

```

mat <- list(10, 52, 28, 35, 0, 11, 19, 40, 26, 12, 9, 7, 6, 5, 16, 15, 10,
99, 87, 81, 19, 22, 33, 91, 200)
prev_mat <- matrix(data = mat, nrow = 5, ncol = 5, byrow = FALSE)
prev_mat

mat21 <- c(1:9)
mat_21 <- matrix(data = mat21, nrow = 2, ncol = 10, byrow = TRUE)
mat_21

# 7.
# Create a user defined function named student function that transposes a
numeric matrix
# (columns become rows) and subsetsvthe matrix in a way so that only the
first 5 rows
# remain in the dataset.

studentfunction <- function(df) {
  new_matrix <- as.data.frame(matrix(nrow = ncol(df), ncol = nrow(df)))
  for (a in 1:nrow(df)) { # this loop goes over the columns
    for (b in 1:ncol(df)) { # this loop goes over the rows
      new_matrix[b, a] <- df[a, b]
    }
  }
  if (nrow(new_matrix) > 5) {
    return(new_matrix[1:5, ])
  } else {
    return(new_matrix)
  }
}

new_prev_mat_3_7 <- studentfunction(prev_mat)
new_prev_mat_3_7

new_mat_21_3_7 <- studentfunction(mat_21)
new_mat_21_3_7

# 8
# Create a user defined function named transformmatrix that takes
# the diagonal of a matrix and calculates a vector with two elements.
# Element one is the mean of the diagonal and element two is the median.
transformmatrix <- function(my_matrix) {
  n <- min(nrow(my_matrix), ncol(my_matrix))
  new_df <- c()
  for (i in 1:n) {
    new_df[i] <- (my_matrix[i, i])
  }
  return(c(mean(as.numeric(new_df)), median(as.numeric(new_df))))
}

my_vector_3_8 <- transformmatrix(prev_mat)
my_vector_3_8

new_mat_21_3_8 <- transformmatrix(mat_21)
new_mat_21_3_8

```

```

# 9.
# For the iris dataset (no need to call a function, iris is part of the
base R)
# create for loop that does the following to each observation:
# a) changes the Species column from a character type to numeric.
# Assign 1 for setosa, 2 for virginica, and 3 for versicolor,
# b) creates a new column that groups the Petal.Length into 3 groups:
# group#1 for Petal.Length from 0 to 2, group #2 from 2.01 to 4.5, and
group #3 from 4.51 to 7.

iris

iris$Species <- gsub("setosa", "1", iris$Species)
iris$Species <- gsub("virginica", "2", iris$Species)
iris$Species <- gsub("versicolor", "3", iris$Species)
iris$Species <- as.numeric(iris$Species)

hist(german_credit$age)
boxplot(age ~ good_bad, data = german_credit)
iris

for (i in 1:nrow(iris)) {
  if (iris$Petal.Length[i] >= 0 & iris$Petal.Length[i] <= 2) {
    iris$Group[i] <- 1
  } else if (iris$Petal.Length[i] >= 2.1 & iris$Petal.Length[i] <= 4.5) {
    iris$Group[i] <- 2
  } else if (iris$Petal.Length[i] >= 4.51 & iris$Petal.Length[i] <= 7) {
    iris$Group[i] <- 3
  } else {
  }
}
iris

#####
# Chapter 6.7: Exercise: 3

# 3.Using the iris dataset:
# a) combine the Setosa and Versicolor into group "0" and label the
Virginica to "1". Create a new variable called iris$Group with the 0 or 1
labels,
# b) build a logistic regression model using any available data that will
predict the observation being Virginica ( value of 1 in Group variable),
# c) calculate the probability of a new plant being a Virginica for the
following parameters:
# Sepal.Width =5
# Petal.Length =10
# Petal.Width =7
# Sepal.Length=9

# a
iris
for (i in 1:nrow(iris)) {

```

```

    if (iris$Species[i] == "setosa" | iris$Species[i] == "versicolor") {
      iris$Group[i] <- 0
    } else if (iris$Species[i] == "virginica") {
      iris$Group[i] <- 1
    } else {
      iris$Group[i] <- 2
    }
  }
}
iris

```

```

# b
my_model_6_7 <- glm(Group ~ Sepal.Width + Petal.Length + Petal.Width +
  Sepal.Length, data = iris, family = "binomial")
summary(my_model_6_7)

```

```

# c
my_model_6_7_func <- function(b0, b1, b2, b3, b4, x1, x2, x3, x4){
  logit <- (b0 + x1*b1 + x2*b2 + x3*b3 + x4*b4)
  odds <- exp(logit)
  probs <- odds/(1+odds)
  return (c(odds, probs))
}

```

```

# numbers from my_model summary
my_model_6_7_func(-42.638,-6.681,9.429,18.286,-2.465,5,10,7,9)

```