Q1

```r
#1a
chest<-c(rep("gold_coins",20),rep("silver",30),rep("bronze",50))
sample(chest,10)

#1b
sample(c("success", "failure"), 10, replace = TRUE, prob = c(0.9, 0.1))
```

```
> #1a
> chest<-c(rep("gold_coins",20),rep("silver",30),rep("bronze",50))
> sample(chest,10)
 [1] "bronze"     "bronze"     "gold_coins" "gold_coins" "silver"     "gold_coins"
 [7] "bronze"     "silver"     "silver"     "bronze"    
> 
> #1b
> sample(c("success", "failure"), 10, replace = TRUE, prob = c(0.9, 0.1))
 [1] "success" "success" "success" "success" "success" "failure" "success" "success"
 [9] "success" "success"
> 
```

Q2

```r
#2a
# Function to simulate the probability of a birthday match for a given n
simulate_birthday_probability <- function(n, num_simulations) {
  # Initialize a counter to keep track of matches
  match_count <- 0

  # Run simulations
  for (i in 1:num_simulations) {
    # Generate n random birthdays (from 1 to 365)
    birthdays <- sample(1:365, n, replace = TRUE)

    # Check if there's a match
    if (length(birthdays) != length(unique(birthdays))) {
      match_count <- match_count + 1
    }
  }

  # Calculate the probability of a match
  probability <- match_count / num_simulations

  return(probability)
}

# Set the number of simulations
num_simulations <- 10000

# Find the smallest n for which the probability of a match is greater than 0.5
smallest_n <- NULL
for (n in 2:365) {
  probability <- simulate_birthday_probability(n, num_simulations)
  if (probability > 0.5) {
    smallest_n <- n
    break
  }
}

# Print the results
cat("Smallest n for which the probability of a match is greater than 0.5:", smallest_n, "\n")
```

```
Smallest n for which the probability of a match is greater than 0.5: 23
> 
```

Q3

```
conditional_prob<-function(P_cloud,P_rain,P_cloud_rain){


  P_rain_cloud<-P_cloud_rain*P_rain/P_cloud

  return (P_rain_cloud)

}
P_cloud<-0.4
P_rain<-0.2
P_cloud_rain<-0.85

ans<-conditional_prob(P_cloud,P_rain,P_cloud_rain )
print(ans)
```

```
> conditional_prob<-function(P_cloud,P_rain,P_cloud_rain){
+
+
+   P_rain_cloud<-P_cloud_rain*P_rain/P_cloud
+
+   return (P_rain_cloud)
+
+ }
> P_cloud<-0.4
> P_rain<-0.2
> P_cloud_rain<-0.85
>
> ans<-conditional_prob(P_cloud,P_rain,P_cloud_rain )
> print(ans)
[1] 0.425
```

Q4

```r
#4
# Load the Iris dataset
data(iris)

# (a) Print the first few rows of the dataset
head(iris)

# (b) Find the structure of the dataset
str(iris)

# (c) Find the range of sepal length
range_sepal_length <- range(iris$Sepal.Length)
cat("Range of Sepal Length:", range_sepal_length[1], "to", range_sepal_length[2], "\n")

# (d) Find the mean of sepal length
mean_sepal_length <- mean(iris$Sepal.Length)
cat("Mean Sepal Length:", mean_sepal_length, "\n")

# (e) Find the median of sepal length
median_sepal_length <- median(iris$Sepal.Length)
cat("Median Sepal Length:", median_sepal_length, "\n")

# (f) Find the first and third quartiles and the interquartile range for sepal length
quartiles_sepal_length <- quantile(iris$Sepal.Length, c(0.25, 0.75))
iqr_sepal_length <- diff(quartiles_sepal_length)
cat("First Quartile:", quartiles_sepal_length[1], "\n")
cat("Third Quartile:", quartiles_sepal_length[2], "\n")
cat("Interquartile Range:", iqr_sepal_length, "\n")

# (g) Find the standard deviation and variance of sepal length
std_dev_sepal_length <- sd(iris$Sepal.Length)
variance_sepal_length <- var(iris$Sepal.Length)
cat("Standard Deviation of Sepal Length:", std_dev_sepal_length, "\n")
cat("Variance of Sepal Length:", variance_sepal_length, "\n")
```

```r
# (h) Repeat the above exercises for sepal.width, petal.length, and petal.width
# Sepal Width
range_sepal_width <- range(iris$Sepal.Width)
mean_sepal_width <- mean(iris$Sepal.Width)
median_sepal_width <- median(iris$Sepal.Width)
quartiles_sepal_width <- quantile(iris$Sepal.Width, c(0.25, 0.75))
iqr_sepal_width <- diff(quartiles_sepal_width)
std_dev_sepal_width <- sd(iris$Sepal.Width)
variance_sepal_width <- var(iris$Sepal.Width)

# Petal Length
range_petal_length <- range(iris$Petal.Length)
mean_petal_length <- mean(iris$Petal.Length)
median_petal_length <- median(iris$Petal.Length)
quartiles_petal_length <- quantile(iris$Petal.Length, c(0.25, 0.75))
iqr_petal_length <- diff(quartiles_petal_length)
std_dev_petal_length <- sd(iris$Petal.Length)
variance_petal_length <- var(iris$Petal.Length)

# Petal Width
range_petal_width <- range(iris$Petal.Width)
mean_petal_width <- mean(iris$Petal.Width)
median_petal_width <- median(iris$Petal.Width)
quartiles_petal_width <- quantile(iris$Petal.Width, c(0.25, 0.75))
iqr_petal_width <- diff(quartiles_petal_width)
std_dev_petal_width <- sd(iris$Petal.Width)
variance_petal_width <- var(iris$Petal.Width)

# (i) Use the built-in function summary on the dataset Iris
summary(iris)
```

```
> #4
> # Load the Iris dataset
> data(iris)
>
> # (a) Print the first few rows of the dataset
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
>
> # (b) Find the structure of the dataset
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
>
> # (c) Find the range of sepal length
> range_sepal_length <- range(iris$Sepal.Length)
> cat("Range of Sepal Length:", range_sepal_length[1], "to", range_sepal_length[2], "\n")
Range of Sepal Length: 4.3 to 7.9
>
> # (d) Find the mean of sepal length
> mean_sepal_length <- mean(iris$Sepal.Length)
> cat("Mean Sepal Length:", mean_sepal_length, "\n")
Mean Sepal Length: 5.843333
>
> # (e) Find the median of sepal length
> median_sepal_length <- median(iris$Sepal.Length)
> cat("Median Sepal Length:", median_sepal_length, "\n")
Median Sepal Length: 5.8
>
> cat("First Quartile:", quartiles_sepal_length[1], "\n")
First Quartile: 5.1
> cat("Third Quartile:", quartiles_sepal_length[2], "\n")
Third Quartile: 6.4
> cat("Interquartile Range:", iqr_sepal_length, "\n")
Interquartile Range: 1.3
>
> # (g) Find the standard deviation and variance of sepal length
> std_dev_sepal_length <- sd(iris$Sepal.Length)
> variance_sepal_length <- var(iris$Sepal.Length)
> cat("Standard Deviation of Sepal Length:", std_dev_sepal_length, "\n")
Standard Deviation of Sepal Length: 0.8280661
> cat("Variance of Sepal Length:", variance_sepal_length, "\n")
Variance of Sepal Length: 0.6856935

> # (i) Use the built-in function summary on the dataset Iris
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
>
```

Q5

```r
#5

calculate_mode <- function(x) {
  unique_values <- unique(x)
  unique_counts <- table(x)
  modes <- unique_values[unique_counts == max(unique_counts)]
  return(modes)
}

data_vector <- c(2, 3, 4, 3, 5, 6, 4, 4, 7)
result <- calculate_mode(data_vector)

cat("Mode(s) of the dataset:", result, "\n")
```

```
> cat("Mode(s) of the dataset:", result, "\n")
Mode(s) of the dataset: 4
>
```