 1, 1,
             0, 1, 0,
             -2, 7, 2),
           nrow = 3, ncol = 3, byrow = TRUE)

solve_by_elim(A, b)
```

```
##      [,1]
## [1,]    3
## [2,]    2
## [3,]   -1
```

```
solve(A) %*% b
```

```
##      [,1]
## [1,]    3
## [2,]    2
## [3,]   -1
```